Inf-3990
Master's Thesis in Computer Science

M2S AND CAIR:
Image based information retrieval in mobile environments.

Anne Staurland Aarbakke

May 1st, 2007

Faculty of Science
Department of Computer Science
University of Tromsø

Inf-3990
Master's Thesis in Computer Science

M2S and CAIR:
Image based information retrieval in mobile environments.

Anne Staurland Aarbakke

May 1st, 2007

To my late sister Ingvil, who encouraged me and supported me through my years of higher education. You inspire me!

Abstract

Images are commonly used on a daily basis for research, information and entertainment. The introduction of digital cameras and especially the incorporation of cameras in mobile phones makes people able to snap photos almost everywhere at any time since their mobile phone is almost always brought with them.
The fast evolution in hardware enables users to store large image collection without high costs. Making use of these image collections requires efficient image retrieval techniques. Traditional image retrieval techniques like text-based image retrieval and content-based image retrieval techniques have shortcomings. New techniques or combination of existing techniques must be established to provide users with adequate image retrieval functionality.

This thesis describes two systems enabling users to retrieve information such as images, textual information, WAP-links or videos using SMS or MMS.

One of the services, M2S is meant for tourists to retrieve information about attractions in Lofoten. M2S uses content-based image retrieval to retrieve the information requested. This service is designed and implemented in cooperation with Telenor R&I.

The other system, CAIR is meant for users who want to retrieve images from an image collection using SMS. CAIR uses a context-based image retrieval to retrieve images. This system is designed, but not yet implemented.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Randi Karlsen for her great patience, support and invaluable feedback during the work of this thesis.

I also thank Sigmund Akselsen, Bente Evjemo and Anders Schürmann at Telenor Research and Innovation (Telenor R&I) for their valuable feedback and cooperation.

A special thanks goes to my family who has supported me through more than 17 years of education, especially my brother Jon for proofreading and feedback of this thesis.

I also thank my dear Johnny Bjørnsen for his valuable feedback and his great patience.

A thanks also goes to my good friend Dan Peder Eriksen for supporting me and contribute to an enjoyable working environment in the lab.

Contents

List of figures

List of tables

List of abbreviations

API	Application Programming Interface
CAIM	Context-Aware Image Management
CAIR	Context Aware Image retrieval
CBIR	Content-Based Image Retrieval
CIRES	Content Based Image REtrieval System
CPU	Central Processing Unit
DC	Dublin Core
EXIF	Exchangeable Image File Format
GIFT	The GNU Image-Finding Tool
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HSV	Hue, Saturation, Value
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IR	Infra Red
M2S	MMS To Search
MASCOT	Mechanism for Attention-based Scale-invariant Object Recognition in Images
MMS	Multimedia Messaging System
MMSC	Multimedia Messaging Service Center
MPEG	Moving Picture Experts Group
MUVIS	MUltimedia Video Indexing and retrieval System
NTNU	Norwegian University of Science and Technology
PATS	Program for Advanced Telecom Services
PDA	Personal Digital Assistant
QBIC	Query By Image Content
R&I	Research and Innovation
RBIR	Region-Based Image Retrieval
SIMBA	Search IMages By Appearance
SIMPLIcity	Semantics-sensitive Integrated Matching for Picture LIbraries
SMS	Short Message System
SMSC	Short Message Service Centre
SQL	Structured Query Language
TBIR	Text-Based Image Retrieval
UiB	University of Bergen
UiTø	University of Tromsø
URL	Uniform Resource Locator
VIPER	Visual Information Processing for Enhanced Retrieval
Virage	VIR Image Engine
WALRUS	Wavelet-Based Retrieval of User-Specified Scenes
WAP	Wireless Application Protocol

1 Introduction

In this thesis we will describe two mobile services that enables users to retrieve images, videos, textual descriptions or other types of information from a data collection using MMS (Multimedia Messaging Service) or SMS (Short Message System). In this chapter we will first present the motivation for this thesis. Traditional techniques for image retrieval and context will then be introduced. A goal for this thesis will be established and sub-goals will be listed. We will then describe what we contribute with in this thesis.

1.1 Motivation

The number of images, both private and public increases every day due to the rapid growth in mobile technology the latest years. Mobile entities include PDA's (Personal Digital Assistant) and mobile phones. This has encouraged the use of digital images as one of the most important communication media for daily life. Images are commonly used on a daily basis for research, information and entertainment. The introduction of the digital camera and especially the incorporation of these into mobile devices such as PDA's and mobile telephones, enables people to take pictures almost anywhere and at any time. Several applications have been developed to support this trend, and more is yet to come. People can for instance distribute images quickly and easily by sending an MMS with their mobile phones. People have got used to retrieve desired information when and where they want it, since their mobile phone is almost always with them. The desired information is often only a call or SMS /MMS away.

Storage capacity to both consumers and businesses has increased due to development in storage mediums and the decreasing costs of hardware in general. Capturing and storing pictures is easy. Since lack of storage space is not an issue, consumers have the possibility of storing images without regard to their quality and future use. Images are captured and stored in large scale and this leads to image collections getting larger and harder to handle.

To exploit the potential of all the images taken, users must have a way to locate desired images. In contrast to the growth of hardware, the software for managing large image collections is quite limited. The biggest challenge concerning management of image collections is probably the retrieval of images. There have been many attempts to make full worthy image retrieval software. The ones that exist can roughly be put in two categories, text-based image retrieval (TBIR) and content-based image retrieval (CBIR). Current image retrieval solves some issues related to image retrieval but leaves some problems unsolved. We want to study and test information retrieval in mobile environment. We also want to investigate context as supplement for text-based and content-based image retrieval.

This is background for designing two mobile services where the goal is to provide users with requested information using MMS or SMS.

1.2 Current image retrieval techniques

Text-based image retrieval finds images based on textual metadata associated with the image. The metadata can for instance be information like location, time, what the picture is about, who is on the picture and who captured it. Text-based image retrieval requires manual annotation of images. Annotating every image in a collection manually is a very time-consuming task that few users prioritize. Even if users annotate the images, this is a highly subjective task and personal phrases are chosen. A user can make use of synonyms and the annotation text can vary a lot from person to person. This makes the retrieval difficult. The query is formulated as text and because string comparison is simple and fast, the search itself is fast, but often very inaccurate.

Content-based image retrieval techniques analyze the actual image. Low-level features of the image like color, shape and texture are extracted and compared to other images to find matches. The query is formulated as a drawing, sketch or an example image. It can also be formulated using specific color compositions. The query and the content of an image collection are compared and results are returned based on a similarity algorithm. There are several problematic issues with these image retrieval techniques that are yet to be solved. Using the visual contents of an image for image retrieval returns far from satisfying results. For controlled environments like fingerprint and face recognition, such applications are successful. However for general purpose applications these are not sufficient techniques [1] [2].

As mentioned content-based image retrieval techniques extract the low-level features of an image such as color, texture and shape. Humans have the ability to perceive high-level features of images that computers are unable to detect, such as the happening depicted in the image. This creates a gap between what the user wants and what he is actually capable to express in a query. This is referred to as The Semantic Gap and is the most important problem addressed in image retrieval theory [3]. Developing applications that meet the human way of formulating queries is very difficult and several approaches are tested [2] [4].

This thesis is part of the CAIM-project (Context-Aware Image Management) [5], which uses context to narrow the semantic gap. CAIM is a cooperation project between the University of Bergen (UiB), Norwegian University of Science and Technology (NTNU), Telenor and the University of Tromsø (UiTø).

1.3 What is context?

Context is a very wide concept and it is used in several fields. Giving a clear definition of context is not easy. However, there have been several attempts to make one. We believe Dey's [6] definition suits our use:

"*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves*".

Context is the where, who, what and when of an object. If a piece of information can be used to characterize the situation of a participant in an interaction or conversation, this information is context [6] [7]. A word can be interpreted different depending on the setting it is brought up in. The same word can be interpreted quite different from person to person. People's background and understanding is essential to have the same comprehension of a context [6] [7].

This can easily be illustrated with an example. The word "Java" can have different meanings. If the word Java is brought up in a discussion among computer scientists, it is most likely the programming language Java they refer to. If the word Java is brought up in a discussion about traveling, it might be in the meaning of the island of Indonesia, or if the discussion is between baristas, the Java in that context is probably the type of coffee.

The Java-example illustrates how a word can have different meaning depending on who interpret it. This also applies to images. Every image has a context and the happening (event) in a picture is interpreted differently depending on the person who sees it. People's minds make their own interpretation of the picture. A drawing or picture of a man with an apple in his hand might be interpreted quite differently between a nutritionist and a physicist. The nutritionist might immediately think of a healthy person and the physicist might immediately think of Isaac Newton and the apple that supposedly hit his head. Even if two persons have a very similar understanding of the world and how it works, the probability that they will interpret the same image differently is very high.

A word can also have different meaning dependent of the situation the person is in. When he is in the coffee-shop, he intuitively perceives Java as coffee, while he is at the traveling agency, Java is the Island of Indonesia and when he is the computer lab Java is the programming language. The person is the same, but the person's context has changed.

Humans use context on daily basis. We do not explain to each other every detail of what we want to communicate because some of this information is already perceived by the user. Computers however, do not have this ability. We want computers to be more context-aware. By context-aware we mean computers ability to be aware of and adapt to the context that surrounds them [7] [6] [8] [9] [10]. Ideally we want computers to be able to perceive the "silent knowledge" that humans share, that enables us to interpret what is communicated without expressing it explicit. Humans are surrounded by different context and handle it naturally. Some of the user's context is of a nature that makes it easy for computers to retrieve images. Time and location are examples of such types of context and they are applied a lot in the world of context-awareness. The far most used context in context-aware computing is location.

1.4 Information retrieval in mobile environments

We live in an information society and people are overloaded with information. TV- and radio-commercials, posters and other types of advertisements are displayed and distributed on busses, airports, in shopping centers, to mailboxes and e-mailboxes. People are getting lots of information without asking for it. This might give rise to the expectation that when they need information about something they will get it right away. Allan et al. [11] have defined what information retrieval concerns:

Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information [11].

Information retrieval technology has continually grown to meet challenges presented by new forms of usage. One new and expanding application environment is that of wireless mobile devices such as mobile phones and PDA’s. It is therefore natural to suggest that users will to a larger extent base their information retrieval on using their networked mobile devices. With mobile applications, some aspects of the user's context, for instance their location, are often available, and this context can affect what information is relevant to the user [8] [12].

Today, information retrieval systems go beyond the retrieval of traditional documents. Question answering, topic detection and tracking, summarization, multimedia retrieval (for instance image, video and music), software engineering, chemical and biological informatics, text structuring, text mining, and genomics are fields where information retrieval plays a leading part [11]. It is common for users to have access to large amounts of distributed information through wireless connections. The mobile phones have played a major part in the fierce entry of mobile entities. This transfer of user environment raises new challenges regarding the development of information systems appropriate for mobile environments. Developers will meet challenges regarding resource management, information retrieval and data distribution [13].

1.5 Goal

The goal of this thesis is to investigate context-aware image retrieval in a mobile environment. This thesis also has two sub-goals:

- An intermediate aim of this thesis is to design and develop a content-based information retrieval for mobile environment. We want to make a fun, easy-to-use, entertaining and useful service meant for mobile phones. We also want to look into content-based image retrieval techniques because this is and has been a debated technique in the literature and many commercial companies are now investigating the use of content-based image retrieval in their services.
- Another intermediate aim is to design a context-based image retrieval service for mobile environment. The use of context is said to solve problems related to content and text-based information retrieval. We wanted to experience context-based image retrieval by designing a system that utilizes this concept.

1.6 Method

We have studied the use of image retrieval techniques such as content-based and text-based image retrieval and investigated the use of context and how context can be used for information retrieval. Mobile environments have been considered and how image retrieval can be applied to such environments. We have designed and developed a service called M2S (MMS To Search) that enables users to retrieve information about a specific entity using MMS. M2S has been developed and tested. We have also designed a service called CAIR (Context-Aware Image Retrieval) that enables users to retrieve information about specific locations using SMS. CAIR is not implemented yet.

When designing and developing we have used an iterative approach. By iterative we mean that during the work with the designs and implementation of the systems we have had to go back to the drawing table and discuss alternative solutions as problems were encountered. Testing and evaluation of M2S also led to new requirements and changes in the design. When designing and planning the implementation of CAIR, we have adopted ideas from the design and implementation of M2S.

1.7 Contributions

This thesis is made as a part of the CAIM-project [5]. M2S is designed and implemented in cooperation with Telenor R&I (Telenor Research and Innovation) and belongs to the MOVE-project [14]. The MOVE-project develops information- and communication infrastructure to provide tourists with digital information and services. M2S has been developed and partly tested. The development and testing was done at Telenor R&I. The other service, CAIR, belongs to the CAIM-project. CAIR is designed but not yet implemented. However, we have a plan for how to implement this service.

There are several contributions in this thesis, but the main contributions are related to the design and development of M2S, the design of CAIR and discussions related to these. The contributions are in the following list:

- First we will describe M2S. M2S is a mobile application that enables users to retrieve information about entities photographed while the user is on the move. The photograph is sent as MMS and is used as a query to retrieve information. A possible use is for tourists that want information about attractions they are passing while traveling. The application is, at the time being, only for a demo purpose and is restricted for use in Lofoten only. The purpose of the demo is to show that it is possible to provide users with requested information initiated with an MMS containing an image.

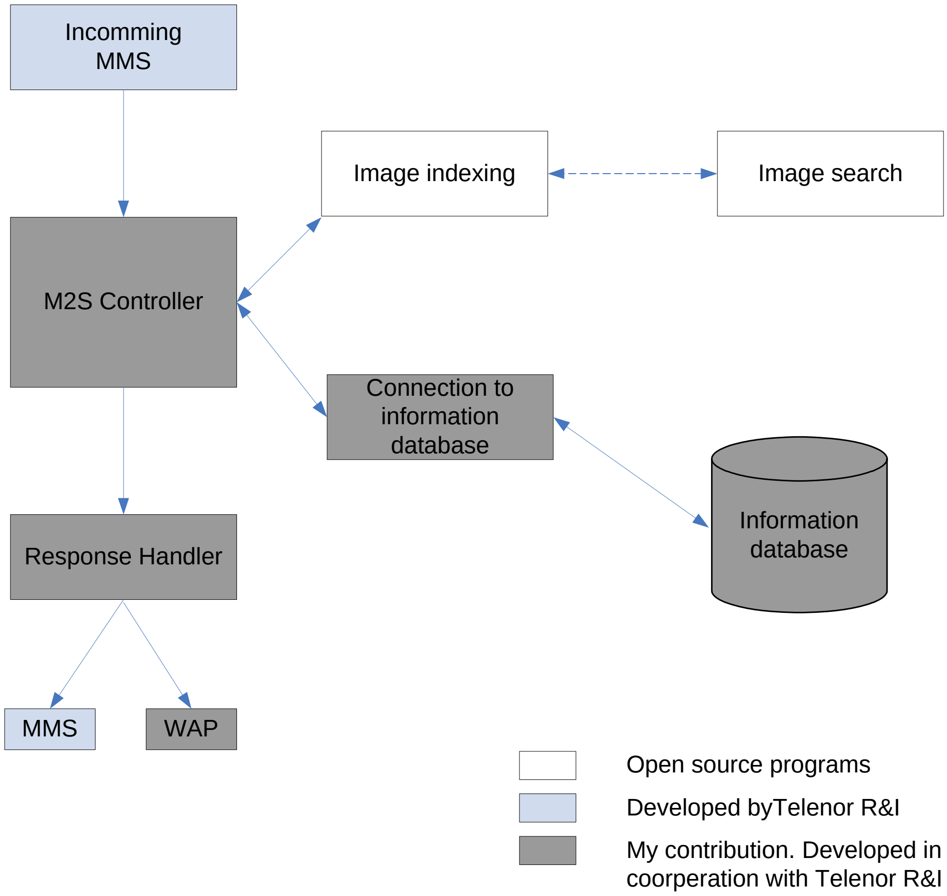


Figure 1: M2S architecture [15].

Figure 1 illustrates what the different participators have contributed with. The dark grey shaded boxes represent the components of M2S that were developed as part of this project. The blue shaded boxes illustrate what others at Telenor R&I have developed [15] and the white boxes illustrate an open source program that we have utilized for image indexing and searching. M2S is fully described in chapter 5.

- CAIR is a mobile application that enables users to retrieve images from image collections based on context and using SMS. The context in this thesis is location and time, it is formulated as text and the query is sent to the service. The system has not been implemented due to time restrictions but we will present a detailed design overview and specifications of the service.

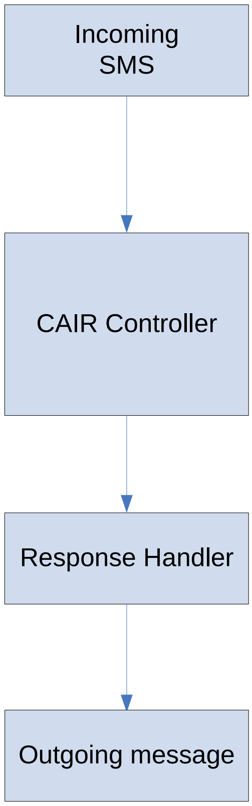


Figure 2: Illustrates the progress in CAIR.

Figure 2 illustrates the process flow in CAIR. For more details see Figure 35. CAIR is fully described in chapter 6.

- We will contribute with a study of different content-based image retrieval services. In Figure 1 the white boxes of M2S illustrate an open source image retrieval function that we have utilized. Before deciding which image retrieval function to use, a study considering several such functions was carried out. This study can be viewed further in subsection 5.5.1.

- We will give a summary of theory that is used in developing M2S and CAIR. The theory includes motivation for M2S and CAIR, discussions about techniques and concepts that we utilize and important concerns to remember when designing for instance for mobile environments. This theory can be viewed in chapter 2. A description of systems that relating to M2S and CAIR is presented in chapter 3.

1.8 Outline of thesis

Chapter 2 reviews the background material and theory for this thesis. The background material is mainly about images and image management, image annotation, traditional image retrieval techniques, the semantic gap and ways to measure the performance of image retrieval techniques (precision, recall and F-score). Chapter 2 also includes discussions about context, how to use context for image management and how to use context in mobile settings.

Chapter 3 gives an overview of the systems that relate to our work. These systems are important references when designing information retrieval services for mobile environments.

Chapter 4 gives an introduction to the services presented in chapter 5 and 6. We will give a short description of the M2S and CAIR and present a scenario where the services are suitable.

Chapter 5 describes the content-based image retrieval service M2S in detail. We present a prototype of a working system. We discuss advantages and disadvantages and we propose alternative solutions of the service. This chapter also contains an evaluation of a selection of content-based image retrieval systems that were considered for use in M2S.

Chapter 6 describes the context-based image retrieval service CAIR in detail. We present a design overview of the service and discuss advantages, disadvantages and alternative solutions of the service. The chapter 6 will also include an outline for future work for CAIR.

Chapter 7 gives a summary of this thesis and evaluates the services discussed in chapter 5 and 6. Chapter 7 will conclude this thesis.

2 Background

This chapter presents the background material for this thesis and together with chapter 3 forms the foundation of the work presented in chapter 4, 5, 6 and 7. M2S and CAIR are based on the theory concerning images, content-based image retrieval, text-based image retrieval, context and context-awareness, image annotation, The Semantic Gap, precision and recall, relevance feedback and theory concerning mobile environments. These concepts will be described here.

2.1 Introduction

People collect images for archiving memories, story telling and personal enjoyment. There are several ways to organize an image collection of paper images. Images are normally organized in wallets, in photo albums in a chronological order, or, for the not so organized ones, images are put in shoeboxes. Today most images do not have a paper copy. The introduction of the digital camera enables people to own large digital image collections and these may become hard to organize in a way that makes it easy to locate specific images. The images are stored digitally on computers and that introduces new matters of concern. Computer based systems to store these collections, facilitating future browsing and retrieval, will therefore become increasingly important. The images can be organized by time, location, events or any other way. Consumers often organize their photos in terms of events both for browsing and retrieval, as well as for sharing selected photos with others [16] [17] [18].

An event is what is happening in the picture, what the picture is about. Events are naturally associated with specific times and places such as a child's birthday, a vacation or a party. However, events are difficult to define quantitatively or consistently. The photos associated with an event often exhibit little coherence in terms of both low-level image features and visual similarity [16]. As an example of an event, consider pictures taken during a holiday trip to Venice. The photos can show different persons, buildings or vehicles. Just imagine how major the differences in the images taken inside and outside of a building can be.

It is easier for the human mind to remember a specific happening like a trip than it is to remember the date the trip was made. However, users seem to remember roughly the date and start browsing the collection up and down with this date as the starting point [16] [19].

The application areas that consider image retrieval as a primary activity are both numerous and disparate. The applications include medical informatics, digital libraries, document image processing and office information systems, remote sensing and management of earth resources, geographic information systems and cartographic modeling, interactive computer-aided design and computer-integrated manufacturing system, scientific databases, law enforcement and criminal investigation, and home entertainment. Because of the diversity in the image retrieval application areas, the features in the existing image retrieval systems have essentially evolved from domain specific considerations [4].

There are several techniques for image retrieval. Some techniques base their retrieval on the metadata of text that physically surrounds the image. Other techniques base their query on the visual content of the image. There are also systems that base their query on the image context. In the next sections we will discuss these techniques and as we will see they have several advantages. We will also see that the techniques have some drawbacks and shortcomings.

2.2 About images

Information can be represented in different ways, as sound, text, symbols and images. Users can hold large amounts of information and it would be a great advantage to have efficient technique in order to retrieve the desired information. The complexity of the information retrieval depends on how the information is represented. To search for a specific piece of text is far more efficient than search for a specific piece of sound based on the text and/or the sound itself.

2.2.1 Image annotation

People seem to use very little time to annotate their personal images. How many amateur photographers are determined enough and have enough time and energy to go through developed pictures, and put them into albums, instead of just sticking the pictures in a shoebox? How many people go through their digital photos and give each one a unique file name in an appropriate directory instead of leaving them in the default directory created by the camera software? Not many [9]. As a result, more and more people have thousands of digital photos with little or no organization, and they are resigned to gaining no more benefit or enjoyment from them than the photos stored in overfilled shoeboxes around the house. Well-performed annotation has the power to transform this almost random collection of images into a powerful, searchable and rich record of events in people’s lives [20].

There are two kinds of information associated with a visual object, which can be either image or video: Structured information about the object, called its metadata, and information contained within the object, called its visual features. Metadata is information connected to the object and can consist of digits and letters that are also referred to as text. It can also consist of sounds sketches or drawings. Visual features are usually automatically extracted from the image. These features are usually size, color, shapes and sketches [21].

2.2.2 What is annotation?

The goal of annotation of images is to assign semantically meaningful information to images. Text is the most common and relevant way of annotation [20]. An annotation can be for instance: “The celebration of Helen’s 25th birthday”. Other ways of assigning information can be by drawings and sketches where these are used as basis/input in searches where the image searched for and the sketch/drawing looks alike [22]. A recording of audio works the same way. An audio file is assigned the image and when the image is searched for, the input can be in form of singing, humming or whistling into the microphone [20].

Metadata may be used in a number of ways:

- Embedding the metadata in the web page using META-tags in the HTML coding of the page.
- As a separate HTML document linked to the resource it describes.
- In a database linked to the resource. The records may either have been directly created within the database or extracted from another source, such as web pages [23].

Time is an important factor when it comes to image annotation. As times goes by, humans forget what the image is about. This specially applies for images that are hard to identify without having other images of the same context to compare it against. This is also a strong argument for annotation and also a strong argument for doing it right away. There exist several standard for organizing and storing metadata.

- MPEG-7 is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group) for describing the multimedia content data in a broad range of applications [24].
- Dublin Core (DC) is another standard to describe the metadata for digital objects to increase their visibility, accessibility and interoperability [25].
- EXIF (Exchangeable Image File Format), and is a standard used in most of today’s digital cameras for storing metadata in image files [26]. The metadata stored is typically date, time, shutter speed and ISO sensitivity.

2.2.3 Methods for image annotation

There are several ways to annotate image collections. This can be either done manually, semi-automatic or automatic.

2.2.3.1 Manual image annotation

This is the “old-fashioned” approach where people have non-digital paper-pictures in photo albums and write the associated text. Manual annotation is a completely human oriented task that deals with human oriented information. This type of metadata can be the event of the image, the photographer, the title and similar information. The advantage of manual annotation is the accuracy in extracting semantic information at several levels. It is the most precise way of annotation and for now, the only way of full value to add semantics to images.

Manual annotation is manageable for small image collections, but for larger image collections it is far too time consuming to annotate each single image in the collection and this is the biggest disadvantage of manual annotation [20] [17]. The investigation done by Rodden [17] shows some of the users' behavior regarding their personal digital image collections. Images are downloaded from the camera, labeled with a software-generated name and placed in a folder. The name automatically generated by the camera software most often consists of letters and digits that do not have any semantic value. Most users do not interfere with the software's decisions then and to change the name of the images later on, is a task that is most often not carried out.

Another drawback is that the task of describing the content of images is highly subjective. The perspective of textual descriptions given by an annotator could be different from the perspective of a user. A picture can mean different things to different people. It can also mean different things to the same person at different times.

> A picture is worth more than a thousand words!

Even with the same perspective, the words used to describe the content could vary from one person to another. In other words, there could be a variety of inconsistencies between user textual queries and image annotations or descriptions [27]. To be able to compose a query that will result in relevant images, the annotator and retriever must have some common vocabulary and a common understanding of the world. If the annotated text and the query-text are completely different this might return no relevant results even if they potentially exist. Based on the work of Kustanowitz et al. [20], we believe it is naive to think that users will manually annotate large image collections if they are given other options [20] [28] –and even if they are not!

2.2.3.2 Semi-automatic image annotation

Semi-automatic annotation is manual annotation with machine extraction of some information. It depends on the user's interaction to some degree. The technical information (see automatic annotation) is added automatically from, for instance, a camera, the user can then be prompted to add additional information to the image. The manually added information is typically semantic information. [29]. Semi-automatic annotation combines the efficiency of automatic annotation and the accuracy of manual annotation of images. Human interaction can also provide an initial query or feedback during the annotation [20] [30]. Wenyin et al. [30] describe a semi-automatic image annotation process that combines content-based image retrieval and user verification to achieve correct high-level metadata. Semi-automatic image annotation carries the advantages and disadvantages of manual image annotation and, as we will see, it also inherits the advantages and disadvantages of automatic image annotation.

2.2.3.3 Automatic image annotation

Automatic annotation is machine annotation, where humans only verify the task. The information added by a camera is of a technical nature and is automatically added. This information is typically time, location, resolution of the image, camera model, which number the image has in the range of images taken, name of the image and other technical information. As we see from this type of information automatic annotation is limited due to computers lacking ability to extract semantic information from images. Even in an ideal world where face recognition and shape detection works perfectly, a computer will not be able to abstract event information like “The celebration of Helen’s 25th birthday” or other deep semantic information [20]. There are several situations where the images are automatically generated and have minimum of information attached. A surveillance camera may take series of photos and store them in a database without any human interaction. The image might be stored in folders annotated with the actual date. Specific images from a specific time will then be impossible to retrieve without browsing the image collection. To annotate each image in such a collection would be useless.

2.3 Traditional techniques for image retrieval

In the next subsections we will discuss traditional techniques for image retrieval. The techniques that we will describe are text-based image retrieval, content-based image retrieval and a variant of content-based image retrieval called region-based image retrieval.

2.3.1 Text-based image retrieval (TBIR)

The most common way to manage large image collections is to store text in the form of keywords together with the image. We separate between text based image retrieval techniques that use the surrounding text of the image and text based techniques where each image or image collection is annotated. The approach that deals with surrounding text searches the keywords that are physically close to the image. Search engines that use this techniques are Google [31] Yahoo [32] and altavista [33]. This way to retrieve images is based on the assumption that the surrounding text describes the image. The technique relies on text surrounding the image such as filenames, captions and the “alt”-tag in HTML and paragraphs close to the image with possible relevant text. A problem with these techniques is that the search engine considers an image relevant because it is annotated with a specific keyword. A relevant image might be left out due to the lack of specific keywords. There are situations where there might be no relevant text surrounding the picture. In fact, there might exist web pages where the surrounding text has nothing to do with the image. In these cases the returned results might be irrelevant and have nothing in common with the requested image.

The other approach uses image annotation of the images and is often a manual task. Annotation of images lets the provider annotate the image with the text (metadata) that is considered relevant. The text can be time, event, location, participants or whatever the user finds relevant. The largest drawback with this approach is that is it very time-consuming to annotate every image, and it is therefore most often not done.

To retrieve images the user constructs a query consisting of the keywords that describes the desired image. There are advantages of text-based image retrieval. They are:

- It is the only way to search for the semantics of the image. It is possible to perform queries like “Honeymoon to Bangkok”. It is of course a premise that the image is annotated with such text.
- It is the most commonly used technique for image retrieval and it is easy to construct queries. There is no need for tools for drawing, audio recognition or other advanced tools for constructing queries.
- The retrieval is fast. Sting matching is a relatively resource-friendly task.

There are also disadvantages of text-based image retrieval. They are:

- Many text-based image retrieval techniques base the comparison on exact string matching. If the query string is misspelled there are no results returned.
- It is dependent on images being annotated. If there is no text associated with the image, text-based image retrieval cannot assist retrieving images.
- In large image collections where images are automatically annotated with a minimum of information (for instance from a surveillance camera), it is hard to retrieve one specific image from the collection without some browsing.
- The retrieval depends on the image annotator and the image retriever sharing some common vocabulary and language. The general user’s usage of the system will be very limited since they are not skilled in manual annotation of the images with appropriate information.
- The use of synonyms would result in missed results that would otherwise be returned.
- A single word can mean radically different things and the result might consist of a mix of concepts that are lexically identical but otherwise different.

Critics of text-based approach argue that to have as correct image annotation as possible, this part must be automated. As we discussed in the section above, the automatic annotation is limited due to its lack of extracting semantic information from the image. Only automatic annotation of images in combination with pure text-based image retrieval will in many cases prove inadequate. The available metadata will be limited to the technical information surrounding the image, such as camera model, time, resolution of the image and name of the image. Critics of text-based image retrieval argue that users may find it difficult to use text to perform a query for some portion of the content of an image. Text-based image retrieval techniques ar naturally limited to search the metadata that is tagged to the image. If the text queried is not annotated the image, the image is not returned. This means that if a particular piece of the image is interesting this must be explicit included in the metadata. If the desired object is not a main part of a main happening in the picture, this is often not described in the metadata and cannot be a result from a query describing such portions of the image. If a user wants an image of a vase this is most likely not to be found if it is not the main happening in the image.

One of the disadvantages of text-based image retrieval is that a word can have different meanings. This problem is best illustrated with an example. Searching any news database for ex-president Gerald Ford would return many hits about the Ford Motor Car Company. While everything but the name is different, the two concepts are lexically identical. The search engines don't have reliable ways to separate the concepts. These problems are present even in systems with automatic synonym lists or thesaurus capabilities [34]. There exist several text-based image retrieval services today, where Google [31] is a large player.

Google Image Search [31] uses text based image retrieval. A crawler (also referred to as a spider or robot) follows all links on the web and finds keywords. These keywords are indexed in a database and when a query is performed using the search engine's web browser, links to the web pages containing the images are displayed as a list. The results are sorted on relevance and how popular the site is. To use Google Image Search, the user submits a text-based query against their database. The results are in form of a thumbnail of an image and a link to the site where the original image is situated.

2.3.2 Content-based image retrieval (CBIR)

Much of the motivation behind content-based image retrieval is for users to avoid annotation of images. In some systems text based annotation and retrieval of images is inadequate. It might include images from surveillance cameras, fingerprint recognition or x-ray pictures. Motivated by the lack of an efficient image retrieval technique, content-based image retrieval was introduced. "Content-based" means that the technology makes direct use of content of the image rather than relying on human annotation of metadata with keywords. From a user's perspective an optimal content-based image retrieval function would allow what is referred to as semantic retrieval where the users can perform query like "find images of a horse" or "find images of Mother Theresa". This kind of task is very difficult for a computer to perform. A Shetland pony looks very different from a Friesian and Mother Theresa does not always face the camera in the same pose. However, if the query is "horse" it might be irrelevant if the image is of a Shetland pony or a Friesian.

Current content-based image retrieval systems make use of low-level features to retrieve desired images from image collections. These image features are normally basic image information like shapes, color and texture. To have efficient image retrieval, tools like pattern recognition and statistics are well used. Different implementations of content-based image retrieval functions make use of different types of queries. There are three main approaches in content-based image retrieval [1]:

- Query by example. The user selects images and the image retrieval system returns a selection of images based on a set of matching points.
- Specifying colors.
- Sketches/drawings. The user creates a rough sketch to find matching images.

Some systems use a combination of these features. Octagon [35] lets users manually adjust to which extent the color should play a part.

To use content-based image retrieval is a big challenge not only to the system designer but also to the users. For the search to return good results the user must have an idea of what the particular image or scene looks like. The users submit a rough sketch/painting of the required image and the results returned are images that have been ranked in terms of the similarity with the submitted sketch/painting. Content-based image retrieval requires users to adopt new and often complex and challenging search strategies based on the visual aspects of images rather than a semantic description of the image's content. The search procedure must have enough information about the indexed pictures to match the query, but it should not straighten the requirements so images that should be a part of the result are not considered [4] [36] [37].

The object on the pictures of Figure 3 is the Merlion in Singapore. The pictures are all taken from different angles and distances. Picture number one and two are relatively similar and a content-based image retrieval system should be able to match these. Picture number three is taken from the backside of The Merlion and it is less likely that a content-based function would be able to detect that this is also The Merlion.

Figure 3: Three images of the same object [38].

In Figure 4 it is very hard to determine what the user is focusing on. At first glance the picture can in theory be of the skyline of any city in the world. At second glance it is possible to see The Merlion by the sea. If the user wants images of The Merlion or the skyline of Singapore this is not a good query image. The Merlion is hardly visible and the probability that the result image will be of the Singapore skyline is relatively small. However, if the user wants a picture of any skyline it can result in a good match.

Figure 4: The skyline of Singapore with The Merlion less striking [38].

Figure 4 shows the skyline with the Merlion less striking than the former image. In fact, the Merlion can hardly be seen at all. In such cases where the primary object of the image is hardly visible it is hard to find matches among the images. Systems that are based on content-based image retrieval include QBIC [2], imgSeek [39], VisualSEEk [40] and GIFT [41]. There are several advantages to content-based image retrieval [36] [4] [37].

- It may be possible to search for objects in the picture that is not part of the main activity of the image.
- The user can select portions of an image, submit this as a query and receive images that look like the partial picture.
- Annotation of the image is superfluous. In a purely content-based approach text is not used.
- If the user has an image and want to find similar images, this image can be submitted as a query-by-example. This is an easier approach than make the user describe the content of the image as text and submit the text as query.

There are also disadvantages to this approach:

- Indexing of large image repositories is time and resource consuming. A major limitation of content-based image retrieval systems is that they are limited to relatively small databases.
- It is not possible to search for the semantic of the images.
- Tools to construct and/or submit a query image can be complicated to use.

QBIC [2] is an image retrieval system that lets users find pictorial information in large image and video databases based on their visual information like color, shapes and sketches. QBIC allows queries on large image and video databases based on:

- Example images. The user searches with a query image. The image can be provided by the user or chosen from a selection of images. The software finds images similar to it based on various low-level features.

- User constructed sketches and drawings. The user draws a rough sketch of the image they are looking for and the software retrieves images whose layout matches the sketch.
- Selected color and texture patterns.
- Camera and object motion.
- Other graphical information.

During the population of the database the images are processed to extract the features that describe the image. These describing features can be color, shapes, textures and camera and object motion and these are stored in a database. Videos are broken up into clips and representative frames are generated for each shot. These frames are processed the same way as images. To retrieve results from the database the users formulate a graphical query. Features are extracted from the query and compared to the features describing the images to find similarities [2].

2.3.3 Region Based Image Retrieval (RBIR)

Region Based Image Retrieval (RBIR) is an extension of content-based image retrieval techniques [42]. Region-based image retrieval systems provide new query types to search for objects embedded in an arbitrary environment. An RBIR system automatically segments images into a variable number of regions, and uses a segmentation algorithm to extract a set of features (like color, shapes and sketches) for each region. A function determines the differences between the database image and a set of reference regions. Unfortunately, the large evaluation costs of the dissimilarity function are restricting region-based image retrieval to relatively small databases [42]. The most crucial part of this approach is the segmentation of the regions. There exists several techniques to do this, but to describe the different techniques are beyond the scope of this thesis. Systems that use region-based image retrieval are Blobworld [43], WALRUS [44] and SIMPLIcity [45].

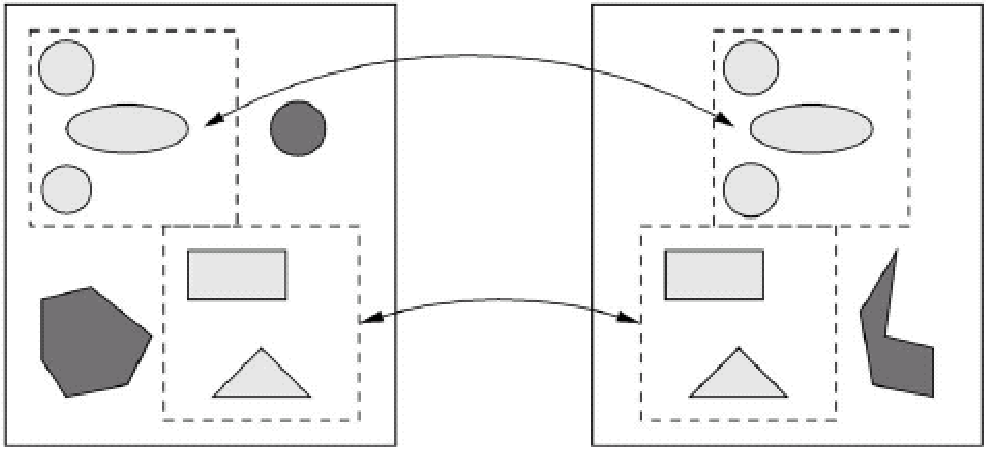


Figure 5: Images with similar regions [44].

2.4 Relevance feedback

Some image retrieval systems, both text-based image retrieval and content-based image retrieval provide relevance feedback [46]. Relevance feedback refers to the feedback from a user on specific terms regarding their relevance to a target image. These terms can be both textual information for text-based image retrieval or sketches/drawings and image examples in content-based image retrieval. Relevance feedback is an important concept in interactive information retrieval. The idea is that based on the feedback from a user during the retrieval process about the previously retrieved objects, the system can adjust the query to better represent the user's information needs [27]. Rui et al. [46] argue that relevance feedback is a way to narrow the gap between high level concepts and low level features that causes the semantic gap. They also argue that relevance feedback in some degree tries to solve subjectivity of human perception of visual content.

2.4.1 Relevance feedback for TBIR

To alleviate the problem related to the inconsistency in vocabulary described in the section for manual annotation and text-based image retrieval, different strategies for relevance feedback can be potentially applied. For general information retrieval systems a strategy utilized by CiteSeer [47] is frequently applied. The strategy provides users with alternative search strings when the performed search was fruitless. If the search string is "mobile context information retrieval" returns zero results, the search engine comes suggestions of how to improve the search and provides a list with alternative search strings using the proposed features. This was the suggestion made by CiteSeer:

Use "or" to separate alternatives.
If searching for an author try using only the last name.
Adjacent query terms default to one word proximity (words must occur next to each other).
Suggested query: information
Suggested query: mobile or context or information or retrieval
Suggested query: mobile and context and information and retrieval

A way to provide relevance feedback for text-based image retrieval [27] is to show textual descriptions of similar images to narrow the user's query. Lycos [48] use this approach. A test with Lycos shows that the query "two friends" result in a list with textual descriptions of sub-queries that will narrow the search:

- Friends at School
- Two Best Friends
- Two Friends Talking
- Friends Playing Basketball
- Cartoon Friends
- Two Friends Clipart
- Poems on Friends
- Old Friends
- Male Friends

2.4.2 Relevance feedback for CBIR

Relevance feedback is more often used with content-based image retrieval than text-based image retrieval. Content-based image retrieval systems often support relevance feedback. Relevance feedback for content-based image retrieval means that the user can mark the results of the query as "relevant", "not relevant" or "neutral" before the query is performed again with the new information [37]. This is a way for the system to learn and to personalize the answers. The query with the new information is resubmitted and improved results are returned. Content-based image retrieval systems that uses this approach are GIFT [41], PicHunter [49] and MindReader [50].

2.5 The semantic gap

The biggest challenge in image retrieval is to bridge the semantic gap. Users often have an abstract idea of what image he wants to find. The problem it to construct a query that the computers understand and provide users with images matching his notion. The low-level features of the images are easily measured and computed, but the queries given by the user is based on high-level concepts that describe what the images are about. The problem is to translate these high-level concepts to low-level features as illustrated in Figure 6. The lack of this translation results in a semantic gap [3] [51].

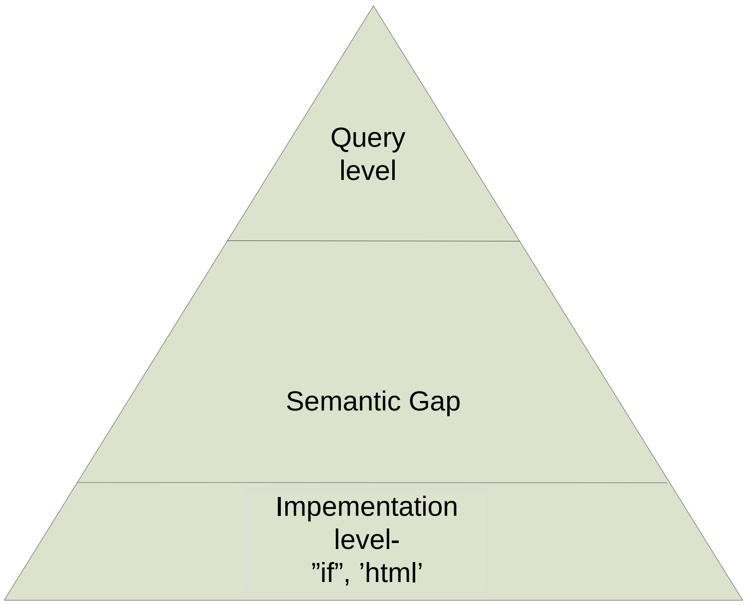


Figure 6: Illustrates what is meant by The Semantic Gap [52].

It is said that a picture is worth more than a thousand words. Image content are much more versatile compared to text, and the amount of visual data is already enormous and still expanding very rapidly. The search engines cannot completely satisfy queries from the user, and therefore the retrieved result is often thousands of pages containing little relevance, or thousands of “hits” known as information overload. Most search engines find matches based on keywords, they do not consider their meanings [3] [46] [51].

2.6 Precision and recall

The next subsections will ways to measure the performance of information retrieval functions.

2.6.1 Image retrieval

Image retrieval is a subclass of information retrieval and inherits therefore many of the aspects that encompasses information retrieval. Image retrieval is concerned with retrieving images that are relevant to the user’s request from collections of images. The essential aims of information retrieval are to be *efficient* and *effective*. Efficiency means delivering information quickly and without excessive demands on resources, even when there is a massive amount of information to be retrieved. Clearly efficiency is extremely relevant to information retrieval where late response is often useless information. Effectiveness is concerned with retrieving relevant documents. This implies that the user finds the information useful. If a user keeps retrieving information of low relevance it is natural to believe that the user quit using the system [9].

An important thing to have in mind when testing the effectiveness of image retrieval systems is that the similarity criteria can vary from user to user. What is adequate for one person might not be adequate for another person. We still have some methods for deciding what is relevant and what is not. Information retrieval is often defined in terms of the location and delivery of documents to a user according to the information needed. This seems to be very easy, but has proven to be a very complex task. Take the Internet for instance, it is very easy to publish information, but it is difficult to find relevant information. Search engines returns a set of results. The results are most often a combination ranging from relevant results to irrelevant results. It is common to evaluate the performance of information retrieval system and there exist several techniques to achieve this. The most common techniques are called recall and precision [12]. Recall is the proportion of the number of relevant items retrieved of the total number of relevant items available. If a search has high recall it refers to the experience of getting thousands of hits, which is also known as information overload. Precision is the proportion of all relevant items of all the items retrieved. Search engines focuses on their software's ability to retrieve high amount of hits and focuses less on the relevance of the pages. A test using Google Image Search [31] shows that if you use the word "cat" you get 3 080 000 hits. Who goes through that many hits to find a particular image?

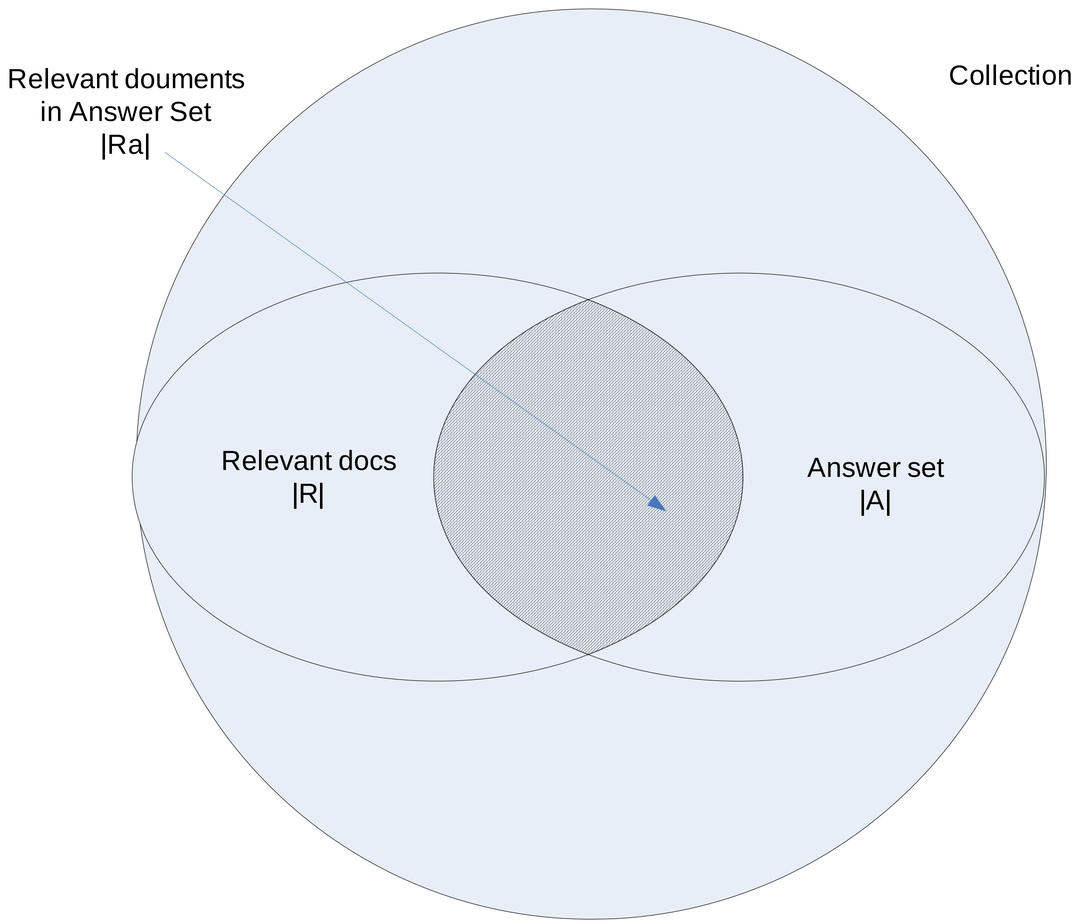


Figure 7: Illustrates precision and recall [12].

Consider an example of an information request I that is a collection of information. In this collection of information there is a set of relevant documents, R, where |R| is the number of documents in the set. Assuming that an information retrieval strategy processes the information request I, and generates an answer set A where |A| is the number of documents in the set. Let |Ra| be the number of documents in the intersection if the sets R and A. These sets are illustrated in Figure 7. The recall and precision are measured as follows [12]:

- Recall is the fraction of the relevant documents (the set R) that has been retrieved.

$$\textbf{\textit{Recall}} = \frac{|Ra|}{|R|}$$

- Precision is the fraction of the retrieved documents (the set A) that is relevant.

$$\textbf{\textit{Precision}} = \frac{|Ra|}{|A|}$$

Precision and recall are tested on a fixed and relatively small set of documents and has predefined queries to decide which documents goes with which queries. This is not a task that is done completely automatic. To construct the predefined set of matches is a manual and highly subjective task. Precision and recall measurements might therefore not be as correct as desired [12]. Because recall and precision are not always the most appropriate measures for evaluating information retrieval performance other approaches has been proposed. The precision and recall scores are often combined into a single measure of performance, known as the F-score. F-score is the weighted harmonic mean of precision and recall. F-score gives an average score of the efficiency of the system without specifying the value of precision and recall [12]. The formula for F-score is:

$$\textbf{\textit{F-score}} = \frac{2 \times precision \times recall}{(precision + recall)}$$

The F-score is also known as the F_1 measure, because precision and recall are equally weighted. There is also F_{05}-measure that weighs precision twice as much as recall and measure F_2 that weights recall twice as much as precision. The idea is to allow the user to specify whether he is more interested in recall or in precision. This value is often given in percentage and the higher the percentage, the better search strategies. All three measures consider documents as either relevant or irrelevant. In practice documents can have degrees of relevancy. In this thesis we refer to precision and recall to determine the performance of retrieval systems.

2.7 Context and context awareness

The next subsections will discuss context, image context, context-awareness and the role of context in information retrieval. We will then discuss time and location as context.

2.7.1 What is context?

Context is a general term used across many different fields. Dey [53] describes context as situational implicit information that helps the communicating parts to understand what is said based on the setting the parts are situated in. Said in other words, context is everything about the situation that is relevant to the application and its users.

When humans interact with each other we are able to use this implicit situational information described above to increase the conversational bandwidth. Humans have a common understanding of how the world works and don't have to describe a situation down to every detail. This concept can be easily explained with an example. If a person leaves a room and another person asks him to shut the door, it is tacitly understood that it is the door closest to them he want to get shut and not some other door far away. The closest door is a part of the situation and is therefore context. The remote door does not affect the conversation (situation) and therefore it is not a part of the user's context. This concept does not transfer well in human-computer interaction and that makes context a poorly used resource in traditional computing environments. The current lack of efficient techniques to provide input to computers makes them unable to take full advantage of this rich source of information. By strengthening the computers ability to use context we will enable computers to communicate in a richer way and make more useful programs in the future [6]. Hofer [54] has a distinction of different context dimensions, physical and logical context. The physical dimension refers to context that can be measured by hardware sensors such as temperature, light conditions, movement, touch, sounds and air pressure. The logical dimension is mostly specified by the user or captured by monitoring user interactions/behavior like the user's goals, tasks, work context, business processes and the user's emotional state.

2.7.2 Image context

We consider pictures to be associated with multiple contexts. An image can have two types of context associated with it, static and dynamic. Static context often refer to context that can be measured by hardware sensors, e.g. location, time, light, sound, movement, touch, temperature, humidity and air pressure. Static context is information added to the image when the image is created/captured. This context never changes in time. After the image is taken, there is no way to change e.g. the location where the image was taken. Because the nature of static context is objective, image retrieval based on static information is relatively easy. By objective we mean that this information is perceived equally by most people. Most context-aware systems make use of static context as it provides very useful data, such as location information. The other type of context is called dynamic context. This context is added after the image is taken and can change in time, situation and it is dependent of who the viewer is.

An example of dynamic context is an image of the leaning tower of Pisa. The tower could be seen on an illustration of Italian old architecture in tourist guides in Italy. The same image can appear in the syllabus for students in construction class as an example of what consequences too soft soil can have in building constructions. The very same picture can also appear in a science book. Galileo Galilei is said to have dropped two cannon balls with different mass from the leaning tower of Pisa to test his hypothesis that their descending speed was independent of their mass. This shows that dynamic context is subjective by nature and events can be perceived different from person to person.

Context like date and time can be automatically annotated when the image is captured so the user doesn't have to be concerned about this. Time and date are automatically attached to the image by all digital cameras. Location is a different matter. Today, few digital cameras and mobile phones have the ability to automatically tag the location to the captured image. We believe that this is only a matter of time before all cameras are equipped with this feature.

2.7.3 Context awareness and context aware computing

Context-awareness is the ability to use context information. When we use context with applications, we call this context-aware computing. Dey [6] defines context-aware computing as:

"*A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.*"

Computers have until the latest years been relatively stationary. A desktop-server approach has been and is still the common way for communication between client and server. This also applies for the applications belonging to this hardware [55]. In the same way researchers have tried to specify the important features of a context-awareness, they specify the features of context-aware applications. Dey [6] proposes a categorization that combines the ideas from previous taxonomies and attempts to generalize them to satisfy all existing content-aware applications. There are three categories of features that a context-aware application can support [55]:

- Presentation of information and services to a user.
- Automatic execution of a service for a user.
- Tagging of context to information to support later retrieval.

Context-awareness has become an important consideration in the development of mobile, ubiquitous and pervasive computing. Schilit et al. [56] defines context-aware applications to be applications that dynamically change or adapt their behavior based on the context. Context-aware applications are applications that are able to adapt their behavior and without explicit user intervention, tries to increase the usability and effectiveness with the use of relevant context [56]. Context-aware systems offer new opportunities for application development and use by collecting context information and adapting systems behavior accordingly. To make such an information system work, the need to distinguish relevant from irrelevant information increases. Because this is a highly individual point of view, it becomes a difficult task. The success will depend entirely on their ability to filter the context and deliver the relevant information at the right place and at the right time. As context can rapidly change it is important with “quick responding applications” [55].

2.7.4 The role of context in information retrieval

Information retrieval has traditionally regarded users working inside a “world of information”, using desk-based computer systems. The development in mobile and wireless computing means that information retrieval systems are now embedded in the real physical world with technology such as sensors to perceive and provide rich information about the surroundings.

The use of context makes software developers able to provide users with personalized services. The application monitors the user’s context and reacts properly to it. The information can be provided to the user automatically, resulting in a minimum of interaction. We imagine users retrieving desired information without asking explicitly for it. The user will have quick access to the information or service that he needs in his current context [57]. The potential areas for context aware applications are many including travel information, shopping, entertainment, event information and different mobile professions. The more sources of context information, the more adaptive, user flexible and useful systems can be developed.

2.7.5 Time and location as context

Time and location are probably the most used context in context-aware computing. This context can be automatically added to images assuming that the camera has location detection functionality GPS (Global Positioning System) and clock.

To use time as context can be problematic. To set time on cameras is an individual and manual task and is it not required to set the time before images are captured so this is often not done. Even if the time is set when the camera is put to use, it can easily be wrong adjusted without users taking notice of it. Even if users set the time correct it has to be recalibrated once in a while. This means that the time set on images can be wrong and then be erroneous to use as basis for a search. Gurrin et al. [58] elucidates some of the problems using time as context. In most places in the world a photo taken at 10 am will most often be in daylight, but it is not always the case. In locations very far north or far south the location will be bathed in pitch darkness in wintertime. A photo taken at 2 am will imply darkness of night, but the locations mentioned above can just as well have midnight sun and daylight conditions.

The number of features that merges photo and mobile entities are increasing and this is currently a hot topic of developers at mobile phone companies. Nokia [59] has just launched a new series of mobile phones and they focus a lot on image and video management. Together with other features, their new model N95 has fully integrated GPS solution with complete maps of the world. The images will be automatically annotated with the GPS position where they were taken. We believe that users will discover the advantages with this feature and that Nokia's competitors like Sony Ericsson will follow this trend.

Figure 8: Nokia N95 with built in GPS [59].

Nokia N95 enables users to automatically annotate images with location information. It automatically codes the pictures with geographical information, storing latitude and longitude details in the EXIF information within the image file.

Location is the far most used context in context-aware computing. In contrast to other contexts, location can today be easily detected. There are different technologies like GPS, mobile phone towers, badge proximity detectors, cameras, magnetic card readers, bar code readers, Bluetooth, IR (Infra Red) and ultrasonic sound to detect users position [60]. Some of these technologies demand clear sight to the receiver and some can travel through walls. Location-aware applications utilizes the knowledge about the physical location of real-world objects such as persons and devices to adapt their functional behavior and their appearance to the user [61]. Location-aware applications can be developed for both indoor and outdoor use. For outdoor use, GPS is a handy way to localize users. For indoor use other approaches such as for instance IR-beacons are better suited.

2.8 Location aware applications

Location information is used in a considerable number of applications. Toyama et al. [62] describes a system that capitalizes geographic location tags on digital images. They argue that the location where the photo was taken is important because it says a lot about the semantics of the picture. This information also makes it easy to index and search though image collections. The geographic location where the photographs were taken provides useful context to the image.

Location is a universally understood context if it is presented properly. Location is tightly bound to the semantic of the picture. If the picture is taken on the top of Mount Everest the location says a lot about the picture without even displaying a single pixel.

CybreMinder [63] is a context-aware system for supporting reminders. CybreMinder is a context-aware tool that supports users in sending and receiving reminders. The delivery of the reminder can be triggered by the user's context such as time, location, situation or more sophisticated types of context. For instance, while a user is in specific situations (for instance cinemas or meetings) the reminder is not delivered before the user has left the situation he is in.

The Active Badge Location System [64] and The Marble Museum PDA application [65] are location-aware systems for indoor use. In The Active Badge Location System the members of the staff wear badges that transmit signals providing information about their location to a centralized location service, through a network of sensors. The system is meant to be an efficient way to keep track of employees in large companies like hospitals where the employees are often hard to localize [64]. There have been several attempts to develop hand-held guides for tourists visiting museums. GUIDE [66] and Electronic Guidebook [67] are such applications. The Marble Museum PDA application is a location-aware guide for visitors of The Marble Museum of Carrara in Italy. The visitor's guide uses automatic detection of a change in environment to trigger the presentation of multi-modal context-dependent information through a PDA. The application uses IR-beacons that localize the visitors in the museum [65]. In chapter 3 we will describe several context-aware systems more detailed.

2.9 Mobile environment and contexts

The emergence of new information retrieval techniques, such as those used in mobile computing, raises new challenges for information retrieval. It is argued that context is going to be an important component when solving these challenges [6] [7] [68].

Mobile applications make use of context in a new way. Information is now made available in situations it previously has not been available in. In "the old days" you had to watch the weather forecast before you left home, but today you can get updated information about the weather on your mobile phone or PDA. Context-aware systems dealing with location information are widespread and the demand for them is growing due to the increasing spread of mobile devices. We have just described several examples of location-aware systems, such as various tourist guide projects where information about the tourist current location is crucial for a successful service.

Some information is of a kind that makes it interesting only if it is delivered the user in a specific situation. Such information can for instance be weather conditions. Weather conditions three days ago is irrelevant to a user that is on his way to Mount Everest and wondering whether he shall continue to the top or not. This example illustrates that the user is embedded in the real world and will most often require updated information relating to their physical surroundings [68]. As the system CybreMinder [63] illustrates the user are often primarily engaged in other activities than information searching and is likely to want rapid access to relevant information to assist them.

It is challenging to develop context-aware applications as a user's context can change rapidly. The context-aware systems must collect and interpret the dynamic context information and adapt the behaviour accordingly. It's always possible to do image recognition that is better than nothing, but it is hard to do something that is perfect.

Mobile computing devices range from semi-portable e.g. laptop, to truly mobile devices like mobile phones and PDA's, though sometimes these devices are incorporated in a user's environment like for instance a car/driver computer. This development change how information retrieval systems are taken "outside the office". An outcome of this is that the users amount of nature and context increases. Sensors are becoming cheaper and more ubiquitous. If the mobile information retrieval system is equipped with such sensors, much physical context such as temperature and humidity, is potentially available. In these settings the users are often primarily doing some other activity, so it is likely that they want the information retrieval system to assist them improve the activity they are in. It is therefore crucial that the system's response is rapid and correct [68] [63]. The response might be in the form of an SMS, MMS or a program to run. We will however not discuss the display-issue in this thesis.

Users have now in much higher degree access to information where they want it when they want it. Here we can use context to help select the information needed. Location as context often plays a key role in context-based mobile applications. However, we believe that use of richer context including i.e. temperature, time, date and sound makes more suitable and wanted applications. A document, such as an image, may have contextual fields representing an associated context such as location and temperature. An outdoor café is for instance only suitable in certain temperatures [9]. This extra information does not necessarily need to be directly involved in the process of query, but can improve the quality of the material delivered to the user. The more an application knows about a user's context, the more likely is higher precision [10].

The introduction of information delivery directly to the user introduces another concept in information retrieval called *proactive information retrieval*, where a device may automatically perform a request to a search engine or may trigger information when the user enters a specific context. Another important issue when it comes to use of context in information retrieval within ubiquitous computing is to determine the manner and timing of any information passed to the user. There exist both hardware and software that can help determine if it is appropriate or safe to disturb the user at a given time in a given position. Since the information requested is based on the user's current context the information must be delivered in a timely fashion since the context can change. The information is often useless if it is delivered outside a context the user has just left [9] [63].

In context-aware retrieval systems the matching scores are very important. As well as being used for ranking, they can be used in deciding weather results should be delivered at all. For example a proactive system may decide that since the best matching results has a rather low score, it does not make sense to distract the user with it resulting that it is not delivered at all [69]. The score might indicate that the information is not relevant enough even if the user is available.

There are well-known problems and restrictions with mobile environments and these are important to have fresh in mind when we develop applications for mobile systems. Known problems are that they unexpectedly disconnect, they can be unavailable for periods and the systems are often very heterogeneous. An obvious restriction is the size of these devices. Handheld mobile devices are far less powerful than stationary computers, when it comes to CPU, memory, link capacity, the limited battery life and so forth. The screen is also much smaller and therefore limits the use of these entities in many ways e.g. search, browsing and information retrieval.

2.10 Summary

This chapter has given an understanding of the background material for this thesis. We have discussed ways to annotate images both manual, semi-automatic and fully automatic and we discussed advantages and disadvantages with these approaches. A picture can mean different things to different people and the manual expression of metadata becomes problematic when the images are searched for.

Traditional image retrieval techniques include text-based image retrieval and content-based image retrieval. There are several advantages and drawbacks to these approaches that we have discussed. Content-based image retrieval makes it possible for users to search for parts of an image that is not a part of the main activity of the image. Because content-based image retrieval techniques base their search only on the contents of the images, annotation is superfluous.

We also discussed that the lack of semantics creates a gap between what the user want and what he is actually capable to express in a query. This gap is called the semantic gap and is a main challenge in image management. The semantic gap leads to low precision and recall; it is frustrating to search for information, get thousands of hits, but few of them are relevant.

Until now computers environment have been relatively static. We discussed mobile technology has made its entry and full worthy applications must be made to support this trend. To develop more software that does not narrow or tighten the semantic gap will only provide another inadequate system to the users. We discussed the role of context in this situation and how it is predicted to support information retrieval in applications. Context-aware image retrieval is a technique which potential has not been fully discovered yet. We used Dey's [6] definitions to understand what context and context-awareness is.

3 Related work

This section/chapter relates our work to other research projects when it comes to image retrieval techniques and context-aware computing in mobile environments. The systems that we particularly want to focus on are systems that use location to derive information in a mobile environment.

It is not a goal for this thesis to give a full description of every system that somehow is similar to our system, but describe some of the systems that explores the concepts that we use. We will also give a short description of the limitation of these systems.

We describe general web-based photo management solutions. We will see how these solutions provide users with search and browsing functionalities but little more than this. We will then describe systems that take photo management to a further step and provide users with additional information. The systems that we first describe are context-based services that provide users with some information. These systems are Cyberguide, The Photo Navigator System and a context-based system for browsing personal digital photo collections. The system SnapToTell uses a combination of content- and context based retrieval to provide information to users. The systems described are not necessarily complete systems in the way that they are fully developed. They can be descriptions with respect to design, architecture and may be partly implementation. For every system described we will compare it to M2S and CAIR and discuss similarities and differences.

3.1 Web-based photo management solutions

There exist several web-based photo management systems such as Flickr [70] and Snapfish [71] that provides users a limited space where they can upload images or browse the image collection from any location. The services most often require the users to browse through the photo thumbnails or search using manually annotated images.

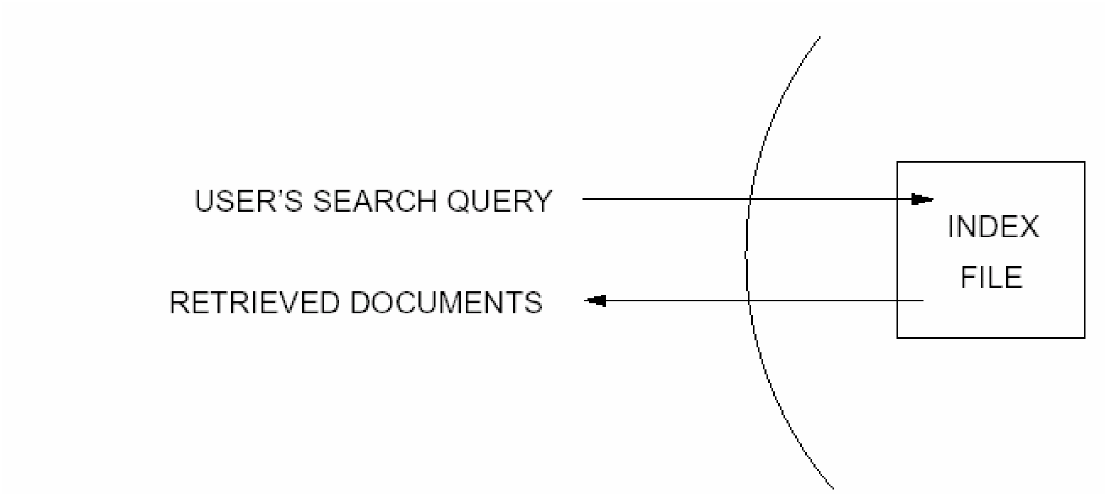


Figure 9: Traditional information retrieval scenario [72].

A we discussed in chapter 2 it is not advised to have a large number of images displayed on small mobile devices or rely on very extensive browse or search functionality. We have also discussed manual annotation of images and how this is a time consuming and error prone task.

Snapfish does not make use of content or context in any way. In Flickr the images can be tagged with location information and be placed on a world map based on this location information. It is then possible to the user in an easy and intuitive way to retrieve images based on the location where they were taken. The next systems that we describe make use of context like location or/and time in a greater degree to provide the user with information.

3.2 Cyberguide

The Cyberguide project [73] is a series of prototypes of mobile context-aware tour guides used for outdoor and indoor use. The prototypes differ in several ways, but we want to describe the core functionality and architecture.

The intention was to develop an intelligent handheld tour-guide that should be as good as a real tour guide. Museums could provide these to visitors allowing them to take personalized tours seeing any sight desired in any order. Knowledge about the user's current location and the history of his past locations, are used to provide a service that is normally expected from a real tour guide. Information about the sights is moved into the hand-held tour guides. Instead of having real tour guides each user is given a hand-held tour guide. As the user moves in the area he's provided information about the sight he is physically close to. The context (such as position and surroundings) of the user is displayed at the screen of the hand-held device. The service has automatic positioning of the users so they can be given correct information according to their context along their tour. The hand-held guide is also equipped with functionality for interaction. The user can ask questions like "Who painted that picture?" and get an immediate answer. The user can also communicate by sending e-mail, printing information etc. The Cyberguide architecture consists of four components. The components functions independently, but each component can vary from prototype to prototype. The four components are map component, information component, position component and a communication component.

The map component has knowledge about the physical surroundings, such as location of buildings or pathways that the tourist can access. This information is displayed to the user in a map. Visualizing and manipulating the map dominates the user interface of Cyberguide. It can be viewed at different levels of detail and scrolled around.

The information component provides access to information about sights that the user might encounter during their visit. This includes descriptions of buildings, who lived there, etc. This component also handles the user asking questions like the ones mentioned above. The information is displayed by a pen touch on the map or by walking up to the sight. The information is realized as a repository of information about physical sights.

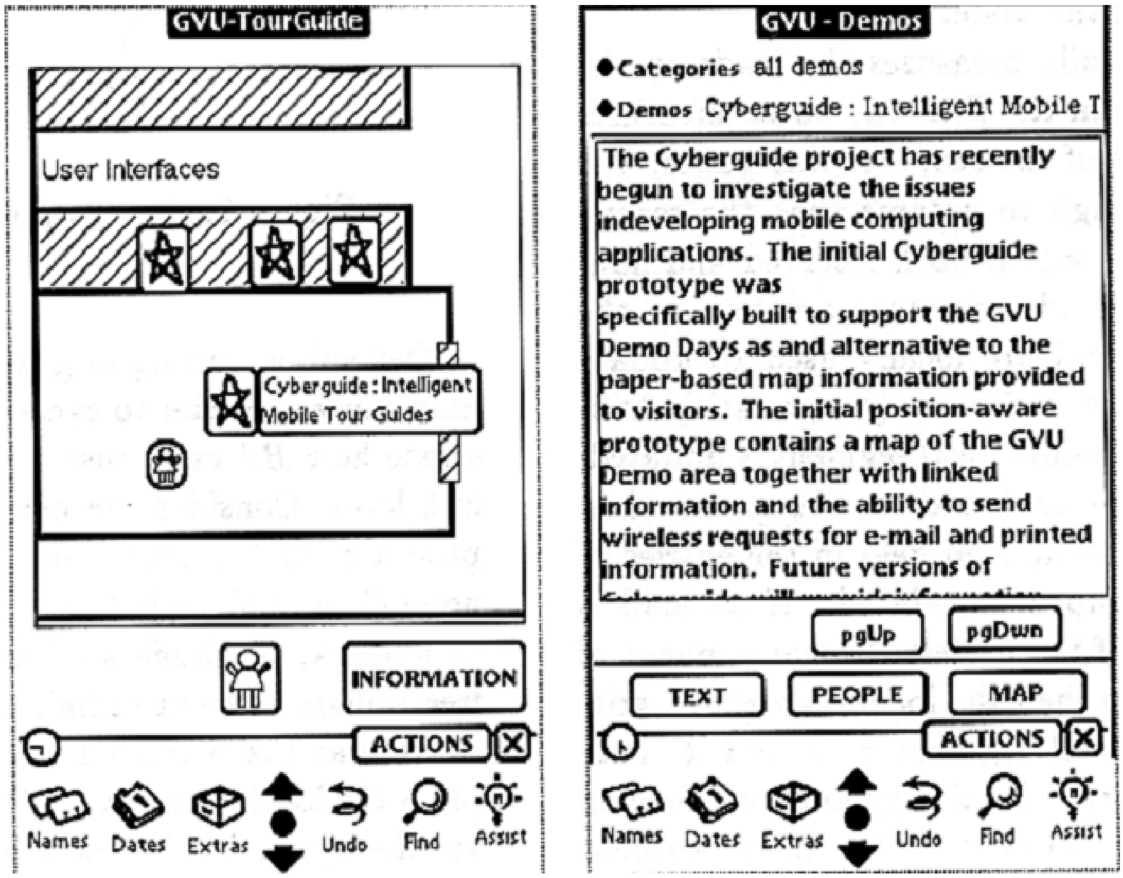


Figure 10: Map and information interface of the Cyberguide prototype [73].

Figure 10 shows the map, to the left, and information interface, to the right, of the Cyberguide prototype.

The positioning component is charting the position of the user within the physical surroundings. The component provides constantly updated information of the location and surroundings of the tourist. To provide a complete service it is important to know where the user is to display the correct surroundings on the map or give the correct answer to questions like “What am I looking at?”. Cyberguide bases the indoor location functionality on IR transceiver units. The area is divided into cells. As the tourist moves and passes into the range of a new cell, the position is updated on the map. For outdoor positioning they use GPS. Positioning information in the entire area is important, but without knowing the orientation of the user (for instance what someone is looking at), the physical position information of the user is of limited value. Rather than distribute a positioning system around the physical area, it is just as useful to collect detailed positioning information around sights of expected interest in the area.

The communication component is the wireless communication component that enables users to send and receive messages. Such information can be e-mails, printing documents, broadcasting of group messages such as “The bus will be leaving in 15 minutes". This component is realized as a set of wireless communication services like IR.

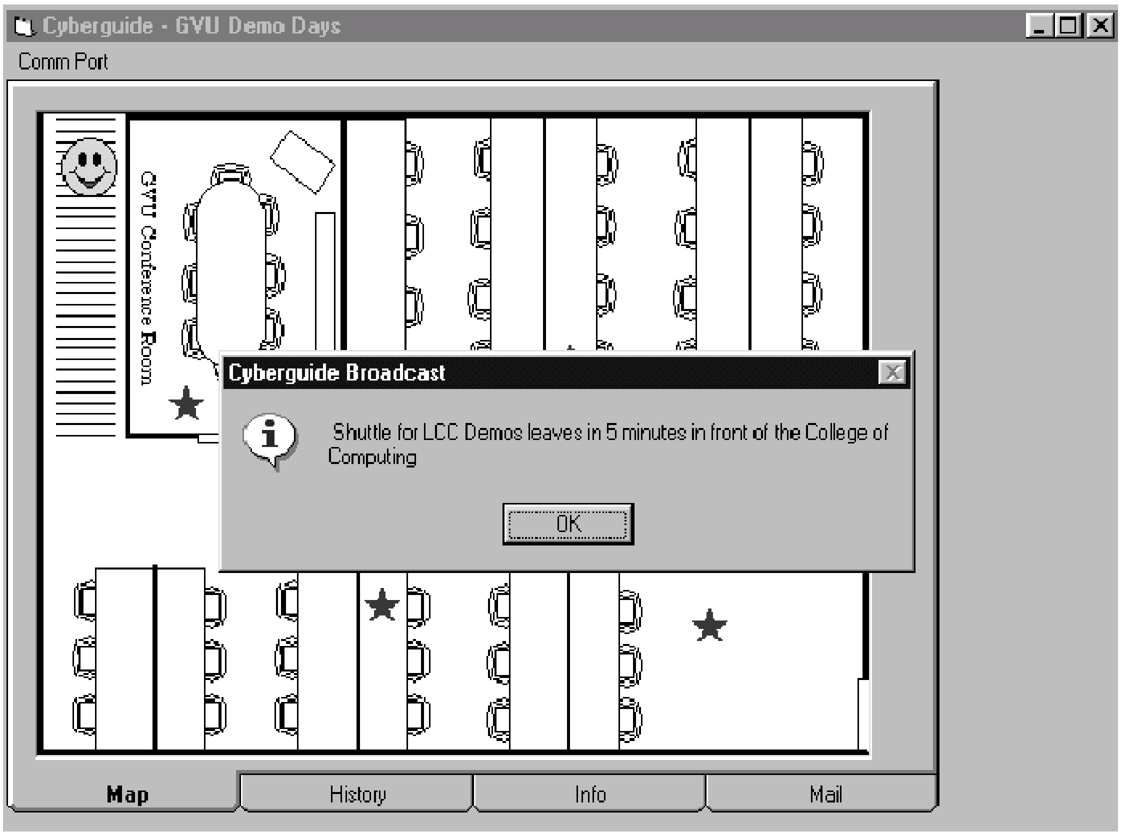


Figure 11: Shows a broadcast message to the individual Cyberguide units [73].

Figure 11 shows an example what a broadcast message from the Network to the individual Cyberguide units could look like.

The architecture of Cyberguide is modular. This means that the different components can be changed with a minimum of influence on the other components. Components can also be added to the service.

Cyberguide is related to our system CAIR in the way that they both utilize user's context (location) to provide the users with information. The information is assumed to be of interest for the user due to his location. The information provided by both services may be of the same nature such as history, happenings, relevant persons etc about the attraction of the location or the location itself. The systems are nevertheless different in several ways. In Cyberguide the user is not necessarily aware of his location, but in CAIR the user must have explicit knowledge about the location, such as the name, he wants more information about. Cyberguide constantly supervise the user's location and constantly provides the user with information at the right time according to his location. This is opposite to CAIR and M2S where the user initiates the queries and is given as many results as he has submitted queries.

If the timing is not right, the information provided might be of less use or maybe useless. If a tourist is looking at a painting made by Leonardo DaVinci and the information presented by Cyberguide is about the artist Michelangelo this information is misplaced and of less value now than if it was actually presented while the tourist was viewing pictures of Michelangelo.

Cyberguide is based on the assumption that the user wants information about his surroundings and is given the user without further instructions. In CAIR and M2S the user always starts the interaction and information about a specific location is asked for.

3.3 The Photo Navigator System

Pauty et al. [74] proposes a system that makes users able to navigate through a photo collection using context. Photo Navigator makes use of the users current location to propose him photos that are taken near him. The motivation behind Photo Navigator is to let a user get an improved understanding of his surroundings. Instead of having a stand-alone image collection the thought is to have a global network of images that can be shared by different users. The context that is used is time and location and the images must be annotated with this context and added to the image collection.

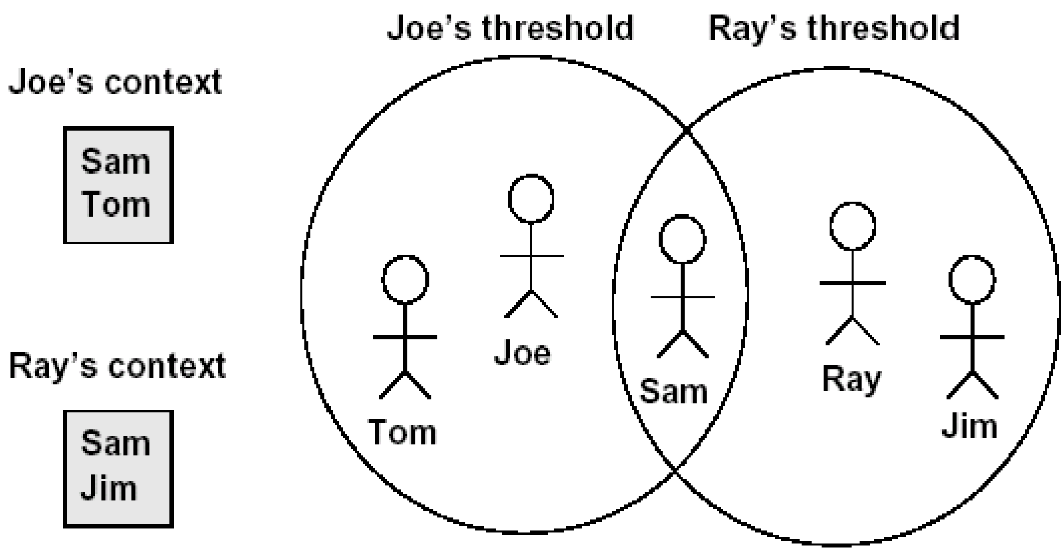


Figure 12: The user's context in the physical world [75].

Photo Navigator consists of two navigators; a physical and a virtual navigator. The physical navigator displays photos from the user's current location. The client terminal is a Pocket PC equipped with GPS receiver to monitor the user's location. The images in the collection are annotated with GPS-coordinates. The user's context is kept updated and is matched against the GPS-coordinates of the images in the collection. The images close to the user location are displayed to the user. The virtual navigator displays photos from the user's current context or the user's virtual context that is by location a bit ahead of the user. The service can in this way be used to see if an attraction is interesting or not. A user scenario will help explain how the service works and how it can be useful:

The user starts the information system and his location is determined. The photos displayed on the users screen are by location close to him. While he moves the user's context is automatically updated and the service responds by reflecting the updates on the screen. The automatic update of user's context keeps explicit user interaction to a minimum. He is now standing outside a castle and wonders whether to enter it or not. He continues the tour, but this time it is virtually. He starts the virtual navigation of the service. The photo of the castle is a part of the user's context while it is close to him. He selects a photo of the castle and the location of the castle becomes the user's virtual location. The user's context is updated reflecting his virtual location and making some virtual jumps the photos displayed can now be from inside the castle. By making virtual jumps in one direction, the user can know if the action is interesting enough to be taken in the physical world. Some buildings are not meant for tourist to enter and this technology still makes it possible.

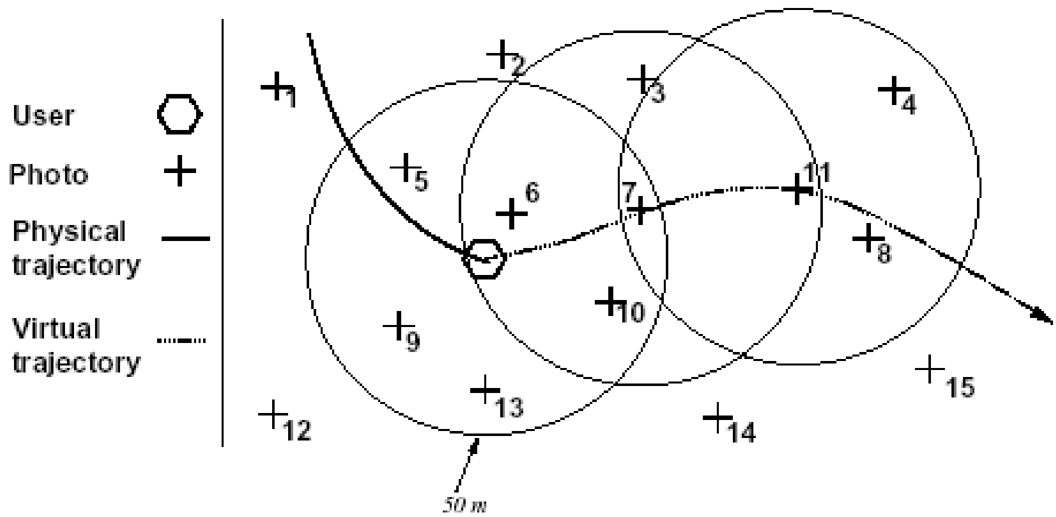


Figure 13: Shows how a user can go from physical to virtual navigation [75].

The application can also use time as context to display photos that are from a different year, time of year or day. An attraction can look very different if it is for instance covered with snow. The user might wonder how it looks in wintertime. The navigator can be set back in time; to wintertime or several years back in time. The user can see the castle covered in snow or he can see the castle and it surroundings from The Second World War and how it looked like then [74].

Photo Navigator is related to M2S and CAIR in the way that the user can explore locations by using his context. Photo Navigator has to constantly monitor the user's location and provide the user with constantly updated information. In CAIR the user's current location is sent in an SMS to the service. After his result is returned he is not provided additional information before he sends another query. The creators of Photo Navigator also stress the advantage that they have a minimum of explicit user interaction. This is however not something that we want to emphasize with CAIR. If the user wants to send loads of queries, that is consistent with our wishes. It is however important that CAIR provides users with “sufficient” information in each query SMS. CAIR is also similar to Photo Navigator in the way that it offers users images from a specified point of time. Photo Navigator can view images from different seasons and it has the ability to show images from specified years. CAIR has functionality to display images from a specified season, but does not have the ability to display images from a specified year. This functionality is however not difficult to add to CAIR.

3.4 Lincoln

Microsoft [76] has developed a research prototype of a service they call Lincoln [77]. The service allows users to search for information about an object by just taking a photo of it. Instead of letting users type the text describing photos, the user snaps a photo with the camera build in the mobile phone. The photo is matched against a database of images tagged with relevant web pages and comments supplied by a community of users. If a match is found this is returned to the user.

Microsoft argue that one way to use this service is to capture a picture of a movie poster or DVD-cover and use this as basis for the search. The result will then contain information about or a link to a website containing information about the current movie. To provide users with valuable results the database must be of large-scale to have a true utility value.

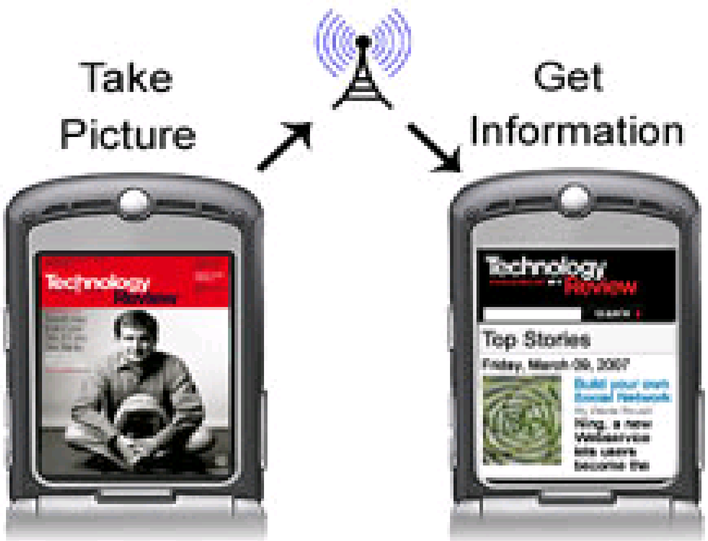


Figure 14: Shows retrieval of website requested using an image [78].

Figure 14 shows how a user can take a picture of a magazine with the mobile phone and be directed to the magazine's website.

At the moment the bottleneck of the system is in the communication part. According to Technologyreview [78] the image search takes about one second while uploading and downloading of images takes about four to five seconds. At the heart of the image-recognition engine is an algorithm that analyzes a picture and creates a signature that concisely describes the picture. This signature consists of information that describes the relative position of the pixels and the intensity of a certain feature in a picture. For efficient retrieval and matching, they group the features into sets of three, known as triplets. Lincoln does not describe any further what a triplet is or how the grouping is done, but an example triplet is shown in image 2 in Figure 15. Take for instance the painting of Mona Lisa. For this picture the algorithm will consider her smile, nose and cheek [76] [78].

Figure 15: Illustration of how Lincoln works [76].

The algorithm creates these data sets and using an inverse look-up table, compares them to already created sets of triplets of the pictures in the database. To verify the correct match, the spatial relationship between the matching triplets is checked for consistency [76].

Microsoft's claims that their approach makes searching through large databases more efficient than other methods that compare a large number of individual features one by one. Microsoft also argue that their engine is efficient because it only has to search for the triplets of data and not the hole image. This is because they consider the odds that there exist many images with the same three data sets are small [76] [78]. Even if there exist several images with the same triplets the developers have to consider the trade-off between computing time and precision.

According to Microsoft, there are two factors that distinguish this technology from others. First, the fact that anyone can contribute with images, links, and comments to the database. Second, they believe that the image-recognition system that they have developed will be able to search through millions of images quickly [76] [78]. Currently Lincoln can only be downloaded for free using Internet Explorer 6 and 7 and it can only run on smart phones equipped with Windows Mobile 5.0 and PocketPCs.

Some aspects of Lincoln are very similar to M2S. They both provide users with information about an entity he has taken a photo of and send to the server. In both services the query is formulated as an image and has a content-based image retrieval algorithm running in the heart of the application. As discussed, Lincoln use a techniques that group features of images in three known as triplets. These triplets form a signature that is matched against the already processed images. M2S use a content-based image retrieval program called Lire (Lucene Image REtrieval) [79]. Lire will be described later.

Both services provide users with information about the photographed entity, but the information is presented in different ways. Lincoln focuses on sending the user a web-link to information about the entity. M2S will first and foremost give users information in form of audio clips, web-links, textual information and series of images.

The services use databases with information. Today, the image database of Lincoln contains about 30.000 images. These are mainly of DVD-covers that link to movie reviews uploaded by Microsoft researchers. To achieve a larger database of images, any user can upload pictures and links to the database. Microsoft hopes that people will fill it with pictures and links to anything from information about graffiti art to scavenger-hunt clues [76]. Currently it is a small database containing MMS' and web-links. The database of M2S is generated by the developers of M2S and none but the developers has access to add any information to the database. Together with the fact that to construct rich MMS' is a more complex task than uploading a web-link to a database the evolvement of M2S's database will go accordingly slower than for instance Lincoln.

The way to communicate between mobile client and server are different. Lincoln uploads images via an Internet-connection between the mobile entity and the server. M2S makes users wrap the image taken in an MMS and send it to a specific phone number. There are disadvantages to the approach made by Lincoln. The bandwidth of the communication link can vary. Large images can take a long time to upload. This is also dependent on a constant link. As we discussed in the subchapter concerning issues with mobile entities, mobile entities can easily disconnect and be unavailable for periods. As mentioned the communication is the bottleneck of the service. At the moment MMS is a more stabile communication channel and is not so error prone. If the user is out of range of the mobile network, the MMS will be reside at the server side until the user is in range again. People are also more used to MMS and are familiar with how to wrap and send MMS'.

3.5 SnapToTell

Lim et al. [38] proposes a service called SnapToTell (also called Snap2Tell). The proposal makes users able to retrieve information about sights by using their mobile phone. The idea is that a user can take a picture of an attraction and send to a service provider via MMS. He then receives either an audio clip or textual information that contains information about the object.

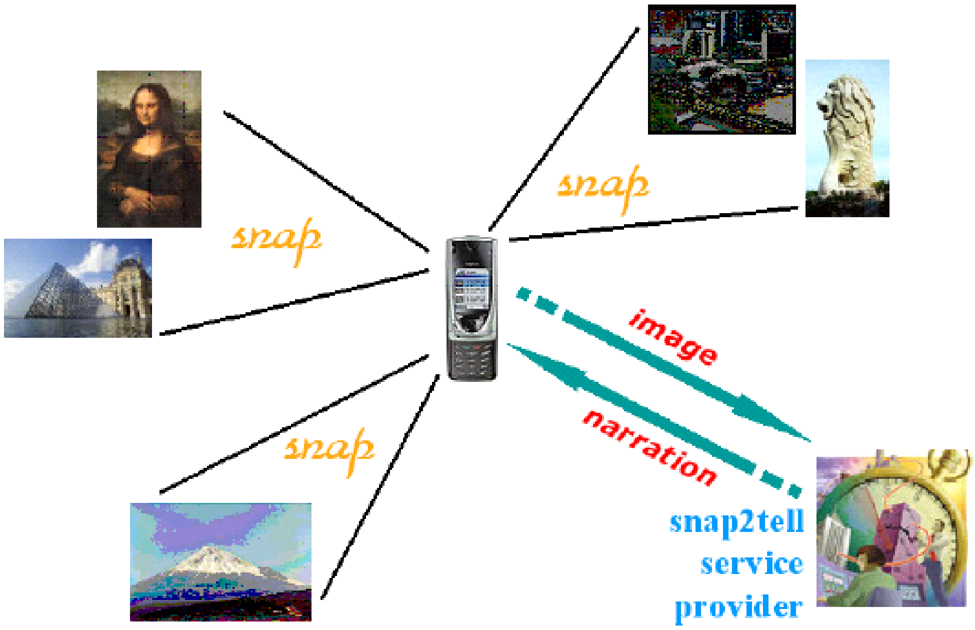


Figure 16: An application scenario with SnapToTell [38].

SnapToTell use a two-phase retrieval approach: one on the context (location) and one of the content (color histograms) of the image. The image is sent as MMS to an application server that handles the request. The user's location is determined and added to the MMS before it is sent. With the location identified, the Snap2Tell server sends a query to the database to retrieve the image metadata for the scenes related to the location. The image is extracted from the MMS and used as basis to search for similar images using an image comparison algorithm. The algorithm for image matching and indexing is based on color histograms (color comparison). If the best matching score is above a certain value descriptions of this best matched image is extracted from the database. Otherwise, a no match situation has occurred. In either case, the reply is formatted as an MMS and sent to the mobile phone of the user who initiated the query via MMS [38].

The developers argue that content-based image retrieval functions are in them selves not sufficient to base the service on. They therefore add the user's location to narrow down possible objects that the MMS contains. It would be possible to use location alone, but this could easily be a more inaccurate service. Detection of the user's location determines the user's position, but cannot capture the user's intention. What direction is he facing? What is he looking at? This is a description of how the service ideally would work. The authors have not developed the full worthy service, but have made a temporary solution due to the problems they encountered during the development.

The developers of the service were dependent on extern technology to work properly. As this technology was not available, they had to use other solutions. Already at the level of communication the technology failed. When the service was designed, there existed no GSM-modem that supported MMS. They therefore had no way to send and receive the MMS'. Instead the developers used a Bluetooth interface to communicate between the client phone and the SnapToTell application server. At the time the system was designed the mobile phones on the marked had no support for determining the user's location. The developers discuss GPS and the GSM cellular network infrastructure as possible tools as solutions. They argue that GPS is the most accurate method to obtain location information, but due to drawbacks such as higher hardware cost, power consumption, warming-up, satellite visibility in urban area they do not foresee GPS-equipped phones to become common in the near future [80].

The client phone (Nokia 7650) was not able to provide the developers with realistic location information. Therefore they had to simulate the location by a location ID sent by the SnapToTell client to the application.

Figure 17: Sample of SnapToTell screens on the mobile client [38].

SnapToTell is in many aspects similar to M2S. They both provide information to users by an MMS dialog. The results however can be different. SnapToTell can provide audio clips and textual information. M2S has the opportunity to send audio clips, web-links, textual information and series of images. As we have seen SnapToTell uses color histograms for image comparison. SnapToTell utilize HSV (Hue, Saturation, Value) color space for color histogram. Color histograms are widely used for content-based image retrieval. M2S use a content-based image retrieval program called Lire [79] to compare images. Lire utilize Caliph [81] to create an index of the images in the database. The images are indexed from scalable color, color layout, dominant color and edge histogram. Lire utilizes the information retrieval API Lucene [82] to the retrieve the most equal images. M2S has the advantage that it does not require users to be at the location of the attraction while taking the picture. As long as the user has the tourist brochure he can use the service from anywhere in the world as long as he is in range of a mobile phone network. In SnapToTell the user must take a picture of the attraction in itself. Together with the location information SnapToTell ties the user to the location where the attraction is situated.

SnapToTell is also similar to CAIR in some aspects. They both provide users with location-aware information, although SnapToTell does not use location alone to achieve this. CAIR is a context based information retrieval service and the query is formulated as text and is therefore wrapped as SMS instead of MMS. The result from the query is also different while CAIR offer the users series of images.

3.6 Mobile Access to Personal Digital Photograph Archives using context

Gurrin et al. [58] describes a system for context-based browsing of personal digital photo collections in mobile environments. They describe how they use contextual data such as time and location of image capture to manage a personal digital photo archive. A key assumption of their work is that use of context information like location, will improve user search and browsing, both on desktop and mobile devices, than time-based organization alone. After the photo is taken the context information is used to fetch additional context-based information such as weather and daylight conditions.

They argue that this is particularly applicable on mobile devices where the data handling, such as data entry, is restricted. With a GPS embedded for positioning, the time and location of image capture are easily available through the camera. Each photo is labeled with time and GPS location. An advantage of using context this way is that photos are automatically indexed and removes the need for manual annotation and indexing of photos. Another advantage is that the service is developed for mobile environments so it gives the users ubiquitous access to their photo collection.

There are about 10.000 weather stations positioned around the world that monitor the weather and log this information. Given this information and readily available access to it through the Internet the images can be automatically annotated with weather information from the weather station physically closest to the user at the time the image was captured. The weather information can be descriptions such as; sunny, cloudy, rainy etc. This means that a user can search for images taken on for instance a sunny day. The subject of the image is not taken into consideration, but the weather is sunny. To fetch additional context like weather information is optional. The camera can be set to use location and time only for annotation and indexing purpose.

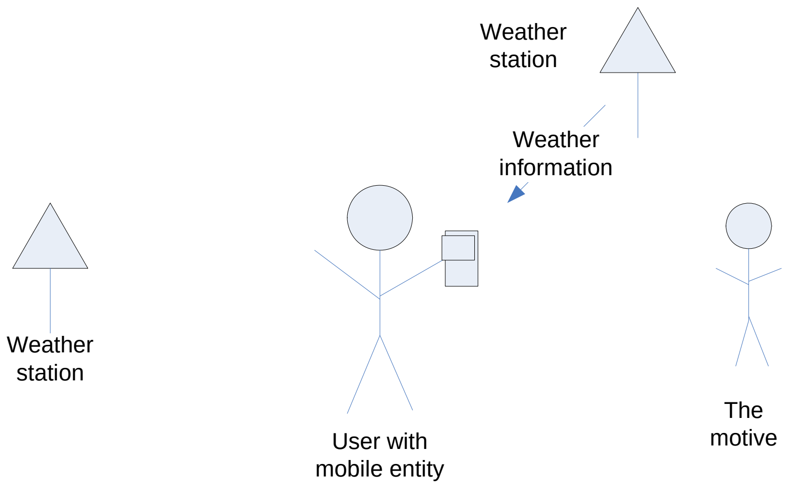


Figure 18: Illustrates the gathering of context information.

Figure 18 illustrates that weather information from the nearest weather station is added to the image as it is captured.

The developers have tried to follow guidelines for developing applications for mobile devices suggested in the literature. These guidelines include:

- Minimize user input. They provide simple user selections or simple hyperlinking instead of asking the user to formulate queries or to use visually demanding browsing that requires careful inspection of the screen.
- Filter out information so only a small amount of the most important information can be quickly and readily accessed.
- Provide proactively search and collect potentially useful pieces of information for a user and present it to him.

In addition they have developed an interface that also follow the guidelines:

- "My favourites" is the first screen the user meets when accessing the photo archive. This is a personal thumbnail displaying the top ten photos based on the user's history of viewing full size photos.

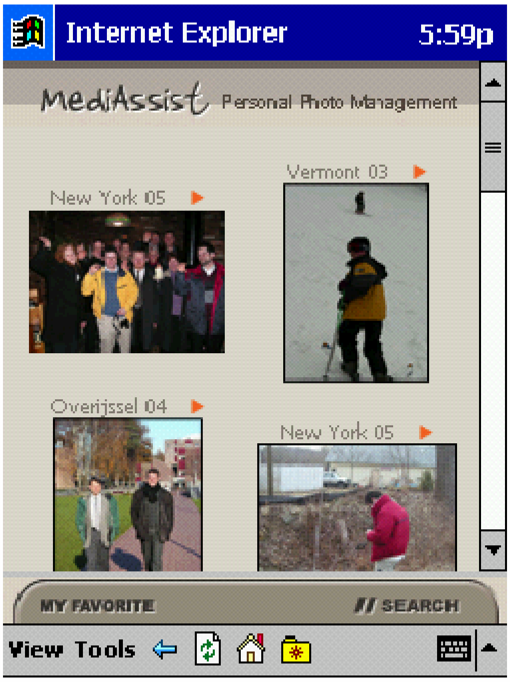


Figure 19: "My favourites" photos with search frame [58].

- "The search" functionality is based on location and automatic annotations. The intention is to reduce the amount of user interaction when locating relevant photos. The snapshots in Figure 19 and Figure 21 are on from mobile entities. To make fully use of the screen the search options are hidden in a slider that comes into view when the user needs it and disappears when it is not in use.

Figure 20: Shows how to search the photo archive [58].

- “Browsing” functionality supports browsing of the images when the above functionalities are insufficient. There might exist photos taken at the same time and location where browsing the images might be the only option.

Figure 20 shows how to search the photo archive. The figure also displays the slider that is displayed when the user needs it.

The first screen that meets the user are the “my favorites” screen. If the desired photo is not among the favorites, the user engages in the process of search and retrieval followed by browsing the result photos. The developers argue against browsing functionality. They also argue that it is important to keep user interaction to a minimum. Despite the effort the developers have made to avoid these issues, they have not come up with a solution that actually employs these.

This system is similar to CAIR in the way that context is used to provide users with information. In Gurrin et al.’s system this is an optional functionality that can be turned off. In CAIR this is the core purpose of the service. In CAIR the user must explicitly know the name of the location he wants information about. In Gurrin et al.’s system the user captures the picture and does not need to know anything about the location or for instance weather stations close to him.

3.7 Similarities between related work, M2S and CAIR

		Web-based	Cyber-guide	Photo Navigator	Lincoln	SnapTo Tell	Gurrin et al.	M2S	CAIR
Image retrieval tech.	CBIR				X	X		X	
	Context		X	X		X	X		X
	TBIR	X							
	Browse	X							
Info.	Constant		X	X					
	Query	X	X		X	X	X	X	X
Comm.	MMS				X	X		X	X
	SMS								X
	Web	X	X	X					
	WAP				X			X	X
Up-loading info	User	X			X		X		X
	Service		X			X		X	

Table 1: Illustrates similarities between related work, M2S and CAIR.

In cases where the developers of the service have not described the technology or choices, this cell in Table 1 is left empty. The table illustrates the similarities and differences of the different systems and how they relate to M2S and CAIR.

3.8 Summary

In this chapter we have described systems that relates to our systems. We have emphasized systems that utilize systems that use context in mobile settings to provide users with information. By describing these systems we have enlightened different ways of using context to serve the same purpose.

Typical web-services provide users with search and browsing functionality that simplifies the user's photo management but offer users little more than this. The users must explicit create queries and browse the photo collection. Cyberguide and Photo Navigator are far more automatic. The information is given the user in a constant flow as they move in the area and the users don't have to have any knowledge about their location.

In CyberGuide and Photo Navigator the user's can interfere with the software, but this is not decisive for the applications to function. In CyberGuide the user can for instance ask questions to "the guide". If no interaction is performed the user is given a standard tour. In Photo Navigator the user can decide to "enter" a building using the virtual guide or he can set the time on the navigator to see how the area looked like in the past. The user must explicitly ask to enter the building or this will not happen.

Lincoln enables users to retrieve information about an entity by taking a photo of it and upload it to specific servers. Databases are searched to find more information (in form of web-links) about this entity based on the image. The information is not given to the user automatically as in for instance Photo Navigator and Cyberguide but the user has to ask specifically for it. User gets as many "answers" as he has performed queries.

In SnapToTell the user has to ask explicitly for information by expressing a query consisting of an image, while the location is automatically added to the image. The user is then offered audio clips or textual information about the object. This form of communication is more like the traditional server-client inquiries and the user is provided with as many "answers" as he has asked for.

In Gurrin et al.'s system the information is automatically added to the image as annotation while the picture is captured, but the user has ask specifically for that information to make use of it. He can for instance search for images taken on sunny days and he will receive images displaying any subject, but the weather is sunny. The strongest quality of this service is the annotation of the image. The query in it self is a traditional client-server communication.

We have described the systems design and architecture and compared these to our two systems M2S and CAIR. We have also discussed some of the limitations, advantages and disadvantages regarding the related systems.

4 Introduction to M2S and CAIR

We have seen some of the systems designed for mobile entities that enable users to retrieve information. We will in this chapter discuss in what ways and situation people use their mobile phones, and further, how this provides a breeding ground for our services.

We will then introduce our services, M2S and CAIR, and see how the basic architecture of the services is the same and illustrates the use with a user scenario that applies to the services. We will give an overall description of the services and how they work. In this chapter we will emphasize the similarities and differences of the services. The similarities of the services are that they share much of the same motivation and they share the architecture, both services communicate by transmitting messages and they both have an image retrieval function. The differences are the type of messages transmitted, the type of image retrieval function used and how the information provided is represented in the database.

M2S and CAIR will be described and discussed in detail in respectively chapter 5 and 6 where the differences of the services will be more apparent. In chapter 7 we will give an evaluation and summary of both systems.

4.1 Motivation

In chapter 2 we argued that users bring their mobile phones almost everywhere and at almost all times. The mobile phone is much more than an entity used for oral communication. People use mobile phones to listen to music, gaming, instant messaging, sending and receiving MMS' and SMS', listening to radio, sound recording, watching TV, capture pictures and videos, surfing the internet while they are richly equipped with tools for alternative ways of communication. Today mobile phones can also communicate with other technical entities such as PCs, PDA's, other mobile phones, projectors and so on. Users are aware of the possibilities of the mobile phone as a multi-tool and users continuously discover the possibilities their mobile phone offers them. At the same time, as we saw in chapter 1, people are getting more used to being overloaded with information in "every" situation. Together, these issues support the users behaviour and perception of information and information retrieval. If the user is getting a lot of information about uninteresting topics at all times, why should he not be able to retrieve information about interesting subjects when he needs it?

We imagine that a user find himself in a setting where he wants more information about an object, location or attraction. He has brought his mobile phone, like he always has, and wants to use it to retrieve the desired information. For the user to be aware of information retrieval services and how they are used the services must be properly marketed. This is a crucial factor but beyond the scope of this thesis. It is now important that the service is easy to use and that it returns satisfying results.

Tourists visiting interesting places take a lot of pictures. Their mobile phone offers them the possibility to take a picture and instantly send it to family and friends together with comments like "look at where we are now"! Standard digital cameras are not (yet!) equipped with communication tools to provide this facility, so the only way to take a picture and instantly send to friends and family is by using his mobile phone. The tourist can of course take the picture, transfer the pictures to his laptop when he arrives at his overnight stop and send it by e-mail. But this is a far larger operation than the former method and the pictures are not sent instantly.

More and more information retrieval services are offered. Filmweb.no [83] provides users with a WAP-message with link to a webpage. The webpage offers a program for all cinemas in Norway. Users can see all premieres the current week and it is possible to watch trailers and retrieve news about the movies and about topics that are related to national cinemas. It is also a "Top 10" list displaying the ten most popular films. It is still not possible to order tickets via this WAP-link, but we believe it is only a matter of time before this functionality is in place.

Interactive TV-shows is another example of how people use their mobile phone by sending greetings to each other, answering quiz's, participate in contests and sending in personal pictures even though they have no guarantees that the message actually will be displayed on TV. As several TV channels offer such programs, this trend is increasing. We believe that one of the reasons why people use these programs is that it is easy and a fun way to use the mobile phone. The contest providers promote attractive prizes like gaming consoles, computer games, money and music collections. The winners of the contests are elected consecutive during the show and one of he advantages with live broadcast of contests is that the winners are elected instantly and they do not have to wait days or weeks for the winner to be announced. In the same way we want users to consider our services a fun way of using their mobile phone for interaction. As long as it is fun and provides satisfying results, we believe it will be used.

It is normal to see tourists carrying around tourist guidebooks. Our services are an alternative to or supplement to reading guidebooks and brochures. A guidebook can only display photographs and text. Our services, meant for mobile entities can display photographs, text, audio, video and multimedia clips.

In chapter 3 we described several systems that utilize mobile entities to provide users with supplying information about an object, attraction or location. We will in this chapter introduce two services, M2S and CAIR. The two services are of the same nature as many of the services described in chapter 3 in the way that both services provide users with desired information in mobile environments.

4.2 M2S and CAIR

The services demonstrate the use of different technology to achieve information retrieval. Both services are standalone services and they are interesting and relevant services because they try to solve user's instant need for information in mobile environments. This is a setting that has become more and more common the last years. The services can in theory be used on any portable device such as PDA's and mobile phones with functionality for camera and to send and receive MMS. It is however only tested on two different mobile phones.

The user is in a situation where he finds himself wanting (more) information about an entity. This is often a tourist setting. After watching commercials for our services he remembers the service number 2034. The user sends a request for information to number 2034. This request is formulated according to the chosen service's pattern for request. The Telenor Message Server sorts the messages sent to 2034 by the first string in the request. The request is then forwarded to the proper service. In our case this will be either M2S or CAIR. The service handles the request for information and sends a reply to the user. The reply is also formulated according to the service's requirements.

The services have similarities, but they also have differences. The services are different in the way the images and the associated information are represented, the way queries are formulated and the formulation of the result. We will return to these differences in chapter 5 and 6. Figure 21 illustrates some of the differences between M2S and CAIR.

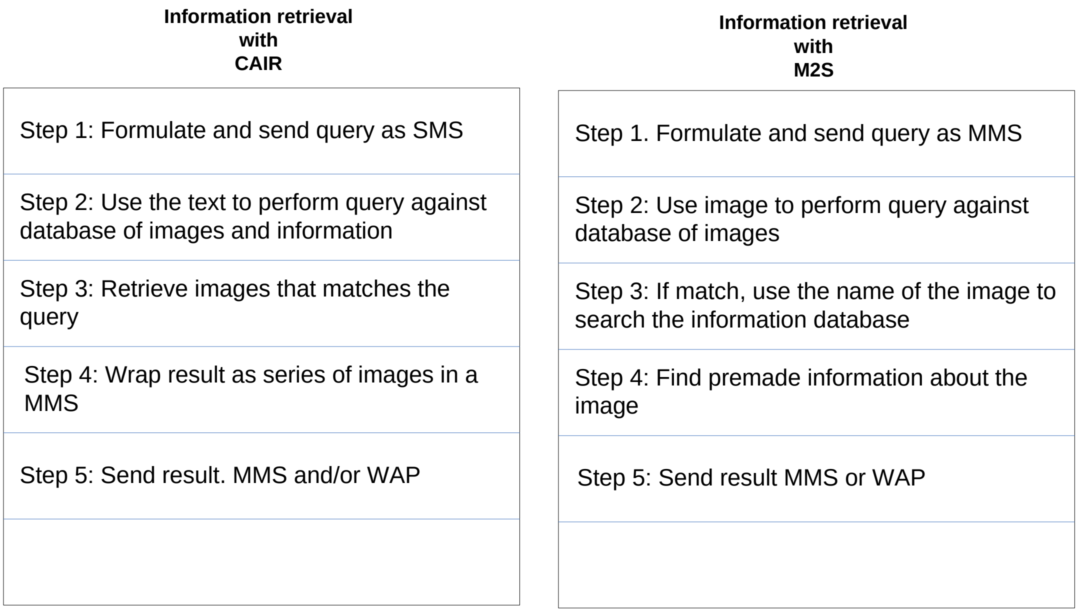


Figure 21: Illustrates how CAIR and M2S works.

4.3 Architecture

M2S and CAIR use the same communication mechanisms, but there are several other differences. The way the information and the images in the databases are represented and he formulation of the query and reply themselves are different. In M2S the query is formulated as an image and sent as MMS. The query image is the photographed entity that the user seeks more information about. In CAIR the query is formulated as text and is sent as SMS. The query text consists of a service name, a location and a point of time. In M2S the reply can be an MMS or WAP URL. In CAIR the reply can be MMS and WAP URL or SMS.

	SMS	MMS	WAP
M2S		X	X
CAIR	X	X	X

Table 2: Illustrates the message type used for communication.

The query and reply will be described further in this chapter and in chapter 5 and 6. The communication between user and service is illustrated in Figure 22. The services use mobile phones to communicate with the remote information retrieval server. Both services are designed for mobile entities with capabilities for MMS.

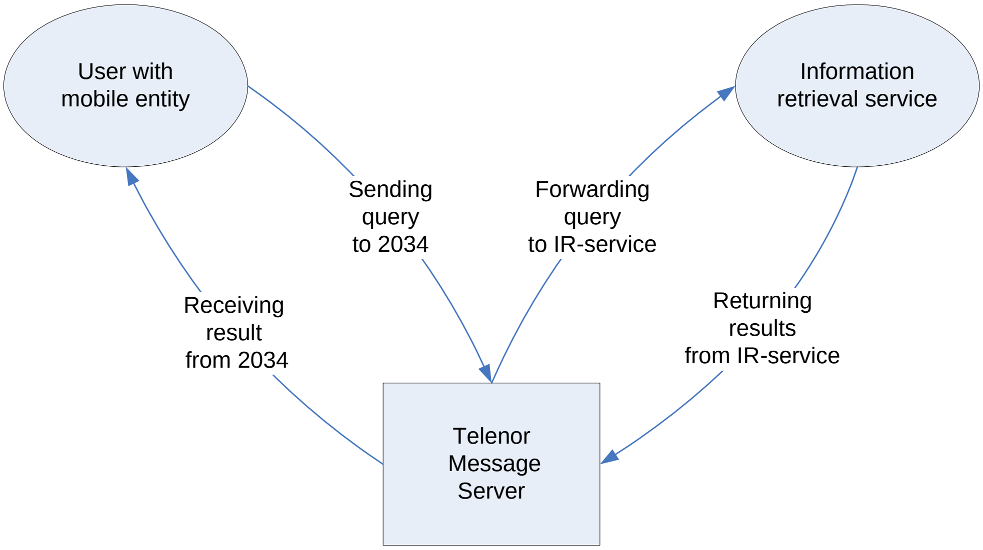


Figure 22: Illustrates how the communication of M2S and CAIR functions.

4.4 Communication between client and server

In both M2S and CAIR the client and server communicate by transmitting messages. The messages used in the services are SMS, MMS or WAP. In M2S the client initiates the communication with an MMS. The service treats the MMS as a request for information. Dependent on the contents of the information database, the service sends a message corresponding to the match. In M2S this can be either MMS or WAP.

In CAIR the initiating message is an SMS. Like M2S, CAIR treats the message as a request for information. The answer of the request is in form of either an SMS or MMS dependent of the information in the database.

4.4.1 SMS

SMS is a message of text that can be transmitted between mobile phones, other handheld devices such as PDA's and stationary telephones. Messages are sent to a

SMSC (Short Message Service Centre), which provides a store-and-forward mechanism. It attempts to send messages to their recipients. If a recipient is not reachable, the SMSC normally queues the message for later retry. Today, practically all mobile phones are equipped with functionality for transmitting. SMS. Transmitting SMS has become extremely popular and has made changes to the way people communicate with each other [84] [85].

4.4.2 MMS

MMS is a further development of SMS. MMS enables user to transmit multimedia messages that can include images, sound, text and videos in one message. With MMS, a mobile device is no longer confined text messages only. As with functionality for transmitting SMS, practically all mobile phone are equipped with functionality to transmit MMS and most media companies, such as Telenor, supports this as well [84]. There are two modes of delivery in MMS: immediate or deferred [85]:

Immediate delivery: When the MMS client on the mobile phone receives the MMS notification, it then immediately (without user intervention or knowledge) retrieves the MMS message from the MMSC (Multimedia Messaging Service Center) that sent the notification. After retrieval, the subscriber is alerted to the presence of the arrived MMS message.

Deferred delivery: The MMS client alerts the subscriber that an MMS message is available, and allows the subscriber to choose if and when to retrieve the MMS message.

As with the MMS sending, the MMS retrieval request, whether immediate or deferred, occurs with an HTTP request. The MMSC responds by transmitting the MMS message in an HTTP response to the MMS client, after which the subscriber is finally alerted that the MMS message is available [84] [85].

4.4.3 WAP

WAP (Wireless Application Protocol) enables users to access Internet from a mobile phone or PDA. A WAP browser provides all of the basic services for a computer based web browser but is simplified to operate within restrictions that come with hand held mobile entities. WAP is now the protocol used for the majority of the world's mobile Internet sites, known as WAP sites. Before the introduction of WAP services providers for mobile contents had very limited opportunities to offer interactive mobile services [84] [85].

4.4.4 API

An API (Application Programming Interface) is a source code interface that a computer system or program library provides in order to support requests for services to be made of it by a computer program [85].

4.4.5 PATS and the four-digit number

Four-digit numbers are also known as a short code [85] and are common to use for this kind of service. This four-digit number is controlled by PATS [86]. PATS is a research cooperation between the Norwegian University of Science and Technology (NTNU) Department of Telematics, Ericsson, Telenor and Hewlett-Packard Background. The PATS lab is facilitating the development of new telecom services such as for instance M2S and CAIR. To register and operate such numbers is relatively expensive and therefore several services share the same four-digit number. PATS must somehow identify which message goes to what service. This problem is solved by including a service name in the messages. A typical message name for M2S could be “M2S” and a typical message name for CAIR would be “CAIR”. These service names should be case- insensitive. The service number is put in the text field of the message (SMS or MMS). When the message is received at the message server at the PATS lab, the message is sorted on the service name and is forwarded to the corresponding service [15].

4.5 The Query

The request for information is always initialized by the user by formulating a query and sent to the service number 2034. As we have seen, the two services require different formulations. The core functionality for the services is the image retrieval functions. This core function requires a picture to compare directly against the service provider’s image database. The query must therefore be formulated as an image.

Because the image retrieval function that runs in the heart of the M2S is content-based, the best solution is to have the query formulated as an image. In this way the image is just extracted from the MMS and is fed directly to the content-based image retrieval function. When the service receives the MMS, the image is extracted from the MMS and is directly compared against the image collection.

CAIR uses a context-based approach (location and time). The query is text-based. A query is formulated and sent as an SMS. To run the image retrieval function in CAIR the query and the image annotation must have the same format. If needed the query is reformatted and compared to the context of the images in the image collection.

As Dey’s definition also includes “*objects that is considered relevant to the interaction between the user and the application*” it is possible to consider M2S a context-aware information retrieval service as M2S utilizes the tourist brochure that can be considered a part of the user’s context. The query could then be classified as context-aware.

4.6 The Reply

In M2S the result information in the MMS’ are pre-made. Complete videos and image series are made in advance, wrapped as MMS and added to a database. A specific query results in specific answers, meaning that a picture of an attraction always returns the same information. There might be several results connected to the same picture but the query will never end in an unexpected result. The information can be represented as textual information, audio, a series of images or multimedia.

CAIR returns series of images that are built on the fly by the server. For each match in time and location the image is added to an array and the array is wrapped as MMS.

In CAIR there are not different formats on the result, but only series of images and they are not pre-made. If a match in time and location is found, the image is added to an array and wrapped as an MMS. If several images is found these are added to the array too. This process is done on the fly. The message will be displayed to the user as a series of images and can consist of one or several images. The query: CAIR LOFOTEN SUMMER, results in a series of images from Lofoten. This series might be different from other results even if it has the same query. It also means that the same images might appear in several series of images.

4.7 Client/server

In these services the user's mobile entity is the client and the service host is the server. In Figure 22 the client is the part marked with "user with mobile entity" and the server is the "information retrieval service".

When designing a client-server application, it must be decided which parts of the task should be handled on the client, and which parts should be handled on the server. As we will see, this decision can crucially affect the cost of clients and servers, the robustness and security of the application as a whole, and the flexibility of the design for later modification or porting [85]. A client-server interaction requires some communication between the participants. The communication can be achieved in several ways, for instance Bluetooth, IR or messages like SMS. This is of course dependent of what tools the client is equipped with. Although there is a range of client architectures in the degree they are equipped, there are two main categories: thin and thick clients.

4.7.1 Thick client

At thick clients (also known as fat or rich clients) the logic and persistent data will reside at the client side. The logic and the data can be updated, but this is a more complicated task while the clients can have different architecture and require model specific software. Thick clients are therefore less portable than thin clients. It can also contribute to clients computing on non-consistent data. Many people prefer thick clients while they can be heavily equipped with applications and can store large amounts of data and the client is able to function independent of the server. This is a major advantage when the wireless connection is unstable as it can easily be in mobile networks. Another advantage is that users have more control over what programs are installed and specific system configuration [87].

4.7.2 Thin client

At thin clients (also known as lean client) the logic and persistent data reside on the server side, also the client-specific logic. The server receives a request for computation and if required sends the result to the client who initiated the communication. A consequence of using thin clients is that users do not have to install any software while the clients are already equipped with the required software needed. The thin client architecture also makes the system very portable. It will in many cases be possible to reuse former developed concepts and therefore could shorten the generation time considerable. Because the data resides at the server side, they are easily updated and requires close to no support. There are also benefits of extremely high security in reduced vulnerability to viruses, spyware and other malware [87] [88].

M2S and CAIR only require use of thin clients while the workload is carried out at the server side. In order to use M2S the client must be equipped with a camera and the client must have MMS, SMS and WAP functionalities. In order to use CAIR the client must have functionality for MMS, SMS and WAP. Today, practically every mobile phone is equipped with these functionalities, so users can employ their personal mobile phone without exchange it for a newer model or updating any software.

4.8 Representation of information

In M2S and CAIR the information and the images are represented differently. In M2S the image itself is used to find similar images. If a similar image is found *the name of the image found is used as basis for a lookup in the information database*. In other words: the information and the images are not stored together and the operation is a two-phase operation. In CAIR the context, time and location, is represented as text. In CAIR the images and information are stored together. The images are annotated with time and location and it is this text that is searched.

We want the results in M2S and CAIR to have high precision. Using a relevant/not relevant classification as described in section 2.6.1 the result MMS as one is in M2S considered either relevant or irrelevant. In CAIR each image in the result MMS might be considered relevant or irrelevant.

4.9 Image retrieval function

The services use different techniques to retrieve the images. M2S use a content-based image retrieval technique. The image snapped by the users is used as query and is directly compared against the images in the database. CAIR use a context-based approach. The image-capturing device automatically annotates the images with context (text) like time and location. This text is used as basis for the search for matches. To search for matches a text-based image retrieval function can be used.

4.10 Summary

In this chapter it is given an introduction of what situations people utilize their mobile entities for information retrieval. An introduction of two services, M2S and CAIR is given. The services are designed to cover a need for information in mobile environments. We have given a scenario for in which situations our services are useful.

The M2S and CAIR have similarities and differences. The similarities are in the architecture and the communication part. Both services use SMS/MMS to communicate with the service. The differences are mostly in aspect to the formulation of query and answer, the representation of the information (how the information and images are stored) and the image retrieval function running in the bottom of the services. We have described the different types of message the services use and MMS, SMS and WAP are described. PATS is also described. PATS provide us with the opportunity to transmit MMS and SMS.

This chapter has given a brief description of the services. Both services will be described and discussed in detail in chapter 5 and 6.

5 M2S

In this chapter we will describe M2S, the motivation behind, alternative solutions to the service, the solution we chose, the design, the implementation, test results and evaluation of the service. This chapter will also include matters like tourist behaviour and tourist information to help build motivation for M2S.

The description of M2S in this thesis is identical to the design of M2S as it was when I finished the summer work of M2S at Telenor R&I the summer 2006. Since then the project has been developed further by Anders Schürmann et al. and the project can be viewed in the Research Note "Let a picture initiate the dialog! A mobile multimedia service for tourists" [15].

5.1 Introduction

M2S is a mobile service meant for tourists in Lofoten. It is a mobile service that offers tourists in Lofoten information in an untraditional way. M2S belongs to the MOVE-project [14] at Telenor R&I. The MOVE-project examines the tourists information need and provides them with services for mobile environments that meet these needs.

The aim of the MOVE-project is to enhance the traveler's experiences through new aesthetic expressions mediated by technological solutions that merge people, objects and places with their network digital representations. In parallel, attention is given to the many kinds of tourist providers and their efforts to face the tourists at the right moment with the right information [89].

The MOVE-project focuses on developing a *digital marketplace for services in Lofoten* [90]. The marketplace will be a junction for actors like buyers and sellers. A buyer will typically be a tourist seeking information and the seller will typically be a tourist agent or agency offering a service. The marketplace has information- and communication infrastructure to broadcast digital information and services. One service offered at the marketplace is our M2S service. Other services that are offered at the digital marketplace are mobileMap [87] and Geopuzzles [91]. The goal was to develop a service that made it possible for a user to take a picture of a tourist attraction and based only on this picture the tourist could retrieve information about the attraction.

5.2 The M2S service

The service enables a person to take a picture of a picture in the Lofoten Info-Guide (provided by Destination Lofoten), send this to a specific phone number and retrieve information about the entity photographed. The query picture is sent as an MMS and initiates the dialog with the service. In less than a minute, the user receives an MMS from the service. The answer MMS can consist of a video, series of images, textual information or a web-link providing rich information about the entity.

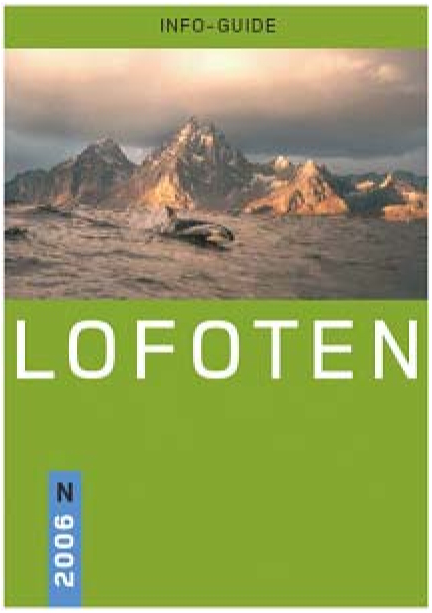


Figure 23: The Lofoten Info-Guide provided by Destination Lofoten [92].

5.2.1 Specification

Telenor R&I wanted developed a service that provides users with information while they are on the move. Their suggestion was that the service was going to:

- provide users with valuable information
- use MMS to communicate with the users.
- use a content-based approach to recognize the query image.
- use a tourist brochure as work-around because of shortcomings of content-based image retrieval functionality.
- use a pre-made content-based image retrieval service.

The pre-made content-based image retrieval functionality should be:

- free of charge.
- open source or have an API that we could program against.

5.2.2 Restrictions of the service

The users cannot take a picture of whatever attraction they see, but has to take a picture of a picture of the tourist brochure, the Lofoten Info-Guide, provided by Destination Lofoten. The service is fully dependent of the tourist brochure and it has to be in the hands of the tourist for him to use it. The desired attraction *has* to be displayed in the tourist brochure and it is not adequate with textual descriptions of the attraction. If a tourist wants information about an attraction that is not in the brochure, this service cannot be used to retrieve information about it. He can of course take a picture of the attraction itself if he happens to be at the location of the attraction, but the chances that the image of a free-standing attraction will trigger a hit in the image index is very small. This is related to shortcomings to content-based image retrieval functions that we will discuss later.

5.3 In what situations is our solution helpful?

Before visiting a new place many tourists carry out research about the destination before they leave home [93] [94]. The information that tourists seek often includes maps of the country and city that will give them an idea about transportation networks. They check essential conveniences like overnight stops, good restaurants (like the Michelin Guide [1]) and other tourist attractions. A tourist may be interested in having a basic idea about the country's way of life, such as its markets, characteristic architecture, traditional clothing and everyday phrases of the national language [93].

We believe that the target group for our service is so called leisure travelers[2]. Leisure travelers are tourists that travel in their spare time. Consumer surveys show that 50% of decisions about which tourist attractions to visit are taken during the trip [94]. It is logical to assume that the tourist will investigate more about possible attractions while visiting the destination. This means that it is really important for agents from the travel-business to provide the user with useful information while they are traveling.

Attraction is a very important keyword in the travel-business. When a facility is so special that it attracts visitors because it is something beyond the ordinary, we are taking about an attraction. It varies a lot how strong an attraction is, or how much appeal the attraction has on the tourists. An attraction that is interesting to some people might be uninteresting to others. We have two different types of attractions; so called "free goods" like fresh air, clean water or nature or it can be something that tourists will pay to use or get access to like museums or boat trips [95]. Leiper [96] holds that there exist several types of attractions. Primary attractions are attractions that can be a main argument for a journey. The attraction is pronounced and known to the tourist in advance of the journey. A good example is The North Cape that is an attraction itself. Secondary attractions are also known in advance of the trip and might support the selection of destination, but are not the main reason for visiting the destination. A collection of secondary attractions can together form the choice of destination. Tertiary attractions are unknown to the tourist before departure but can be discovered during the journey based on information retrieval. In such cases the tourist explores their destination to enrich the journey [96] [15].

Traditional ways for the tourist to retrieve information is from tourist brochures, TV-commercials, travel agencies and other travelers. We believe that users are more demanding when it comes to information retrieval today than a few years ago. People are used to getting information faster and therefore their expectations of when and where they can get desired information is increasing. The mobile phone is a powerful tool to help developers meet the user's expectations in this area. This is meant to be an easier and more fun way to retrieve information. We want to exploit the features of mobile phones. The mobile phone makes users able to retrieve videos, sound and images instantly independent of the user's location. In this way one can distribute information beyond the abilities of traditional media's.

The information sought helps the tourists decide what to see and what to do at their destination. This information often consists of other tourist's description about the place they are about to visit, images, videos and so on. Consumer surveys also shows

[1] http://www.michelinguide.com
[2] Fritidsreisende, freely translated.

that the active tourists are most satisfied. This supports the idea on which the MOVE-project is founded, namely to ease access to regional and dynamic information and make it accessible for users via a wireless mobile connection. [14] [15].

Dynamic information includes information about local happenings such as football matches, hotel vacancy, restaurant menus, weather dependent activities, etc. This information may change, some more rapidly than others, and therefore cannot be printed in a tourist brochure that is distributed months before the tourist season begins. A service that offer tourists information about such dynamic and regional events can enrich a tourists visit by providing such valuable information about attractions that are otherwise not easily available [15].

5.3.1 Information retrieval in a tourist setting

A growing tendency in the Nordic countries is that tourists travel around on their own in a private car, with or without a caravan or mobile home. This challenges the way of distributing information to the target group in a systematic way. Flognfeldt [95] says there are three points of time for tourists to retrieve information:

- Information retrieval before the trip is made.
- Information retrieval during the trip to the destination.
- Information retrieval after arriving at the destination.

We want to introduce a fourth point of time of information retrieval:

- Information retrieval after the trip is made.

The information retrieval before the trip is made is also called marketing on domicile and is the information retrieval before the user leaves home. The information about the destination can be in form of an advertising film, posters, brochures, or from the Internet [95]. The former kind of advertising is normally expensive and difficult because it demands that the distributed information distinguishes from all the other information from the other actors in advertising. The Internet has become a common channel for retrieving information about conceivable destinations. A frequent problem related to Internet as information retrieval channel is that it often gives too much information. The information is also often outdated, insufficient and of varying quality and to retrieve correct information fast is often not as easy as it seems. Much of the information sought in this phase can be hard to bring into the next phase of the travel. The most obvious solution is to print the information or note keywords of the information. However, this information might consist of videos and these are not that easily brought [87].

The information retrieval that is made during the trip to the destination is typically brochures received at traffic junctions like ferries and airports or information agencies [95]. The information in this phase is normally specific to the user’s current location or information about their final or next destination.

The information retrieval after arriving at the destination is typically in form of a brochure from a local travel agency or information agency, the Internet or it can come from other travelers [95]. The rapid development and improvement of the hand held devices as PDA's and mobile phones together with fast communication channels, enables user to transfer relatively large amounts of data. This enables users to digitally orientate themselves during their travel and after arriving at the destination. Location-aware services would have a great potential in these phases. As mentioned, the information sought is most often specific for their location. A mobile phone could function as the tourist guide and constantly provide him with updated information in a personalized way. The information can be customized to fit the user's language, interests or otherwise fit the users context such as for instance his sense of hearing. The information retrieved can be everything from overnight stops, places to eat, events, shopping facilities, traveling options, time tables and attractions [87].

The information retrieval after the trip is made covers the information retrieval after the user has arrived home. At this point he might want to seek more information about an attraction he visited during the trip, either for a later visit or to show friends where he has been. This phase is an important phase where people remember the trip. As long as the tourist has the brochure it is possible to retrieve information using M2S. We hope that M2S can provide users with information in all these phases of time. We hope that users will consider M2S an untraditional, modern and fun way to retrieve information about a destination and a supplementary to traditional tourist guides.

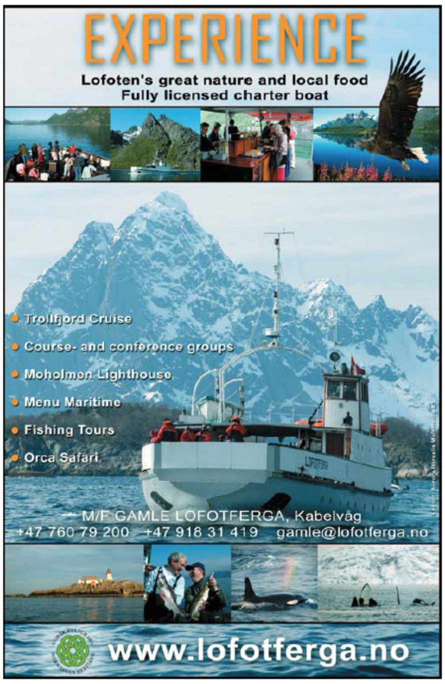
We imagine users, a young couple from Bergen, have decided to spend their holiday in Lofoten. They are particularly interested in fishing. To seek more information they visit http://www.lofoten.info. This site promotes M2S and the young couple gets curious. They want more information about options for fishing and they decide to try this information retrieval service. They order the tourist brochure and it arrives in the mailbox after a couple of days. The brochure gives them a good impression of Lofoten and using M2S gives them additional information of for instance fishing facilities. As they travel north they can easily fetch information As long as they have the brochure they can fetch information as they travel. This solution provides mobility and independence of opening hours of the tourist information office and other local information providers. As the couple returns to Bergen they can still fetch information and it can be shared with friends and family.

The reason why we chose Lofoten as demo-destination was because this is a geographically limited destination that requires transportation by ferry or airplane, and thus an opportunity to provide travelers with brochures. Lofoten has also been, compared to other Norwegian destinations, profiled a lot abroad. Destination Lofoten has its own tourist brochure that covers many of the attractions in Lofoten. Lofoten offers many attractions a great variation in the types of attractions, which is also reflected in the tourist brochure The Lofoten Info-Guide. Variation in types of attractions also gives variation in the information we want to provide.

5.3.2 User scenario

M2S is built upon the following scenario [15]: A tourist is visiting Lofoten and at a petrol station he finds the Lofoten Info-Guide. It is a well-written guide full of advertisements and glossy pictures of attractions in Lofoten. He looks at the beautiful pictures taken on a sunny day in an advertisement for The Ferry of Lofoten. The pictures look appealing maybe a Trollfjord cruise might be amusing?

To get more information about the cruise, the tourist picks up his mobile telephone and takes a picture of the picture of the ferry. He dials a four-digit number that he remembers and sends the picture to this number.



Within a short period of time he receives an MMS containing a video from the last days Troll-cruises and an inspiring and confident voice invites him to participate. The weather forecast for the next day looks promising and there are still tickets for tomorrows' cruise available. He cannot wait to forward the MMS to her friends with her personal message: Look, this is tomorrow's adventure! [15].

The described demo is meant for tourists in Lofoten and is dependent on the Info-Guide, but as content-based image retrieval techniques become more highly developed, we believe it will be possible to free us from this dependency and it will be possible to take pictures of freestanding attractions and use as basis for search for information about the motive. Problems with content-based image retrieval techniques and future work will be discussed later.

Figure 24: Illustrates what an MMS ready for transfer could look like [15].

Figure 24 shows a query MMS containing a picture of the Lofoten Ferry ready to be transferred. The image is extracted from the MMS and is used as basis for the query.

The visual content of the image is compared to the other images in the database and the best match is found. The associated information of the image is sent back to the user

.

Figure 25: Shows images from Lofotferga.

Figure 26: Shows the URL attached.

5.4 Design and architecture of M2S

Figure 27 shows an overall description of how M2S functions. The figure illustrates how the transmission of messages in the service functions.

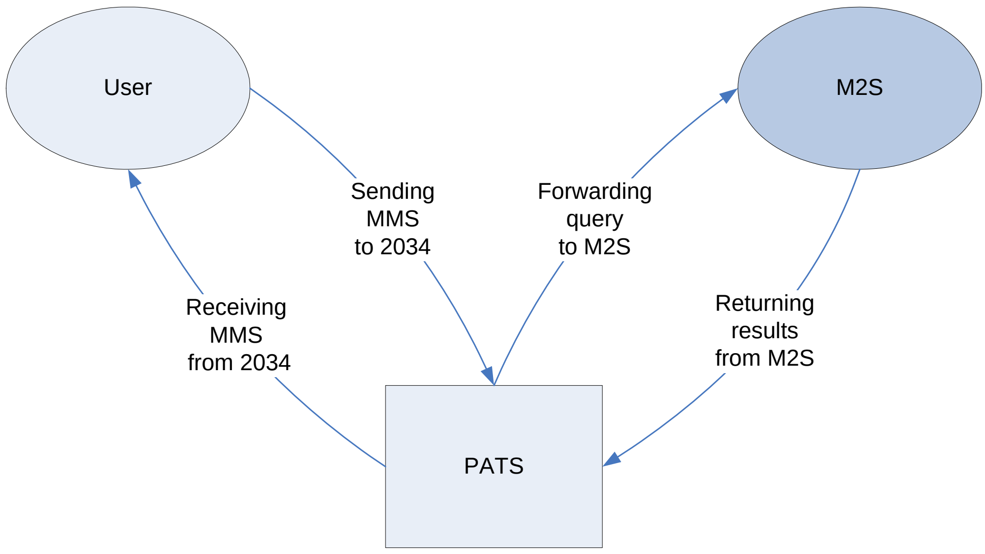


Figure 27: Gives an overall description of how M2S functions.

Figure 28 shows a more detailed description of the "M2S" part of Figure 27. The figure is also a more specific variant of Figure 22. Figure 27 illustrates the communication between the user and the M2S server. The query is send to the short number 2034, the messages go via PATS and ends at the M2S server. The M2S server, where the service runs, is marked with dark blue shadings of Figure 27. The details of this part are illustrated in Figure 28. As can be seen in Figure 28 we have chosen a modular design of M2S for easy exchange and updates modules handling the image comparison and messaging functionality. In Figure 28 the blue shaded boxes illustrates the main progress in the service. The grey-shaded boxes illustrate the image retrieval functionality and the white boxes illustrate the information retrieval functionality. This means that at the moment there is available a improved image retrieval programs the former image retrieval program can easily be exchanged with an new one with a minimum of impact on the rest of the service. The database can also be exchanged in the same way.

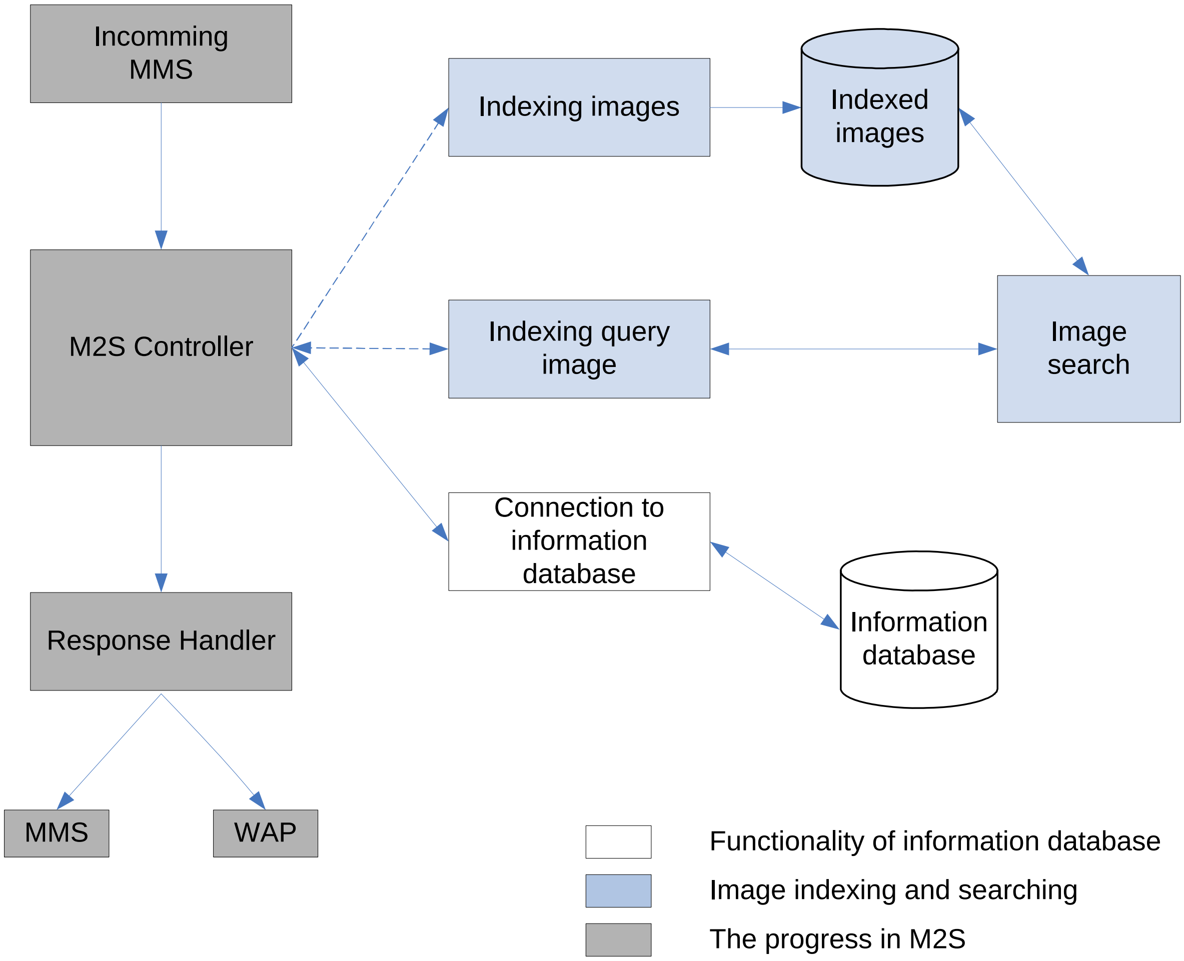


Figure 28: Shows the process flow in M2S.

Figure 28 is based on a figure 2.2 in Schürmann et al. [15]. The figure also shows that image retrieval is possible with Lire [97], Eikon [98] and any image search program that has an API that it is possible to program against. The dotted lines to the search programs (Lire, Eikon and other) illustrates that this part is more loosely tied to the rest of the service than the other modules. To make the program dynamic in this way, we have made use of a concept called reflection.

Reflection is when the program has the ability to detect and possibly modify its own structure and behavior to improve its flexibility [99]. Reflection allows us to bind to a specific searching service at runtime. The name of the search program is read from a configuration file (properties.txt) during runtime and the main method of the program is run. To change the search program, the name of the program in the configuration file is changed. The search program that is run must take a string as parameter. This parameter is the name of the image searched for.

5.4.1 Information flow in M2S

Figure 29 illustrates the information flow in M2S. The figure shows the progress in M2S depending on the events. The preferable progress of M2S is following the horizontal line of events, meaning that an MMS is returned.

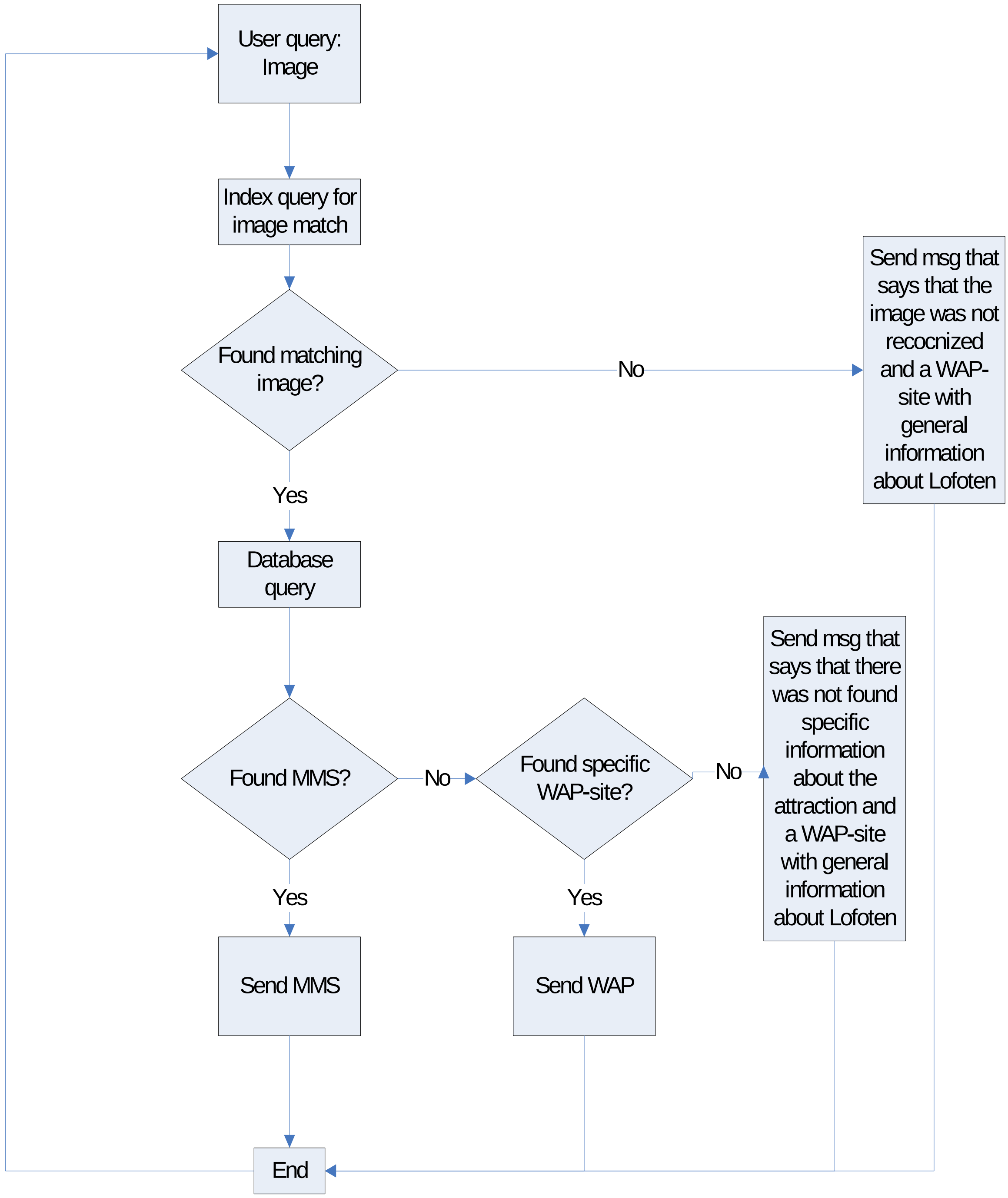


Figure 29: Information flow in M2S.

5.4.2 User requirements

It is not required for users of the service to download or install any software on the client entity. All the functionality needed such as transmitting messages is nowadays included in the majority of mobile phones. It is advantageous that the users do not have to install anything before it can be used. If users are confronted with tricky settings or software that is hard to install we believe many users easily give up even before they have actually tried. Even if the software is said to be easy to install we believe this is a threshold for most people. How big/high the threshold is, depends on among other things the user's technical skills, what he believes he can gain from using the service (self-efficacy) and pressure from his social acquaintances (social pressure) [87].

The mobile phone must have functionality for transmitting MMS and WAP URLs and it must be equipped with a camera.

5.5 The content-based image retrieval functionality

This chapter will discuss issues related to the content-based image retrieval in M2S.

One of the requirements of the service was that it should be easy and fun to use. We believe that snapping a photo of an attraction and use as query, either if it is from the brochure or not, satisfies both these requirements. Receiving a photo without any metadata, we had to somehow identify what the picture was about. We concluded that a content-based image retrieval function could solve this problem.

Developing a content-based image retrieval function was a too complex task for the summer project and this thesis. We therefore decided to use a freely available program to handle this part of M2S. Before deciding what content-based image retrieval program to use, several content-based image retrieval programs were evaluated.

In chapter 2 we described content-based image retrieval and what difficulties that surrounds this concept. We have experienced some of these difficulties personally and based on these experiences we believe that content-based image retrieval has a long way to go before it is an adequate technique to be used in large scale for image retrieval.

We will present the programs considered. The content-based image retrieval program we chose for M2S is called Lire. Lire will be presented and we will describe some of the details about how Lire functions.

5.5.1 Evaluation of content-based image retrieval programs

Since M2S is, for the time being, only for demo-use we want to use a content-based image retrieval program that is free of costs. This means that the function that we have chosen to use is not necessarily the most efficient one. Commercial systems like QBIC [2] might be give better retrieval results then the ones we have tested, but this is not necessarily the case.

The content-based image retrieval system had to be free of costs. It had to be open source or at least a published API that it would be possible to program against. The system had to be able to find obvious features of images and find the same features at similar images. If for instance the systems did not have a published API or they provided relevance feedback, we did not consider the system further. We have considered several search programs, but most of them have features, or lack of them, that makes them unfit for our use. Among the considered programs are Lire [79], Octagon [35], MascoT [100], The GIFT [41], Eikon [101], BolinOS [102], ImgSEEk [39], ImageSeeker [103], CIRES [104], In-Two [105], MUVIS [106], Riya [107], SCHEMA [108], Viper [109], VIRAGE [110], Istorama [111], Blobworld [43], WALRUS [44], PictureFinder [112] and SIMBA [113] .

Because the image retrieval is a completely automated task with no human interaction, systems like Viper and Istorama where the systems rely on relevance feedback are not applicable for M2S. As discussed in section 2.4, relevance feedback is a completely manual task and cannot be applied to an completely automated system. Even if the system is otherwise a program matching our requirements, they are unfit because of the relevance feedback feature. Some systems like COMPASS [114] provide relevance feedback as an optional feature. COMPASS is a non-commercial system that neither meets our requirements due to its lack of a published API. Systems like Riya and Istorama provides a content-based search over the Internet only and therefore cannot be used by M2S.

SIMBA only had online demo of the system with a predefined set of query images. The nature of the images in the predefined set of images was totally different from the ones we wanted to test with. SIMBA might be a good enough content-based image retrieval system, but as we could not test it with our own pictures it is hard to say if it would suite our need. Systems like Blobworld and WALRUS are research project and do not have any demo or API. Only a research paper is available. PictureFinder, Octagon and MascoT allowed users to upload personal images for testing and some of the results from Octagon looked promising, but none of these had an API available. The developer of Octagon considered developing one, but he had no timeline for this and we could not base our work on an extern source like this.

Most of the programs that we tested did not recognize images even if they to the human eye looked very similar. We are not sure why this was so, but we believe the image comparison algorithms are not complex enough. After experimenting with a lot of content-based image retrieval systems, we have not found one system that completely meets our requirements. If the systems were comparing relatively simple images such as a picture of an orange on white background, almost all the systems were able to find similar pictures. If the query image was more complex, like a photo of nature, this was a much harder task. Even the best content-based system came up with a lot of results that were not similar to the query image at all.

The two systems that best fit our use were Lire and Eikon. They performed relatively equally, but at this point of time we had already started to explore Lire so we decided to use Lire in M2S. Lire is easy to use and we believe it is easy to integrate with M2S. As discussed in section 2.3.2 there are three ways content-based image retrieval techniques can be used: query by example, specifying colors and by composing a sketch /drawing. Lire utilizes query by example that suits our use well. Lire also has can also adjust the weighing of colors.

5.5.2 LIRE

Lire is open source, Java-based and freeware program that utilizes the Apache Lucene library and files from the Caliph/Emir program to provide indexing and searching of images (the developer of Lire and Caliph/Emir is the same person). Caliph & Emir are MPEG-7 based Java prototypes for digital photo and image annotation and retrieval supporting graph like annotation for semantic metadata and content based image retrieval using MPEG-7 [24] descriptors. The Lire library a simple way to create a Lucene index of image features for content-based image retrieval. The ImageSearcher uses a Lucene IndexReader and does the retrieval with a linear search in the index. The results are returned as ImageSearchHits object, which aims to simulate a Lucene Hits object. Apache Lucene is also referred to only as Lucene and is a free and open source information retrieval library implemented in Java. However, Lucene has been ported to other programming languages including Perl, C#, C++, Python, Ruby and PHP [97].

Lucene is a powerful, flexible and scalable architecture that enables developers to implement traditional indexing functionality. Lucene is a library, not an application that offers users prototyping and a variety of indexing and search strategies by providing an API. One of the core functionalities within Lucene is the presentation of the “documents”. For every source to be indexed, the appropriate parser is used to create a Lucene document of searchable fields, where each field is constructed using a name-value pair. Lucene allows indexed text to be stored with easy retrieval and display, without having to retrieve the documents from any secondary storage [82] [115].

When indexing, Lucene enables developers to maintain information about the documents such as expression counts such as the frequency and position information such as location of the indexed expressions from a particular document. Lucene lets developers utilize this information to determine the relevancy of queries and ranking of results. The ranking can for instance determine in what order the results are presented to the user [79] [115] [82].

5.5.3 Rating of images in Lire

There is also a weighting of the different matching aspects of the images in Lire. These matching aspects are *colorHistogramWeigh, colorDistributionWeight* and *textureWeigh*. The different weightings should be in notation [0,1] whereas a weight of 0 implies that the feature is not taken into account for searching. colorHistogramWeight defines the importance of overall color of the image and colorDistributionWeight defines the importance of color distribution, which color is where, in the image. textureWeight defines the importance of texture, which edges goes where, in the images [79]. Lire also has a limit for maximum hits in the image index that returns a list of the images that best matches the query.

Together, the weighting parameters described above, except for maximum hits, form a score. When an image comparison is done Lire returns a list of the maximum number of images that is most similar to the query image. Independent of how similar the images that best matches the query are, the list is returned. The images in the list are rated where the best score is 1.0. In M2S we chose to set the limit to 0.4. It is possible to have a list of 10 hits of the index where the score is ranged from 0.1 to 0.3. If the images are rated lower than a predefined limit this is considered a poor score and the query image is probably quite different from the listed images. Therefore no lookup in the information database is performed and hence no results returned. If the image's score is high enough that is reaches the predefined minimum value, the information associated with the image matched is looked up in the information database. The higher the score limit is set, the smaller chance for a result returned to the user but the higher chance that this is actually what the user requested. The lower score the higher chance for a returned message but the higher chance that this is not what the user requested.

M2S does not actually need the list of maximum number of images returned. How many matches found in M2S is irrelevant, only the result with the highest score and the score of this result is interesting. The weighting parameters and the score can be changed if its needs to be adjusted one or several of these and this is done in the source code of the service.

5.6 Communication using MMS

Because a content-based image retrieval approach was chosen, we considered MMS a natural tool for posting the query. Uploading images via web-links was also an option, but this requires more technical understanding of the mobile phone than using MMS. Tools for navigation on web-sites are limited and this makes uploading of images a more complicated task than using MMS. We also believe that most people consider MMS a fun way of communicating.

5.7 Development tools

5.7.1 Hardware

The M2S is developed on an IBM Thinkpad T23 with an Intel(R) Pentium (R) III Mobile with CPU 1133 MHz and 384 MB RAM. The service also functioned as server as it was tested on this machine.

The service is developed to run on practically all mobile phones but is was developed considering a Sony Ericsson W600i with camera resolution 1,3 megapixels and Sony Ericsson K700i with a 0,3 megapixel resolution camera [116].

5.7.2 Software

The operating system used during the development of the system is Windows 2000 and Windows XP. The implementation of the core-service is developed in Java. Because we use a free-available, open source image retrieval function which only requirement is that it is possible to program against, this part is not necessarily developed in Java JDK 1.5.0.07. The system was implemented using Eclipse release 3.1.2 [117]. As we have seen M2S utilizes Lire [79] and Lire utilize the Caliph and Emir project [81] for content-based image retrieval.

The database we have used is MySQL version 5.0.22 [118], which is a free, open source and easy-to-use database. The details of the implementation of the database will be described in chapter 5.8.1.

5.8 Implementation

This chapter will describe the implementation of the service. We will first give a detailed overview of the database and then give describe the implementation of the service.

5.8.1 The database

In cases where the image search finds a similar image, a suitable answer has to be created and sent back to the originator of the MMS. The mapping of images and answers is kept in a database. The database is a very simple one, but it can easily be expanded if it should be necessary [15]. The database is called M2S and it contains information about the attractions, the name of the images and provides a mapping between these. A row in the database looks like this:

M2S

Image ID	MMS	WAP URL

Table 3: The data field in the M2S information database.

The database contains only one table. *Image ID* is the primary key of the database. *MMS* contains the MMS belonging to the image. *WAP URL* contains a website to the specific attraction.

Every attraction in the tourist brochure has a unique name that is the *Image ID* field. The image ID is stored in a database together with the information about the attraction. The information comes in different formats, either MMS or WAP-link, and they have different priority. We want to provide rich, entertaining and good-quality MMS' for most attractions. The MMS can contain series of images and/or textual information and/or a video. We consider this a fun, entertaining and alternative way to receive information by mobile phone. If there exist an MMS about the attraction in the database this is sent to the user. If it does not exist any MMS, a WAP-link to information about that specific attraction is sent. If it does not exist specific information about the attraction a WAP-link to a site containing general information about Lofoten (http://www.lofoten.info) is sent to the user. We will however avoid the last option, because this is not what the users wants and we believe that too many results like this will lead to less or no use of the service.

The way M2S utilize the concept of content-based image retrieval differ from traditional content-based image retrieval services. Traditional content-based image retrieval services searches a set of relevant images that is returned. The number of images returned varies from service to service. M2S searches a collection of images and returns (preferably) an MMS that is mapped to the image matched.

5.8.2 Client side

Both the initiating message and the answer message transmitted with the client is done using already existing functionality at the client side, there is no need for any further implementation at the client side.

5.8.3 Server side

All the implementation done resides on the server side. These are the text files, classes and interfaces used.

5.8.3.1 loadDatabase

This file a text file contains what kind of information the attraction has connected to it; if the type of information is an MMS, a WAP URL to general information or to specific information.

5.8.3.2 properties

This is a text file that contains the properties of the application such as the name of the image retrieval program that is to be used. This file also contains the telephone number used when testing the service because it would normally be contained in the incoming MMS and this functionality is not in place.

5.8.3.3 Lire

This class contains the Lire program. This code is downloaded from the home page of Lire. The class contains methods that creates an index and searches the index. If the properties file contains another name but Lire, for instance Eikon, the Eikon class is loaded instead of the Lire class. This file is part of the Caliph and Emir project [81].

5.8.3.4 LookUpData

This class controls the database. The class is used for the information retrieval when a match is found. This class is also used to load the information into the database.

5.8.3.5 ControlFile

This is the control file of the application. The class holds the name of the images that are to be indexed. This files connects to the database and invokes the main method of the search program that is read from the properties file.

5.8.3.6 SendRecieveMsg

This class handles the sending of WAP sites. As the opposite to transmitting MMS, this is an easy task that required no additional functionality in the mobile network than the ones that already existed.

5.8.3.7 PushClient

This class is a help class to provide the sending of WAP URLs.

5.8.3.8 AbstractDocumentBuilder

This class uses javax.imageio.ImageIO to create a BufferedImage (a Lucene document) from an InputStream. The identifier can be used like an id (e.g. the file name or the URL of the image). This file and the rest of the files listed are part of the Caliph and Emir project.

5.8.3.9 AbstractImageSearcher

This class searches for images similar to the given image using javax.imageio.ImageIO.

5.8.3.10 DocumentBuilder

This class creates a new Lucene document from a BufferedImage. The identifier can be used like an id (e.g. the file name or the URL of the image).

5.8.3.11 DocumentBuilderFactory

DocumentBuilderFactory creates a DocumentBuilder, which will create Lucene Documents from images.

5.8.3.12 ImageSearcher

This interface, ImageSearcher, retrieves images that is similar to the indexed images.

5.8.3.13 TestImageSearcherWeighted

Returns a new ImageSearcher with a predefined number of maximum hits and the specified weights on the different matching aspects. The search can be weighted with a maximum number of returned hits, colorhistogram, color distribution and texture.

5.8.3.14 ImageSearcherFactory

ImageSearcherFactory creates an ImageSearcher, which will retrieve the images from the index.

5.8.3.15 FileUtils

Returns all jpg images from a directory in an array.

5.8.3.16 ImageUtils

Helping class that scales down an image into a box of maxSideLenght x maxSideLength.

5.9 Advantages and disadvantages with M2S

M2S has several advantages compared to similar systems and today’s traditional ways of retrieving information. Most systems also have disadvantages of some kind and M2S is not an exception. Both advantages and disadvantages will be listed here.

5.9.1 Advantages

As with other content-based image retrieval programs there are advantages with M2S related to this. M2S does not need to manage metadata, neither for annotating images or extracting metadata from the image files.

As described in the chapter concerning content-based image retrieval there are advantages to content-based image retrieval. What is referred to as the biggest disadvantage concerning content-based image retrieval is that there is no way to capture the semantics of an image. There hardly exist any tools to capture and describe the happening in the image and the ones that exist are very limited. One of the advantages with M2S is that there is no need to extract the semantics. The query image is formulated the same way as the answer is, as an image, and the algorithm searches for similar technical features of images. The use of a query image makes it easy to understand the principal means how the service works even for the not so technical users. In principal this also makes the service easy to use, but as the content-based retrieval part is very sensitive to its input, this is not the case.

Because the service depends on the tourist brochure this means that as long as the tourist has the brochure in hand he is able to retrieve information from our database regardless of his location, Bergen, Stavanger, Trondheim and Oslo or any location gives results. This is clearly an advantage of M2S. We have not tested the service from abroad, but we believe the Norwegian four digit numbers cannot be reached directly outside the borders of Norway.

If there is a misprint or changes in the brochure after it is distributed all the tourists will be misinformed. It is very expensive to inform travelers later on and this is most often not done. If there are several misprints and this leads to unsuccessful trips for the tourists, the destination can easily gain a bad reputation. M2S can help prevent such consequences. The brochure can be changed to only contain images and core information of attractions like name and what the attraction is about and leave the detailed information on the database that can be retrieved for users of M2S. This will be an economical and environmental friendly solution. This means that the tourists can retrieve updated information about opening hours, availability and prices about the attractions. This requires that the database must be kept up to date to be successful.

If the query were text-based there would have to be some functionality to extract a textual formulation of the semantics of the image. As discussed earlier annotation can be a source of much errors and it is a time-consuming task. It also requires that the images are annotated in some way to retrieve the images by textual queries.

We believe this is a fun way to retrieve information and as long as we provide correct, interesting and rich information this will be a service that tourists want to make use of.

5.9.2 Disadvantages

As mentioned among the restrictions M2S is dependent of the tourist brochure. As the tourist brochure provides mobility and independency from several actors, it is also a disadvantage in the way that the user must have the brochure at hand. The brochure is expensive to distribute in large scale and cannot be uncritically distributed.

Most of the problems with M2S are related to the content-based image retrieval program. We discovered several disadvantages with content-based image retrieval. The light conditions on the image taken and the indexed image had to be almost identical. Even if the light conditions were relatively equal, the program was not able to recognize the image. We experienced the same difficulties with the resolution. If the two compared images had just the smallest differences in resolution the images was not considered equal. To make sure that the search program does not exclude otherwise similar images, we have to adjust the images so all have the same resolution. This can be done in two different ways. One way is to inform the users that the images taken meant for M2S must be of one specific resolution. This is not a very good solution while not all camera phones come with settings for this.

The users must also somehow be informed of the current resolution. It can of course be printed in the brochure but that means more room for errors. It is easily happen that users overlook the information about the resolution and this can result in unsuccessful searches.

The easiest and best way of solving this is to automatically adjust the resolution of the image when it arrives at the M2S server to match the resolution of the images in the database.

The short number and service name must be remembered or printed in the brochure. This can be a major challenge while there is much competition among services that are based on interaction with a short number. There are however solution to help people remember the service number. Digits can be formed as easily remembered phrases that mentally replace the digits of the number with letters. For instance does Rema 1000 in Norway use phone “number” 0REMA instead of number 07362 as the former is easy to remember while the latter is not.

5.10 Alternative solutions to the content-based approach

The service was going to be fun and easy to use and give good results. We also wanted to experience content-based image retrieval by utilizing and testing this technique. We believed it would be advantageous that it is ready for use and users do not have to install anything in beforehand. Content-based image retrieval has been discussed relatively much lately (especially after Google bought Neven Vision[3]) that inspired us to want to experience content-based image retrieval.

We had several alternatives of what search criteria we were going to use that enabled us to provide the user with the best experience possible. Options considered included the use of number codes, bar codes, context-based, content-based, text-based or combinations of these.

[3] http://www.nevenvision.com/ and http://picasa.google.com/

5.10.1 Bar code

A bar code can be placed in a top corner of every image in the tourist brochure for the program to recognize. When a picture of a picture in the brochure is snapped and used as query, the bar code can be used as basis for the search instead of motive on the picture. A bar code reader could be used but this is not very practical. Instead we could use the bar code as basis for the content-based search. The search functionality would still be dependent of a clear image that brings out the bars in the bar code, but the search is much faster due to its black/white appearance and simple structure.

There are some advantages to this approach. The search time is heavily reduced compared to a traditional content-based search. As we discussed in the background chapter traditional content-based image retrieval is quite resource demanding.

Assuming that the bar code is read correct the search with bar codes are also more precise than today's content-based retrieval techniques so the possibility that it will return relevant results is much higher. This is possibly the strongest argument for bar codes because the probability that it would lead to satisfied users is much higher than the probability that the content-based search would.

Bar codes are not only advantageous. For the service to return correct results it is crucial that the user includes the bar code in the picture. If the bar code is partial or completely left out of the image, the program has no foundation for the image comparison and therefore cannot return any relevant information. Again the service will be completely dependent of the brochure and future versions of it.

5.10.2 Unique numbers

A number code can be used the same way as bar codes if it is uses within content-based image retrieval. It is a unique number placed in a corner of the image that is to be recognized by the program. Depending of the clarity of the image, this approach can easily be more accurate than bar codes.

A text-based approach where the user must type in the unique number, is also applicable here. An advantage is that a number can to a certain degree be remembered by a user so this is a more flexible solution than bar codes.

A bar code is more dependent of the brochure than the use of unique numbers. With the use of unique numbers it is possible to free us from the tourist brochure. Each unique number would map to an attraction. A bar code is a too long sequence of numbers for normal people to remember or bother to type. A unique number would (for this demo) consist of about 3 digits and is non-problematic to type in to the mobile phone. It is then possible for the user to type in the number directly into an SMS or MMS and receive information based only on the number.

In this way the service can be used without the brochure, but the users must somehow be informed about what number corresponds to what information. The tourists can for instance retrieve this information at special places like the tourist information office or at the destination itself. This makes our service less mobile and flexible: an approach like this moves away from the concept of delivering information there and then when the user want it.

5.10.3 Text-based information retrieval

Text-based image retrieval is probably the most traditional and most used way to retrieve images. When it comes to information retrieval users are relatively impatient and they most often want the information as soon as possible. We believe that much of the reason for the success of text-based image retrieval is that the search is done very fast. It is also a method that is easy to understand and easy to use. Most often the user just types in keywords describing the image and receives results of varying relevance.

5.10.3.1 Specific keywords

The user can type in keywords describing the attraction in an SMS and send to number 2034. Based on the keywords the user can retrieve information about the attraction. Varying combinations of the keywords gives different results. The keyword describing the attraction can be name, what it is, location etc. One combination can be keywords like viking, museum, Borg and ship. Based on these keywords the user can retrieve a video from the Viking Museum demonstrating how the Vikings acted/operated at sea. Another combination like viking, museum, Borg and weapons can result in a video of how the vikings made weapons and how they were used. These are quite advanced combinations and we believe that the users must be somehow told what keywords it is possible to combine. The keywords could be printed on a board together with information about our service at the tourist information office or at the attraction itself. The moment the users are located at the attraction the need for information about the attraction from extern information retrieval service decreases. Visiting attractions such as the Viking Museum at Borg one expect to receive the desired information about the museum and the vikings during the visit. Users visiting other types of attraction, such as the whale safari have a larger benefit from M2S. At the departure place for whale safari, it would be possible for users to watch a video from whale safari to decide whether to join the safari or not.

The use of specific keywords printed on boards makes the service dependent of a board that is not portable. This makes the information retrieval the opposite of the solution we want to develop: a mobile and flexible solution.

5.10.3.2 General keywords

The users do not necessarily need to be informed about the combinations of keywords. It is of course possible to think logically and guess keywords that describe a specific attraction but then we cannot in the same way guarantee relevant results. It is logical that the combination *galleri and Espolin* Johnsen could return a video guiding the users through the Gallery Espolin. It is not however so logical that the combination *field trip and mountains* would give any results. We are facing the problems that are typical for text-based image retrieval. Espolin Johnsen is concrete keyword that has no synonyms and most users would intuitively use the name of the artist in a query. *Field trip and mountains* on the other hand has synonyms and can easily result in a miss in the database despite that there were results, but the use of wrong keyword did not trigger them.

A text-based image retrieval approach has sources of conflicts. It is more time consuming to type in the keywords describing the attraction, than to take a picture of it. It is also more time consuming to type in the keywords in a foreign language than the users first language. It can easily arise type errors that can result in a response from the database that it would not normally give. Another source of conflict is if the user has a different keyboard than the Norwegian one, (i.e. the Russians use the Cyrillic alphabet) and has to switch the language on the mobile phone before they can use our service. Since we in the future want to completely break free from the tourist brochure, we do not want to be dependent on other written information. This is the approach that CAIR use. CAIR will be described in chapter 6.

5.10.4 Combination of content-based and context-aware image retrieval

As we saw in the chapter concerning related work, it is possible to combine several concepts to achieve image retrieval. SnapToTell [38] combines content-based image retrieval with information about the users context. The context they use is location. Like SnapToTell it would be possible to add context, such as location, to the search to narrow down the number of possible attraction the users want information about. The use of location requires an approach to determine the user's location. The most applicable strategy would be to use GPS. In chapter 2 we introduced a mobile phone, Nokia N95 that comes with GPS. The query image could be automatically added with the location information before it was sent to the service.

The service would use the location information to define the area where the user and the actual attraction reside. The service would then compare the query image against the images of attractions in the database that have GPS-coordinates ranging within the area provided by the users GPS.

There are clearly disadvantages with this technology. Let us say that the radius of the users position is set to one kilometer and all attractions within this range is considered the actual attraction. GPS can give wrong position information, if the position of the user is measured incorrect, the selection of possible attraction would be wrong. The actual attraction would maybe not be considered at all and clearly, information about the attraction would not be returned.

5.11 Testing

The next sections will discuss the testing done on M2S. We will discuss the testing compared with testing of other systems. Problems encountered will also be discussed here.

5.11.1 Testing of M2S

When testing this service we had a limited selection of images and associated information to these images.

If a service like this is going to be a success, the response has to be quick and precise. One of the core needs we want to meet is the user's demand for information there and then when he needs it. If he sends a request for information and the service is slow responding, the user's location and interests might have changed since he started the interaction, and the information delivered might now be uninteresting to him. This is something that we want to avoid. Since this is a alternative and fun way to retrieve information and not a service that is strictly necessary, we believe that this must be a reasonable service and will rather calculate on frequently use.

During the testing we experienced that to retrieve hits in the image database using a query image, the image practically identical to the matching image in the database for the service to find a match. As we mentioned earlier the light conditions when the image was taken and the resolution of the camera had major impact of what results we retrieved. Very different light conditions resulted in far from similar images. Little light could result in dark and unclear images that did not reflect the colors of the query image in the brochure. Too much light could give reflections in the surface of the image photographed. The same thing applied for the resolution. When we had similar light conditions as the images in the database and more alike resolution we got more precise hits.

Because the functionality for transmitting MMS was not working, we never got to test the service as whole. For testing the rest of the service we transmitted the picture photographed with the mobile phone to the server by Bluetooth. The image was then compared against the image database and the belonging information was fetched. The functionality for WAP was in place, so we were able to send WAP messages as result to the initiating mobile phone. To send a response to the client the telephone number of the client had to be known to the service. Normally this number would be contained in the incoming MMS. Because the functionality of MMS was not in place, the telephone number instead had to be fetched from a properties file for testing purpose.

5.11.2 The testing compared to other systems

Performance evaluation is often a neglected topic of content-based image retrieval research [119]. Hard information on the performance of automatic CBIR techniques is difficult to come by. It is difficult to say anything about how effective Lire compared to other content-based image retrieval systems while there has not been performed any comprehensive comparative evaluation studies [1]. The testing that is done is often carried out on small fixed sets of images where the visual contents of the images are simple. This restricts their prime usefulness to specialist application areas such as fingerprint matching, trademark retrieval or fabric selection. Within these applications, content-based image retrieval technology appears to be capable of delivering useful results [1] [119].

5.11.3 Problems encountered

We encountered several problems during the development of M2S. The biggest challenge was to implement functionality for transmitting MMS. At first we planned to use the PATS-lab that has an interface for sending and receiving MMS. PATS is a research agreement between the Norwegian University of Science and Technology (NTNU) Department of Telematics, Ericsson, Telenor and Hewlett-Packard [86]. PATS is among other things a test-bed for projects like M2S. PATS have a node in Tromsø that we can communicate directly with. We were told that to send MMS via

PATS would not be a problem, but the problems would appear with receiving MMS. There is a lot of traffic on these four-digit numbers and the services has unique service names or codes that separates the MMS' to make sure that they end up in the correct service. There was however no support for service names/codes in PATS and we had therefore no way of filter out the MMS' that were meant for our service. This could be solved if M2S got its own unique four-digit number, but this was not an option. We believe it was because PATS manages a very restricted amount of these special numbers. It is important to notice that PATS do function for systems that are commercially available.

The communication is a very crucial part of M2S. Without functionality for sending and receiving MMS we would not have any service. A company called Geomatikk, a former subsidiary of Telenor, could maybe provide us with tools for transmitting MMS so we could make a solution that worked. At the time the summer project was finished the tools for developing functionality for transmitting MMS was not in place. However, at the time this thesis is finished the functionality is in place and as far as we know, it works well. This functionality was developed by Schürmann et al. [15].

5.12 Future work and the potential of CBIR

To have a well working content-based image retrieval program has an enormous potential within several areas, but particularly in the field of web shop this will be especially applicable. Lets imagine that we in the future can take a picture of a handbag or a pair of shoes and retrieve these from the Internet, without having any further information about the article like the item number. The article might be taken out of production, is not available in normal stores and therefore hard to find. We imagine further that users can get information about where it is possible to buy this article and for what price.

Like Microsoft, Google [31] has several projects that deal with content-based image recognition. Google has bought Neven Vision to enhance the service Picasa. The vision is to make users able to take a picture of a movie-poster, send it to a specific number and based on the picture they can retrieve a movie trailer, localize the nearest cinema and order tickets. A hit in the database can trigger other services like downloading ring tones, price lists and map services [120].

If we see this in a tourist perspective it would be possible for the tourist to snap a photo of whatever freestanding attraction and submit as a query for information. It does not have to be limited to constructed attractions but can for instance be landscape or mountain formations. Actors like Google works with a solution like this.

The concept of retrieval of information that has similar features of the query information can be applied to other fields like sound. It is possible to recognize parts of a soundtrack and provide users with information about this soundtrack. This concept easily applies to other areas like for instance the music industry. The introduction of portable MP3-players and specially the integration of these into mobile phones make it easier for users to listen to music. People bring their mobile phone and also their playback unit. Portable CD-players used to be the best alternative for mobile music, but these do not have the same functionality to record music as today’s music playback units. The usage of this concept can be illustrated with an example.

A user is sitting on the bus as is about to turn on his MP3-player combined with radio. When turning it on the song that is playing on the radio just ends but he recognizes the song from earlier playbacks. He likes the song and would like to purchase it but he do not know the name of the song. How is he going to find the soundtrack at the Internet where he can buy it? He can of course go to the nearest music store and sing or hum parts of the soundtrack hoping that the staff recognizes the soundtrack but this is wanted by far from all people. Instead he can record the last part of the song and send it to a specific service. The song is recognized and he can for instance get an MMS with information about the song such as the name of the artist, the name of the actual album and maybe other albums that the artist has made. Best of all, he can get a web-link to where he can download the song and the music video and to what price. Wang [121] describes the Shazam music recognition service that is based on this type of technology.

5.13 Summary

In this chapter we have described the information retrieval service M2S. We have seen how M2S fits in a tourist setting. Most tourist seek information while traveling and M2S can offer rich and entertaining videos and images and provide tourists with web-links to additional information about Lofoten. We have seen that the dependency of the tourist brochure, the Lofoten Info-Guide, is an advantage and disadvantage.

We have described the design and architecture. M2S has a modular design that makes exchange of components as easy as possible. We want exchange of one component to have as little impact as possible on the rest of the service. Regarding the architecture we have described what development tools we have used and how the database is constructed.

We have described the implementation and what classes we have used. We also discussed concepts like reflection that we have made use of when including Lire in M2S.

The service has not been tested properly yet while there was not yet developed a test bed for this at Telenor R&I. The testing was limited due to this, but we did a workaround to test the service minus the communication parts. The service as it is now with the Lire program would not be a complete service. The image retrieval part is very limited due to its sensitivity. A user cannot snap a picture of a picture in the brochure without consider the light conditions and resolution of the picture.

Alternative solutions to the content-based approach and advantages and disadvantages related to this have been discussed. We have seen that the use of bar codes and unique number could help improve the service. The use of text is also a solution that is possible to use in M2S. We decided to use an existing content-based program and during the process of finding one we got personal experience with several content-based image retrieval programs.

We encountered problems during the development of the service and we have described these issues. The problems were mainly concerning the content-based image retrieval functions and the transmission of MMS. We have discussed future work and the potential of content-based image retrieval. We have seen that when more adequate techniques for content-based image retrieval exist, the fields of use will be several.

We will give an evaluation of M2S in chapter 7 together with an evaluation of CAIR that will be described in the next chapter.

6 CAIR- a context-based image retrieval service

We imagine that CAIR can be used to retrieve images from an image collection. The retrieval is accomplished by sending an SMS that contains a textual query. CAIR utilizes concepts like context and mobile environment that are well-used. We want to combine these concepts into one solution that, hopefully, has not been investigated before, that is using SMS to initiate the image retrieval. In this chapter we will discuss the motivation behind CAIR and introduce the service. We will then discuss different ways location can be used to retrieve information. CAIR is designed but it is not yet implemented. A detailed design overview will be given together with a description of how we want to extract the metadata from the images. The section concerning future work will give a plan of how the service can be implemented. We will then discuss advantages and disadvantages.

6.1 Introduction

As we have seen, many of current image retrieval techniques are insufficient in the way that they often do not meet the user's query. What is searched for is often not what is retrieved. There have been several attempts to provide users with services that retrieves exactly the information the user wants and little or preferably nothing else. This is however much harder than it appears. One of the projects that try to resolve issues related to this is the CAIM-project. The CAIM-project aims to "develop methods and tools for context aware image management in distributed, multimodal and mobile environments" [5].

With the use of context information to support semantic-based image retrieval the CAIM-project [5] aims to tighten The Semantic Gap. A goal of this thesis was to design a system that enabled users to retrieve images from a specific location and point of time to his mobile phone. Using mobile phone or retrieving images does not bring anything new, but using SMS to retrieve images this way has, hopefully, not been done before.

6.2 Motivation

M2S and CAIR share much of the same motivation. As with M2S, the motivation for CAIR is that mobile phones with image capturing functionality is brought almost everywhere. This enables people to capture events and create memories in a spontaneous way and it provides a great opportunity for users to exchange images while on the move. As mobile phones with an cameras have become more common, storage capacity on the mobile phones are much larger and the transfer rate increases, people share images between these devices in a larger scale. This requires an efficient way of retrieving images from a collection.

In chapter 3 and 5 we discussed M2S and problems with content-based image retrieval. As discussed, the biggest challenge to content-based image retrieval is that there are few and very limited ways to capture the semantics of an image. While textual queries are the most common way of retrieving images and is based search on metadata, this is a problematic combination. In this chapter we introduce another concept, called context-based image retrieval that aims to solve some of the problems that content-based image retrieval brings.

As discussed in chapter 2 the amount of context-aware systems dealing with location information is increasing due to the growth of mobile phones in the telecommunication market. Some examples of location-aware applications are the variants of digital tourist guides where location dependent information is displayed to the users. In section 2.8 we introduced such systems and in chapter 3 we described Cyberguide quite thoroughly.

The motivation was to make a program that utilizes context to retrieve images from a image collection using SMS.

6.3 The CAIR service

CAIR is context-aware in the way that it utilizes the time and location of the user the moment the user captures an image. When the image is snapped it is automatically annotated with information about time and location. When the image is added to the database, image information is extracted and added to its respective column and is used for later retrieval. The user formulates a textual query in an SMS describing the location and the time he wants the image series from together with the service name "CAIR". The SMS is sent to the short number 2034 that is a message server managed by PATS. PATS's message server forwards the SMS to our service, CAIR. The service searches the database for image's context matching the specific location and time. If matching context is found, the corresponding image or images is added to an MMS and sent to the user. The MMS contains a series of images from the location and time the user requested. The database is also searched for WAP URLs. If there is a corresponding URL this is added to the MMS for users to enter after watching the image series. The URL provides the users with additional information about the location and possibly how to buy tickets, opening hours etc. There cannot be a WAP URL in database if it does not have an image associated. If there is no match on image's context in the database an SMS is sent to the user explaining that there was no images found matching the query context. The use of many types of contexts is possible, but in this thesis we use location and time

The context in form of metadata has on beforehand been extracted from the image and added to separate column the table in the database. The user can receive a maximum of n number of images. We believe a suitable number of received images are approximately 12 images. An SMS request might look like Figure 30.

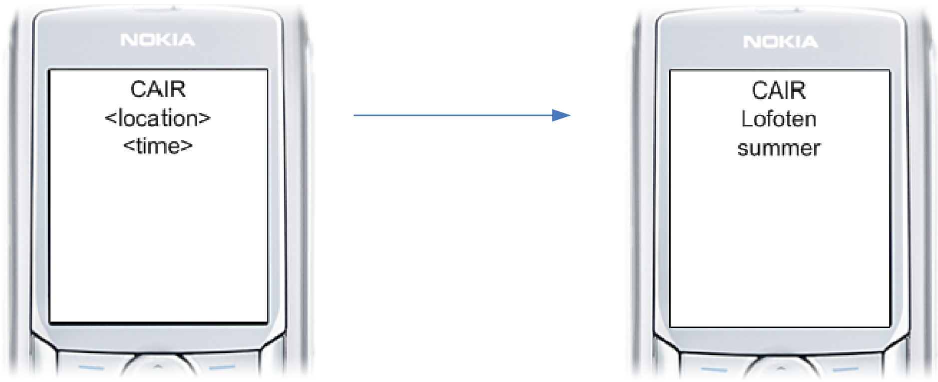


Figure 30: What the initiating SMS might look like.

The request SMS must contain a service number so the message server can sort the message to the correct service. This is illustrated in Figure 30.

The <location> parameter is required. If no parameter for location is set, the service will not know what location the images shall be from.

The <time> parameter is optional. If this parameter is set, the images will be from the current point of time. If there are no images in the database from the specified time, the user receives images from the location, from random periods of time. If no parameter for time is set, the user will receive a mix of period of times. It can be a combination of images taken in summer time, wintertime, autumn and spring; some of the images can display the sunrise and some images can be taken in the evening or in the night.

If the database does not contain images from the specified point of time, but it contains images from the requested location but from other periods, these images are sent as MMS to the user instead. The MMS also contains a message that explains that there were no images from the requested point of time, he is offered images from the same location but from other periods. The MMS will also contain a message that explains that the image database did not contain any images from the requested period of time.

If there are no images in the database, the user receives an SMS explaining that there were no images found. Similar to M2S this is a case that users are not satisfied with, so we want to avoid this case. The service can of course be designed with more options, like for instance an option for a maximum or minimum amount of images a user wants. However, this makes the service harder to use because there is more for the user to remember. In this design we use the service name, the location and time parameter.

6.3.1 Information flow

Figure 31 illustrates the information flow in CAIR. The figure shows the progress in CAIR depending on the events. The preferable progress of M2S is following the horizontal line of events, meaning that an MMS with associated WAP URL is returned.

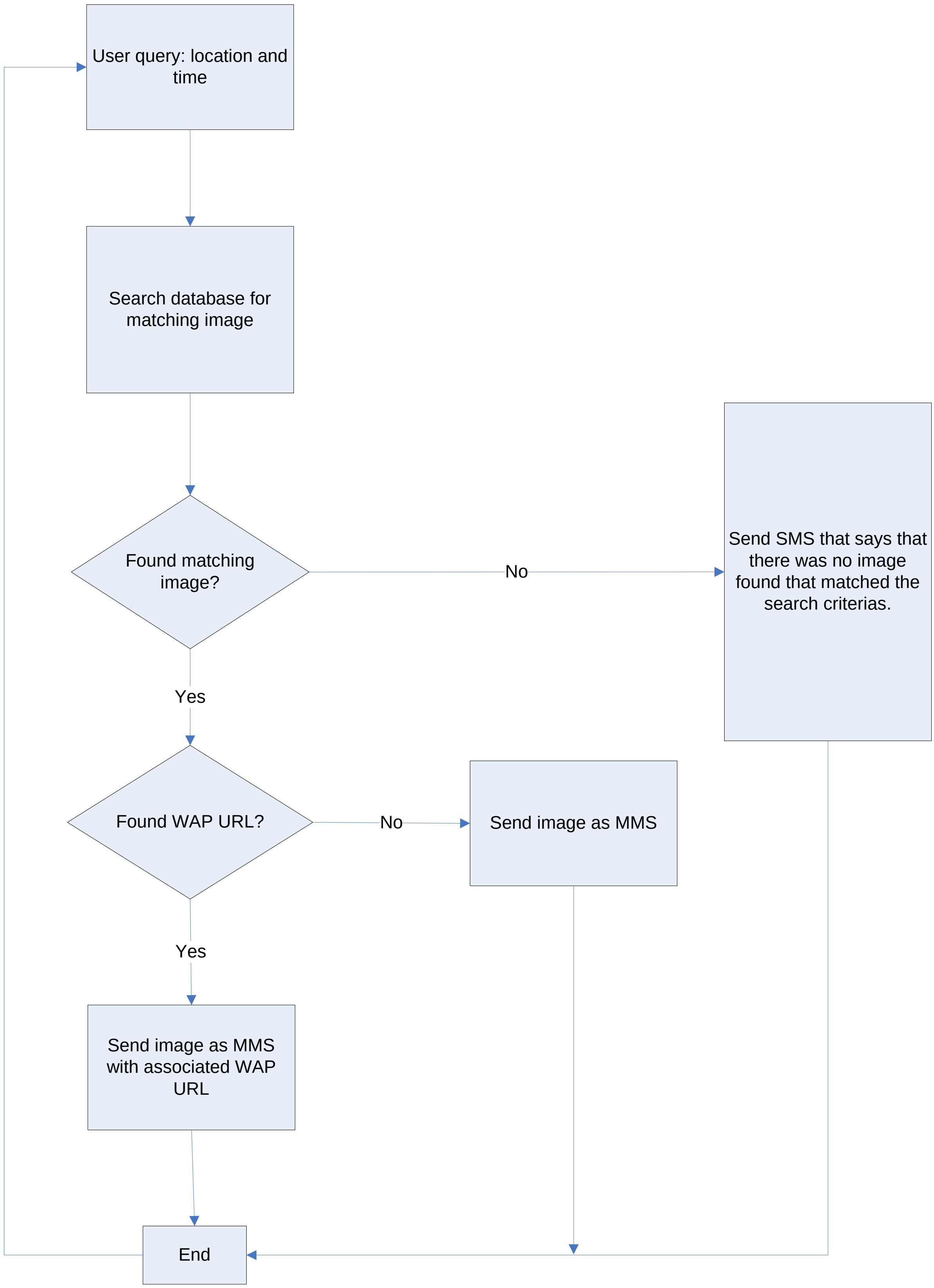


Figure 31: Information flow in CAIR.

6.3.2 Restrictions of the service

CAIR is designed for mobile entities and specially mobile phones. The mobile entity must be equipped with functionality to transmit SMS, receiving MMS and WAP URLs and display images.

CAIR is designed to use time and location as context. It is possible to use other types of context too, but we have limited the use context to include these two types. The service is limited to use the images in the database. It is therefore crucial that the database contains nice pictures that the users find interesting.

The use of MMS can also be disadvantageous when requesting information. Most users of search engines use relatively powerful computers that allow constant retrieval of web-links. If a search does not return the expected results, a new query can easily be performed without any further costs. This does not apply for the use of MMS. In CAIR, if a query returns unexpected results, the user maybe tries another time, but if this does not either return expected result he will probably not try the service repeatedly.

6.4 In what situations is our service useful?

We hope that a service like CAIR could be useful for most people. When people decide to visit places this decision is often taken based on what information they have about the location. The information is often formulated as images. One major advantage with images as information is that they are independent of language. If you speak English, Norwegian, German, Mandarin, Swahili or other languages an image does not have to be translated. This does not mean that images are unambiguous, but we believe that most people would find an image of the midnight sun, the mountains in Lofoten and the northern light beautiful and attractive.

CAIR provides a general service that provides information about any entity or location. The information provided suites tourists as well as the local population. The usefulness of the service is best elucidated with a user scenario.

6.4.1 User scenario

Two friends from Oslo are visiting Lofoten in summer time. So far they have enjoyed their stay and they have visited several tourist attractions and participated in many activities, such as the whale safari. The weather is perfect; it is sunny and it is a gentle breeze. At the moment they are a bit short of time and they are wondering whether to stay in Henningsvær or to travel towards Sortland.

It is almost impossible to visit Lototen without noticing the spectacular mountain formations. They enjoy sports and both have tried mountain climbing before.

They just heard about a particular mountain called Svolværgeita. They just met a couple that had been there. The couple had taken some photos that made the mountain look tempting. Because they are a bit short of time they want to set their priorities right. To decide whether to stay or travel further they would like to see some more pictures of Svolværgeita.

They compose an SMS with “CAIR” “svolværgeita” and “summer” and send it to a short number. The service handles the request and returns a series of images from Svolværgeita in summer time. The two friends find the images very satisfying. It looks steep and challenging, but no more than they can handle, so they decide to go.

Figure 32: Svolværgeita in summertime.

Figure 33: Shows the URL attached.

The WAP URL in the end of the image series links to a site where they get information about guided climbing in Svolværgeita. There are places left for today's tour, and via the WAP URL they buy tickets and reserve mountain climbing gear. They also receive information about weather conditions and suitable clothing.

The example is from Lofoten, but this service is not restricted to Lofoten. As opposite to M2S, CAIR is not dependent of any brochure. As long as the information is in the database it can be retrieved.

6.4.2 User requirements

The mobile phone must have functionality for transmitting SMS, receiving MMS and WAP. It must also have functionality to display images.

As with M2S it would not be required for users of CAIR to download or install any software on the client entity. All the functionality needed such as transmitting messages is included in the majority of mobile phones. As discussed in section 5.4.2, downloading and installing software can be a threshold for many people and it is a clear advantage of this service that no software has to be installed.

6.5 Design and architecture

As mentioned in the introduction to this chapter, CAIR is not yet implemented. However we have made a design overview of the service CAIR and have planned some details regarding the implementation. The design and plans for implementation will be given in this section. The solutions that we have chosen and plan to implement are inspired from the design and implementation of M2S.

Figure 34 shows an overall architecture of CAIR. The figure illustrates transmission of messages. The user initiates a query and sends it to a short number. The short number is directed to a message server at for instance Telenor. The message server sorts the message on the service name and the query is forwarded to our service. The query is handled and a proper response is composed. The answer is sent, via the message server, to the mobile phone with the initiating SMS. Figure 35 shows a detailed description of the blue shaded box of Figure 34.

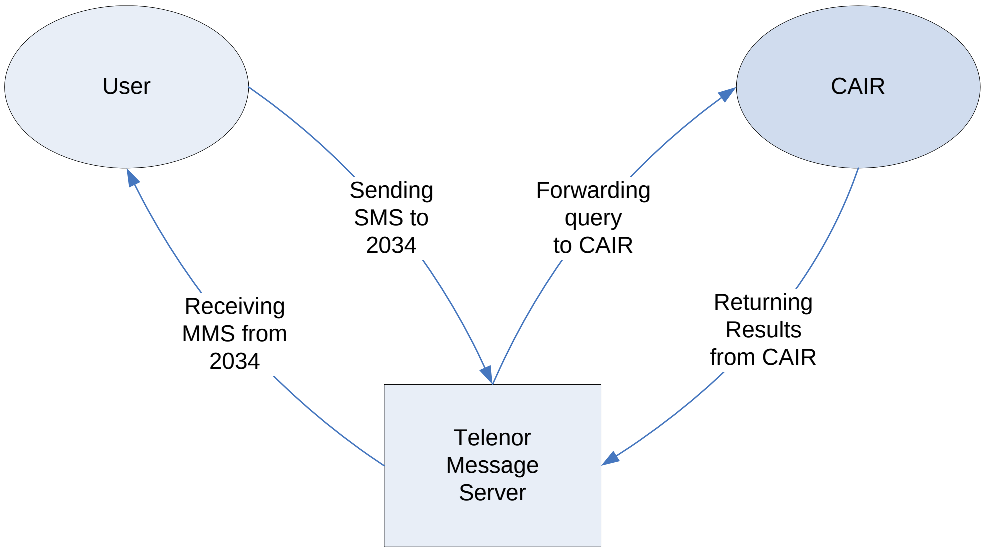


Figure 34: An overall description of CAIR.

As with M2S we want to make a modular design for easy exchange and updates of modules. The functionality for mapping the notations of locations, the database and messaging functionality can be easily exchanged. In Figure 35 the blue shaded boxes illustrates the main progress in the service. The dark grey-shaded boxes illustrate the mapping of location notation and the white boxes illustrate the information retrieval functionality.

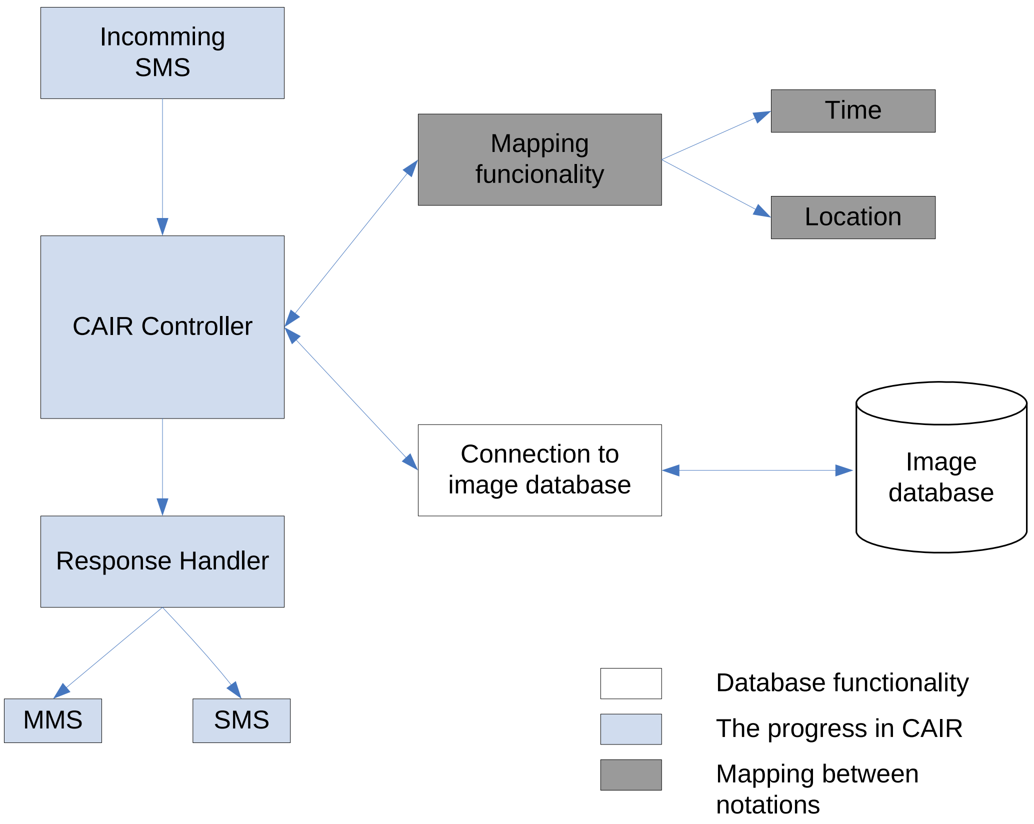


Figure 35: Shows a more detailed description of CAIR.

In M2S the MMS' are pre-made. They can be contained in the database ready to be sent to the user if the MMS is the answer for a query. In the CAIR database the MMS' can be made on the fly when suitable images are found. If a series of, lets say 12 images are found in the database, these are wrapped as MMS and sent to the user.

6.5.1 Communication using SMS

Because the query was context-based (text) using SMS was a natural choice of communication medium. Basing the communication on SMS is convenient because most people are familiar with sending SMS. SMS is also cheaper than for instance MMS.

6.5.2 Image database

The database should contain nice pictures that users find valuable. We want the image database to grow quickly and soon have a broad collection of images with associated WAP URLs. The database will be called CAIR and contain information about images and information associated with them. We imagine a table in the database could look like this:

CAIR

Image ID	Image	WAP URL	Keywords	GPS coordinates	Time

Table 4: Illustrates the database of CAIR.

Image ID is the primary key in this database. *Image* contains the images and *WAP URL* contains the URLs mapping to the images. *Location* is the textual annotation of location. *GPS-coordinates* and *time* are the context extracted from the image's metadata.

To add rich images, associated metadata and URL is a time-consuming task for large databases. To quickly achieve a large and heavily equipped database we want the information in the database, images with associated metadata and URL's, to be uploaded by users. As users take interesting images these are uploaded to the database. If the images have a relevant URL, this can be uploaded too. The metadata that is automatically added to the image, time and GPS-coordinates, if available, are extracted from the images. The images are then annotated with keywords describing the location. The images are then annotated with keywords describing the image. As discussed this can be done manually, automatic or a combination of both.

There are problems related to let everybody have access to the uploading functionality. The database can easily contain images that are not interesting for the general user and the images can easily lack or have a misleading annotation. Because users often do not bother to annotate images for private collections, we do not expect users taking the time to be exact when annotating images in a public image collection like CAIR. The database can also easily contain irrelevant, offensive, illegal and/or strictly commercial URLs. Controlling this manually can easily become time-consuming as the database becomes larger. Controlling these issues automatically is problematic while some actors often have creative solutions of notation that is hard for software to detect.

The users can be obliged to establish a user profile protected with username and password. This is a practical way of controlling the access to the service. If information of the unwanted type is uploaded it can easily be detected from what account the information was uploaded and consequences can be made. Another way to control the uploading of information is to let trusted groups be responsible for this. These groups could be travel associations such as Destination Lofoten or The Norwegian Travel Association. As these groups take interesting pictures they can be uploaded to CAIR and then be shared with users. To let trusted parties control the uploading of information is a way to make sure that also the URLs are appropriate. It is also important that the URLs are relevant to the images and not URLs linking to strictly commercial sights that have nothing to do with the images.

6.5.3 Extracting metadata of the image

Most digital cameras now use the EXIF format for storing metadata [26] and is the standard that we have in mind when designing CAIR. There exist several libraries for extracting metadata from image files, but because we plan to implement CAIR in Java we want to use a Java class called ImageInfo to handle this part [122]. This is some sample code on how to use the class. The implementation works and parts of the output can be seen below.

```
package test;

import java.io.File;
import java.util.Iterator;

import com.drew.imaging.jpeg.JpegMetadataReader;
import com.drew.metadata.*;

public class Test2 {
	public static void main (String [] args) throws Exception {
		printMetadata (new File ("c:/test.jpg"));
	}

	// This approach reads all types of known Jpeg metadata in a single call
	public static void printMetadata (File file) throws Exception {
		Metadata metadata = JpegMetadataReader.readMetadata (file);
		Iterator directories = metadata.getDirectoryIterator ();
		while(directories.hasNext ()) {
			Directory directory = (Directory) directories.next ();
			Iterator tags = directory.getTagIterator ();
			while (tags.hasNext ()) {
				Tag tag = (Tag) tags.next ();
				System.out.println (tag);
			}
		}
	}
}
```

The tags hold the EXIF info. A sample of the output looks like this:

[Exif] Model - Canon DIGITAL IXUS 430
[Exif] Orientation - Top, left side (Horizontal / normal)
[Exif] X Resolution - 180 dots per inch
[Exif] Y Resolution - 180 dots per inch
[Exif] Resolution Unit - Inch
[Exif] Date/Time Original - 2007:03:20 16:22:01
[Exif] Date/Time Digitized - 2007:03:20 16:22:01
[Exif] Shutter Speed Value - 1/158 sec
[Exif] Exif Version - 2.20

The EXIF info holding the time is highlighted. If GPS becomes standard in digital cameras and mobile phones, the position information can be added to this list.

6.6 Metadata notations

In the next subsections we will discuss metadata notations for location and time.

6.6.1 Notations of time

Practically all image capture devices today have functionality to tell the time. The time is automatically annotated the image when it is snapped. When the image is added to the database, this metadata is extracted from the image and added to a specific column in the database. The point of time when the image was captured is now searchable. The notation of the time is very precisely, for instance 12.May 2006, 12:20:12.

Queries are most often not this accurate. Common queries when requesting a point of time could be: "winter", "August", "12th May", "daytime" and "night".

Because there can be inconsistency between query and the metadata, there has to be a mapping between points of time. "August" and "12th May" are unambiguous. Other notations, such as "winter", "daytime" and "night" must be defined. An example of mapping between dates and seasons is shown in Table 5.

Time of year	**From date**	**To date**
Summer	16th May	15th August
Autumn	16th August	01st October
Winter	02nd November	31st March
Spring	01st April	15th May

Table 5: Mapping of dates and seasons.

Table 5 shows an example of mapping between dates and seasons. There are several issues related to this mapping. If a user types in "summer", this is not as unambiguous as it seems. What is "summer"? From what date and to what date is defined as summer? If an image in the database is annotated with the date 20th August, is this summer or autumn? This will also vary from what part of the country the image is taken. The autumn comes earlier at northern parts of Norway than the southern parts. At 20th of August it may be autumn at Svalbard, but still summer in Oslo. It can easily snow in Tromsø on 17th of May even if this is in the above table is defined as summer. If the user wants images from Tromsdalstinden at summertime, images from Tromsdalstinden with autumn colors and leaves falling from the trees is not what the user requested. If a user snaps photos in Australia, which is situated below the equator, in Australian summer time and ads these to the database, these images will then not reflect the Norwegian understanding of summer. The same thing applies with images snapped with cameras having the wrong settings for time. Some of these issues were discussed in section 2.7.5.

6.6.2 Notation of location

Some image capturing devices, such as Nokia N95, are equipped with a GPS and have functionality to add the current GPS-coordinates to the image's metadata the moment it is captured. A GPS can determine the user's position with small margin of errors. When the image is added to the database the GPS-coordinates, if they are available, are extracted and added to a specific column in the database. The location where the image was captured is now searchable. However, typing a GPS coordinate manually to an SMS is not consistent with the human way of formulating queries. Human ways to post queries is for instance "Tromsdalstinden", "Svolværgeita" and "church in Tromsdalen". The images therefore must have metadata attached that can be searched. The user who added the image can now add metadata to it. As we have discussed earlier few users take their time to annotate image collection. If this should be the case there are two ways to automatically annotate the image with metadata:

- **Mapping of GPS and textual description.** Check if the image GPS-coordinates maps to a textual description of the location. If there is a mapping, the location information is added to the image.
- **Inheriting images from parent image.** If there is no mapping between GPS-coordinates and location the image can inherit metadata from "parent" images that the image is within the range of.

6.6.2.1 Mapping of GPS and textual description

Lets say that an image has a GPS-coordinate X. This GPS-coordinate of the image is given a radius. If the image's radius is within another image's radius the image adds this image's metadata to its own metadata. This metadata might describe the image well or it might not describe the image at all depending on how big the radiuses are set and how much of this area they have in common. Figure 36 and Figure 37 illustrates the differences.

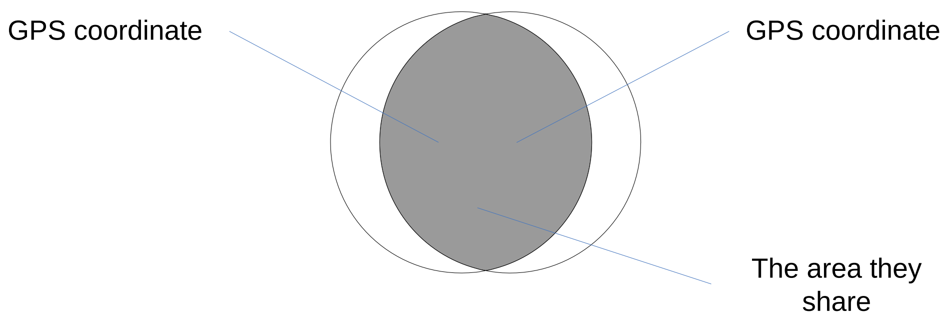


Figure 36: Images sharing a large area.

Figure 36 shows images that share a large area.. Much of the metadata shared between these images may be very describing for both images.

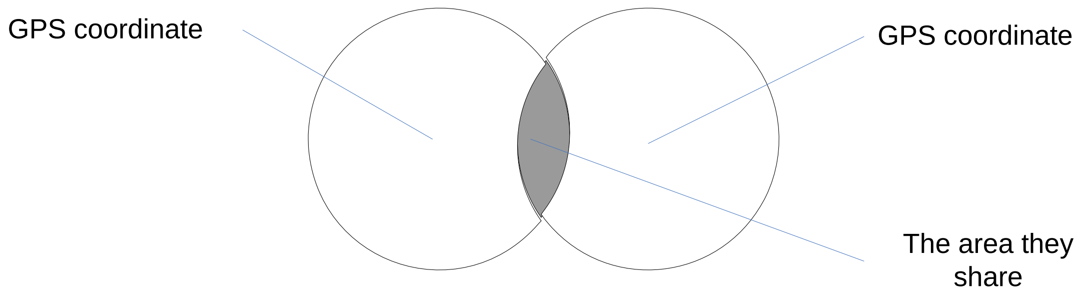


Figure 37: Images sharing a small area

Figure 37 illustrates images sharing a small area. Metadata adopted from another image that share such a small area, might for the image be very inaccurate.

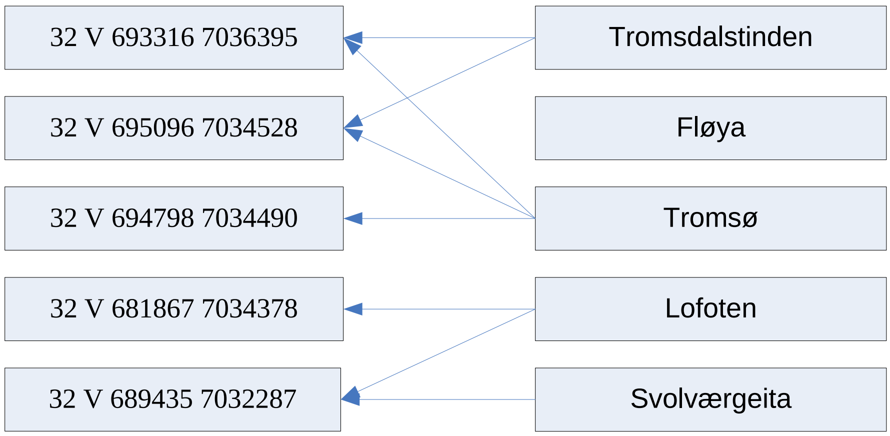


Figure 38: Shows an example of mapping of location notation.

Figure 38 shows that a GPS-coordinate[4] mapping to Tromsdalstinden should also map to images annotated with "Tromsø" but not to images annotated with "Fløya". However a radius set for Tromsdalstinden could easily include Fløya while these are close in location. The same goes with GPS-coordinates mapping to Svolværgeita. These should also map to images annotated with "Lofoten".

The GPS-receiver is quite accurate. Let us say that GPS 32 V 699435 7032287 maps to the top of Tromsdalstinden. As with Tromsdalstinden, many interesting locations are not confined to one spot, but span over an area with undefined borders. When users request "Tromsdalstinden", what does the user actually want? Does he want images displaying the view from this mountain or does he want images displaying the mountain in itself? In this area there might be other attractions of totally different

[4] The GPS-coordinates are fictive, but inspired by http://www.bymarka.net/gps.php

nature with the same undefined borders. How do we decide what attraction the user is requesting? It must be decided for each attraction how big its radius should be. If the radius is set to be small the smaller chance for a hit in the database, but the higher chance that the information the user retrieves is relevant. The longer radius the higher probability that it will result in a hit in the database, but the chance that it will result in irrelevant hits is higher.

The use of GPS-coordinates does say where the image was snapped but not necessarily what the image is displaying. A solution to this can be to compare the visual features of the image with the other images in the database. When the image is added to the database, content-based image recognizing can detect what the image is displaying and automatically annotate the image with a metadata of images that are visually similar to it. Correct mapping of metadata assumes that some images in the database have images with correctly annotated text.

6.6.2.2 Inheriting metadata from parent images

We want the images in the database to have comprehensive metadata that is to be added to the keywords column in the database of CAIR. To achieve this we want the images to inherit the metadata from other pictures that also apply to these images. This requires that there are images in the database that is possible to inherit metadata from. These images are called ground images and we know from beforehand that these images have correct metadata and GPS-coordinates. It is very important that these images are correctly annotated so the inheriting images do not get metadata that does not apply to them. These ground images and GPS-coordinates could exist for large areas such as for instance the largest cities in Norway. The image is annotated with a location and GPS-coordinates of the center of this area. These GPS-coordinates are given a radius on a given number of kilometers depending on the size of the area. For Tromsø this radius could for instance be 5 km radius, while Oslo, that is a much bigger city, could have a radius of 10 kilometer.

The idea of ground images is also a way to avoid having attractive images in the database that are poorly annotated and therefore harder to retrieve. We must however be cautious to automatically adopt too much of other image’s metadata. The metadata of images can then be too general in proportion to the image’s visual content. If images share the same position does not necessarily imply that the visual content of the images are the same.

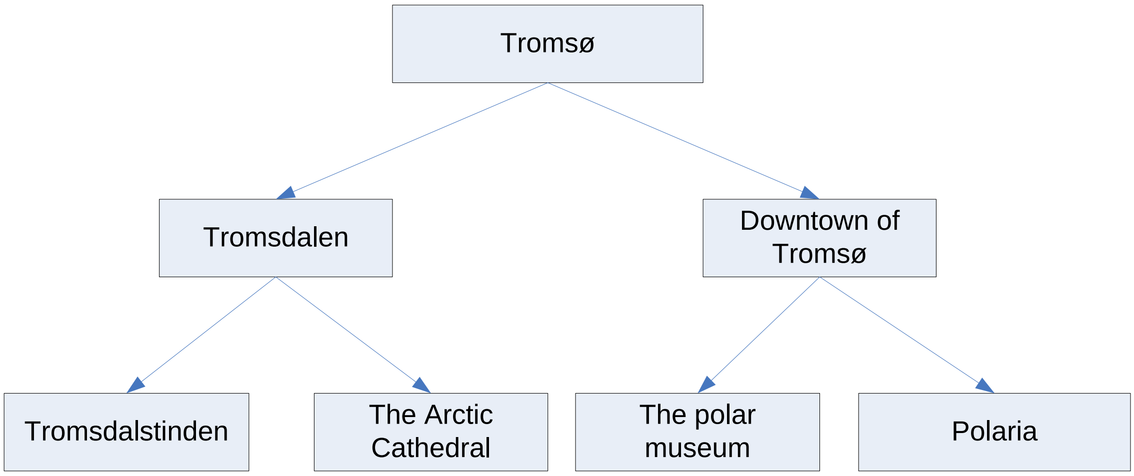


Figure 39: Illustrates a hierarchy of annotations.

The figure illustrates that the images can inherit metadata from its parent images in the hierarchy. Tromsdalstinden inherits the metadata of images of Tromsdalen and Tromsø, but not downtown of Tromsø, The Polar Museum or Polaria.

In CAIR we hope that users will annotate the images. If images are not annotated we want to utilize these ways to automatically annotate images.

6.7 Different use of location as context

There are two different ways to use location as context: specified location and current location.

6.7.1 Specified location

This approach uses a specified location to retrieve information about any attraction regardless of the users current location. It requires the annotated image to have location information associated. The user can send the location of an attraction to the short number and get information of the attraction based on this context. If the user wants information about The Viking Museum at Borg the user could for instance type *Borg* in an SMS and send to 2034. The user would then retrieve information about The Viking Museum. This would work because it does not exist so many tourist attractions at Borg so the probability that it is The Viking Museum the user wants is high. *Svolvær*, however, is a far bigger area and has several attractions so it is not obvious what information the user actually is requesting. The search would probably result in general information about Svolvær. If this is what the user wants it is a successful hit, but if the user wants specific information about an attraction in Svolvær this can result in a dissatisfied user.

This approach has much of the same advantages and disadvantages as the text-based image retrieval discussed in section 2.3.1. The users have to type in textual information that can easily contain spelling-errors and therefore not return results that would otherwise have been given the user.

Not all locations have a specific name. Lets say that the user wants to take a guide trip into the mountains of Lofoten and he wants to use CAIR to find out whether there exist such guided tours. What location shall he write? "Mountain of Lofoten" or "guided tours" or a combination of these phrases? The keyword must be associated with an image if it is going to return any results.

6.7.2 Current location

This approach differs from the former one in the way that it is the user's current location and not a specified location that is used to retrieve information. This approach requires support to determine the user's current position, such as the Nokia N95. It also assumes that the GPS location can be automatically added as a parameter to the query in the SMS. The user adds information about his current position to an SMS, add the service name and time-option and send the SMS as usual. Based on the GPS-coordinates the program searches the database to find the attraction that maps the coordinates and the user gets the belonging information. A restriction of this approach is that the user will only receive information about the location he is visiting and no other locations.

When the user is actually at the location of an attraction he can snap pictures on his own. This approach can be used to retrieve images from the current location displaying another point of time. Tourists reaching The North Cape on a foggy day do not get to see the view. They may want to bring home some pictures so they use CAIR to retrieve images from The North Cape hopefully taken on a sunny day.

6.7.3 A combination of general location and current location

As we have seen there are advantages and disadvantages to both approaches. The best approach depends on the user's situation. In some situations the use of his current position could be best. The user could then add the current location as parameter to the query SMS. In other situations the ability to specify a location could be best. If he wants information about a remote location he can send the name of the location as an SMS to the service. As we can imagine this will result in a more flexible solution and can handle more cases than the approach with current location. Specifying locations could return images from locations where the user never intends to visit. Using current location, the user has to actually be at the location to retrieve desired images.

6.8 Alternative ways to formulate the query

There are alternative solutions to this service. The approaches for CAIR that we want to discuss are based on a query of the following combinations of context:

- Textual information.
- Image + GPS-coordinates.
- GPS-coordinates + textual information.
- Image + GPS-coordinates + textual information.

6.8.1 Textual information

The performed query is based on textual information only. The information retrieval will be based on the image's metadata. As discussed in the background chapter this approach is dependent that the users have a common understanding of what information describes the image. It also requires that the images are annotated with at least the name of the location. However, as we have seen in section 2.3.1, this is not a straightforward task. A location can have different names. If the name in the query and the name of the attraction in the database do not match, this will not return results.

6.8.2 Image + GPS-coordinates

As we have seen typing in a textual description of the location is not unambiguous. Tromsdalen is a large area, what exactly does he want retrieved? Does he want general images or of a specific entity? To detect this, the query can be a combination of content-based image retrieval and GPS-coordinates. Using GPS-coordinates gives the user's current location. The query is narrowed down with the use of location and the content of the image can determine the user's intentions. This makes the image search much faster way than using the visual content of images alone as query and it can be much more precise than using location alone. Using GPS-coordinates together with the content-based image retrieval in M2S could prove to be a powerful combination. A system that uses this approach is SnapToTell [38] that we discussed in section 3.5.

6.8.3 GPS-coordinates + textual information

The combination of GPS-coordinates and textual information is another approach for information retrieval. The query is formulated as the user's current GPS-coordinates and the textual description can be the name of the location or attraction. The query can be narrowed down using the GPS-coordinates and confirmed by using the textual description. The location tells where the user is, but not what he is looking at and is interested in. The textual description is a more precise way to tell the user's intentions. If the GPS-coordinates maps to Tromsdalen and the textual information says "Fløya" the user can be requesting images of Fløya taken from Tromsdalen.

6.8.4 Image + GPS-coordinates + textual information

This has the potential to be a powerful combination. Since content-based image retrieval is insufficient when searching within large datasets, it could be very efficient to narrow down the query with use location information. Like the former combination, the location can be formulated with the user's current position using GPS-coordinates. The textual description can narrow the even more and the image search can be used to detect what the user is looking at and want information about. If the textual description, the location and there is an image match in the database, this is probably what the user had in mind when formulating the query.

6.9 Rating of results

The database can contain many images from the same location. Lets say that the user formulates a query with the location “Svolværgeita” and the database contains 500 images from this location. It is irrelevant to send all 500 images to the user. To select which images shall be sent, there must be a rating of the images.

Providing users with an MMS with a set of almost identical images is not so interesting. When a query is performed the images chosen should of course satisfy the criteria but otherwise be different. Some images could be of Svolværgeita in daylight, others of Svolværgeita at night.

6.10 How to upload images to the database

We imagine that users can upload images to the database via a web-browser. Figure 40 is an example of what the user interface could look like. The user can browse his desktop for images and upload them to the database. The images uploaded will be displayed to the user as a thumbnail. Next to each image there is a field to type in keywords describing the image and its relevant surroundings. When uploading the image, the image’s time and location, if it exists, are extracted from the image and displayed. If this information is not available through the image’s metadata this must be added to the keywords in the text field.

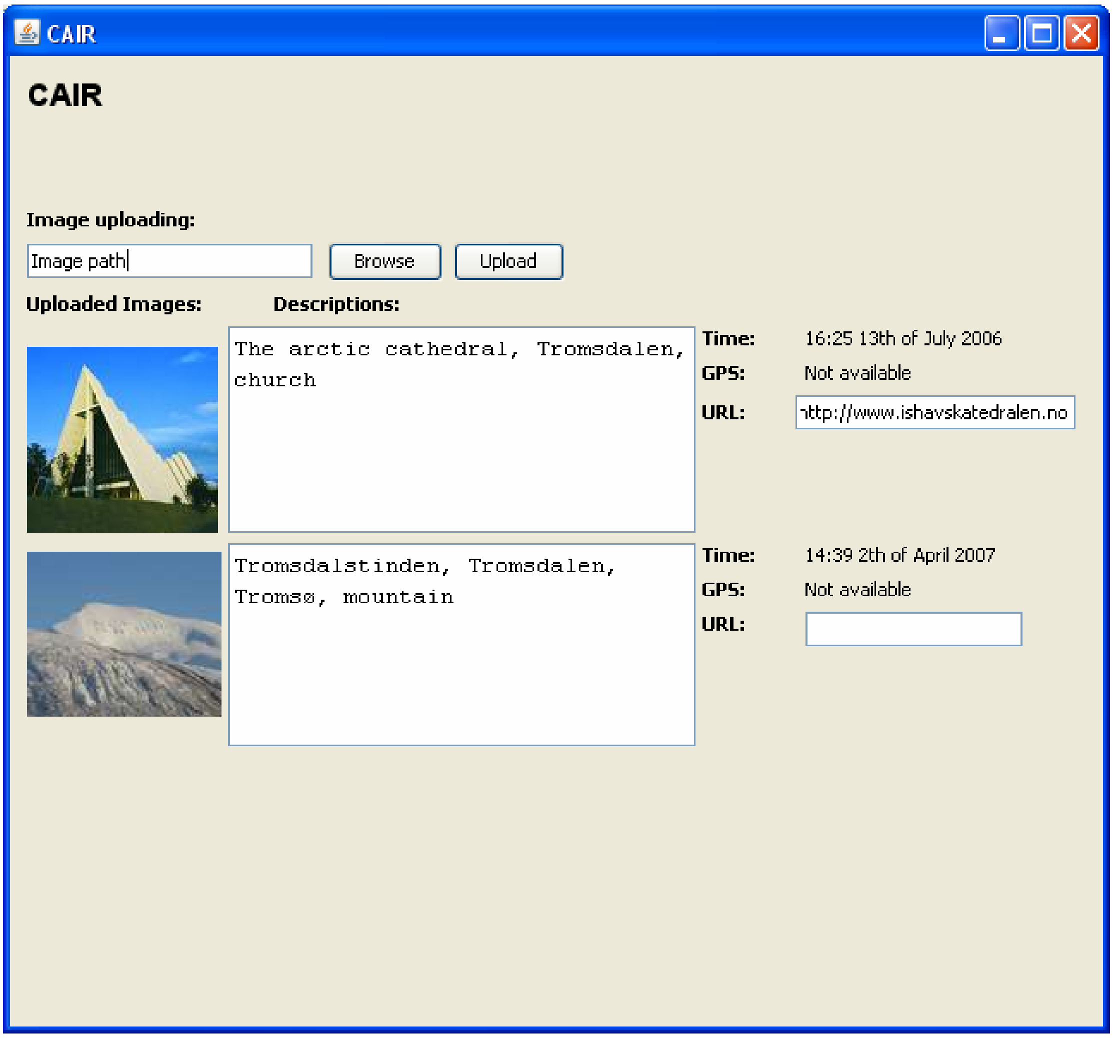


Figure 40: Illustrates how to upload images and URLs.

This is only an illustration of what the graphical user interface (GUI) could look like. The figure does not have any functionality.

6.11 Advantages and disadvantages of CAIR

CAIR has some advantages. CAIR would be fun and easy to use and these are important aspects of CAIR. It is also CAIR also avoids browsing of images. The result is sent to the user and all images in the MMS are displayed. Ideally all the images in the MMS should be relevant. Having well-annotated images and good mapping functionality for images, CAIR can deliver attractive images on the user's mobile phone.

One of limitations of M2S is that it is dependent of the tourist brochure. CAIR would not be dependent of any tourist brochure or the user being at the actual location when using the service. If a user wants information about Mount Everest this can be retrieved by users typing "CAIR", "Mount Everest" and lets say "summer".

CAIR also has some disadvantages. Users compose textual messages on daily basis. However, to remember the syntax of the query is not necessarily so easy and the short number must also be remembered. The syntax for the query is relatively short so this is manageable for users to remember. As with the short number, we have a solution in section 5.9.2 of how to remember it. Location information tells where the image was taken but it does not say anything about what is displayed on the image. The images retrieved might be of the wrong motive even if the location matches. As discussed in section 6.8.2 this can be solved using a content-based image retrieval function.

The service also depends on users uploads attractive images and that the images having information associated. This can become a strength of the service just as much as it can become a disadvantage. If users upload attractive images that are annotated with correct metadata, the probability that the image is according to the users query is higher than for instance if the images are poorly annotated and the images in themselves are quite uninteresting.

6.12 Future work

This section will describe a detailed plan for implementing CAIR. Subsection 6.12.2 will give examples of how CAIR can be further developed to provide a richer service.

6.12.1 Plan for implementation of CAIR

First and foremost CAIR has to be implemented, tested and evaluated based on the testing.

To implement CAIR it is possible to use much of the same development tools, both hardware and software, as we used when implementing M2S. This development tools was described in section 5.7. In CAIR the users can upload images with associated information. So in contrast to M2S CAIR require a GUI (Graphical User Interface). We have used and will use NetBeans verion 5.5 to implement the graphics of CAIR. This uploading GUI can be seen in Figure 40. As we have discussed Java has classes that handle images and we plan to use the ImageInfo class to extract metadata from the images. The extracting functionality has been implemented as we saw on page 94.

In case images are not annotated, the mapping of image annotation described in section 6.6 can handle this. Inheriting images can also be implemented but used in cases where images do not map.

Functionality for transmitting SMS and MMS must be supported by Telenor and PATS. An agreement must be established so permission to use the PATS message servers can be used. The PATS-lab can be used to transmit SMS, MMS and WAP. A service name, for instance “cair” must also be established so the SMS’ received at the message server are forwarded to the right service.

6.12.2 Extensions of CAIR

CAIR can be extended to include other kinds of contexts such as for instance weather conditions. The information can be added to the image the moment it was captured and makes users able to search for images based on the weather conditions. It will then be able for users to search for images based on the image’s weather conditions displayed on the image.

As discussed in section 6.8.2 content-based image retrieval functionality could be added to the service. The service could detect based on the visual features of the images what attraction at the location the user is requesting information about.

6.13 Summary

In this chapter we have described CAIR. CAIR is a part of the CAIM-project and is a context-based image retrieval service that offer users a series of images from a given location and time.

CAIR shares much of the motivation as M2S. As users bring their mobile phones practically everywhere and many have a desire for information, our service can be used to retrieve information. The information provided in our service is mainly images, but WAP URLs are also offered.

CAIR has been discussed in detail and we have given an example of what situations the service can be used. We also have illustrated the usefulness with a user scenario.

We have discussed several ways of using location to retrieve information, current location and general location and we have seen that context combined with other concepts such as content-based image retrieval can be powerful combination. We have also discussed location as context and problems related to this. We have seen that images annotated with GPS coordinate has to have a mapping to textual description of the location. We have also seen that some of the same issues apply to time as context and that there must be a mapping of points of time.

Restrictions of the service and restrictions of the user, advantages and disadvantages were discussed.

Future work could be adding other types of context such as for instance weather information to the images. It will then be able for users to search for images based on the image’s weather conditions at the moment it was captured.

We will give an evaluation of CAIR in chapter 7.

7 Evaluation and conclusion of M2S and CAIR

In this chapter we will discuss the work presented in this thesis. We will first give a brief summary of M2S and CAIR. We will then discuss the services and have a closer look at parts that could have been done in a different and maybe better way.

7.1 Summary

The introduction of digital cameras and especially the incorporation of these into mobile phones, makes people able to snap photos almost everywhere at any time while their mobile phone is brought almost everywhere. The rapid growth in hardware allows users to store large image collections at low costs.

Image retrieval software has not had the same development as the hardware. Current image retrieval techniques have shortcomings and lack functionality to provide users with the images they want. Current image retrieval techniques make use of low-level features such as the visual contents of the images. Humans use high-level features when formulating queries. What the user wants to retrieve, he is often not able to formulate as a query that computers understand. As we discussed in section 2.5 this is referred to as the Semantic Gap and is the most important topic of image retrieval. Several projects such as the CAIM-project aims to develop techniques of image retrieval to narrow this gap. This thesis is made within the CAIM-project.

The goal of this thesis was to design two encyclopaedia services meant for users with mobile phones. One intermediate aim was to design and developing a service that could provide users with information based on the visual contents of a query-image. Another intermediate aim is to design a context based image retrieval service that offers users with images from an image collection. The goal and sub-goals have been accomplished.

In this thesis we have described M2S and CAIR that enables users to retrieve information such as images or textual information, WAP-links and videos using SMS and MMS. M2S and CAIR have in common that they try to cover a need for information in mobile settings. However, they use different techniques to achieve this. One of the services, M2S is meant for tourists retrieving information about attractions through an image query issued using MMS. The query MMS contains an image of a photographed attraction. This image is used as basis to retrieve information about the requested attraction. M2S uses a content-based image retrieval technique to provide users with information such as videos, WAP-links, images or textual information or a combination of these. Due to limitations in content-based image retrieval we have chosen that the image have to be of an image from the Lofoten Info-Guide provided by Destination Lofoten. The design and implementation of the service has been described and discussed and can be viewed from section 5.4 through section 5.8.3.

The other service, CAIR is meant for users who want to retrieve images from an image collection using their mobile phone. The location and time is extracted from the images metadata when the image is added to the database and additional location information in form of keywords can be added manually. A query for images is composed with context information, such as location and time, formulated as text and issued using SMS. The query is matched against the metadata of the images. If matches are found the images of the matched metadata are sent as MMS to the user initiating the dialog. CAIR is designed but not yet implemented. This thesis has described a design overview of CAIR that can be viewed in section 6.5.

7.2 Evaluation

This thesis demonstrates that it is possible to retrieve information about attractions and locations using different technologies for image retrieval by using MMS and SMS. The implementation of M2S shows that it is possible to snap a picture of an attraction from the tourist brochure and use the visual contents of the photograph to retrieve information associated with this photograph using MMS.

The design of CAIR describes how the use of location can be used together with a point of time to retrieve information about the specified location. CAIR is a simple design of a service to find out if it is possible to retrieve specific images from a remote photo collection using SMS. The design also shows that mapping to a certain extent can solve problems related to overlapping locations and points of time.

When designing and developing M2S we experienced content-based image retrieval. Developing a content-based image retrieval function for M2S was a too complex task so we wanted to use an already existing program for this. The program should be free of charge, preferably open-source and be able to match a query image to the images in the database. The programs that we investigated can be viewed in section 5.5.1. Lire [97] was the program that best matched our requirements. However, Lire was not an adequate image retrieval technique, but the best of the ones that we tested. The experiences and the shortcomings of the different content-based image retrieval techniques and details of Lire can be viewed in section 5.5.

Content-based image retrieval for this type of image collection it not simple. However, content-based image retrieval is very suitable for controlled and limited areas like face recognition fingerprint matching, trademark retrieval and fabric selection, but having pictures that can contain much visual information is hard for the retrieval function to handle [1].

Because of these shortcomings to the content-based image retrieval techniques, the users of M2S cannot snap a picture of any attraction, but had to use the attractions reproduced in the Lofoten-Info Guide. As discussed in section 5.2.2 the use of this brochure has advantages and disadvantages. The use of the brochure makes people able to be anywhere and retrieve information about attractions printed in the brochure. As we have seen in section 5.3.1 tourists collect information about the destination they want to visit. If they receive the tourist brochure via for instance mail, they can retrieve additional information in form of WAP-links and videos using M2S. Being dependent of the brochure is also a disadvantage. If users could snap pictures of attractions independent of the brochure and retrieve information about the photographed entity we believe this could easily become a popular service. The tourist brochure contains information in form of textual descriptions and images. So convincing the users to spend money on additional information about attractions they already have information about might be difficult.

Another problem that we encountered during the development of M2S was to make the communication using MMS to function. This was discussed in section 5.11.3. Sending WAP URLs functioned well. This was because the testing environment for sending WAP URL was in place.

This thesis has also experienced context-based image retrieval and the use of context in mobile settings. While content-based image retrieval solves some problems the use of context brings other types of problems than for instance content-based image retrieval. There are for instance some issues related to CAIR. CAIR does not avoid annotation as can be viewed in section 6.6. In CAIR the annotation can be either done partly manually or automatic based annotation got from images that have the same location. Even if GPS with automatically annotation of location of the images were common, this would not solve the need for manual annotation. The human way of formulating queries is a textual location of the image and there must be a mapping between the technical and the human way of expressing descriptions. As discussed in section 6.6.2 a location can be areas of varying sizes that overlap. The same principal applies to points of time and this makes queries using such context ambiguous. Using location alone is also problematic because there might be several attractions in the same area and the location alone does not detect which one the user wants information about.

It can be questioned if it is tactical to build a database with images and information while there already exist amounts of images and associated information. Search engines like Google [31] and Sesam [123] has large databases of images and associated information. Because the information surrounding the image is automatically associated with the image, the information is not necessarily describing the image. A search with Sesam with query “Svolværgeita” and “summer” gave zero hits. A search with Google gave many hits but few of these actually displayed Svolværgeita in summer time. The information has to be annotated correctly to provide useful information in a service combining techniques used in M2S and CAIR.

When I finished the summer work at Telenor R&I the functionality for transmitting SMS and MMS was not working due to a lack of support for this. Telenor R&I have now completed the implementation and M2S has been briefly tested. The service functioned as whole. Brief testing of the application on a fixed set of images, gave a non-official testing result. The hit ratio for Lire was of approximately 20%. As discussed in section 5.11, performance evaluation is often a neglected topic of content-based image retrieval research. The image sets are often small and fixed [1] [119]. There are few measurements of content-based image retrieval services on complex images such as the images used in M2S. Measurements done on simple images in a small dataset are irrelevant to compare Lire against. Anyway, a hit ratio of 20% is not a sufficient hit ratio to base a commercial system. Sending MMS is relatively expensive and customers expect a hit ratio between 90 and 95% if not even higher.

Most users are quality-conscious. They want quality services that provide rich and valuable information. For M2S to be a full worthy service that can be offered to customers, there has to be some improvements. The main improvement concerning M2S is related to a better content-based image retrieval function. As discussed in 5.12 content-based image retrieval has a huge potential. The commercial market for services based on content-based techniques exists and there is a lot of research going on in this field. This is mostly because most current text-based image retrieval systems does not cover the need for information retrieval and has several shortcomings as we saw in section 2.3.1 [1]. Content-based image retrieval techniques are not yet effective enough, but their capabilities seem to improve. When sufficient techniques are available we believe will see a growth in services based on content-based image retrieval. We hope that within few years we will see the incorporation of adequate content-based image retrieval techniques in commercial services.

While investigating new and better content-based image retrieval techniques it is possible to combine concepts like context-based and content-based techniques. We believe that combining concepts like this we can provide more powerful services.

7.3 Future work

CAIR must be implemented and then tested before we can tell if the service will actually function and prove useful.

While sufficient content-based image retrieval functionality do not exist yet, we have to utilize other concepts like for instance context to provide better service. What we have experienced is that location alone does not solve the problems related to image retrieval. Detecting the user's location is relatively unproblematic, but the location alone cannot detect the user's intention being at the location. However, combining location information and content-based image retrieval could provide a solution that enables users to take a picture of a free- standing attraction and retrieve information about it. The user's location could be used to narrow the area for the search and a content-based image retrieval function could unveil the user's intention when photographing. Such a solution could increase the chances of retrieving correct results. This is an approach similar to Lim et al.'s [38] SnapToTell. However, we include more context than only location such as for instance time which can provide users with information from specific points of time. Including time and location can be used to fetch other data such as for instance weather information and add it to the images metadata.

Using a combination the user's location could be detected using a GPS. From a given location, X, it is limited what the user actually can photograph. If the GPS-coordinates are stored together with the images, it is possible to search for images based on a combination of this information. From a location, there could be several attractions and images that potentially could be snapped from that location are fed to a content-based image retrieval function. If the image does not match it is not the attraction the user is photographing and the search continues. This assumes that the database is richly equipped with images. This solution is independent of any tourist brochure, but is assumes that the user is at the location of the requested attraction. Snapping a photo of an attraction and adding GPS-coordinates and time to the query could provide users with valuable information. The location information and the image decide what attraction the user is requesting information about.

A combination of different technologies and a richly equipped database that is often updated can provide users with a tailor made service. The users could receive information such as timetables, the time it takes to travel between attractions and prices. The user will be able to retrieve updated dynamic information like the menu of the day, closing hours, availability, weather information, how long the queues are to enter facilities etc. Some attraction/facilities are dependent on weather conditions. Tourist would be able to retrieve updated information about whether these facilities are open or not. The system can keep track of his searches and store this information in a personal profile. Based on the information about what kind of interests the user has and his location the service could give him information about similar activities the user could be interested in. If a tourist is interested in information about paragliding 5 miles away and this activity is closed due to bad weather, the service can, based on the user's search history presume that the user is interested in sports and give him information about sailing facilities close to his current location together with information about prices and time of departure.

7.4 Conclusion

We believe the idea of CAIR and M2S is good. As they are designed today, they have shortcomings that make them unfit for commercial use. Retrieving desired images from an image collection using SMS is a good idea. Using location alone cannot support a satisfying service. When GPS is standard in mobile phones and content-based image retrieval techniques gets better and makes users able to retrieve information about any free-standing attraction, we believe M2S can be a popular service.

Bibliography

1. Eakins John, Margaret Graham, *Content-based Image Retrieval*, in *A report to the JISC Technology Applications Programme*. 1999, Institute for Image Data Research, University of Northumbria: Newcastle.
2. Flickner M., H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker, "Query by Image and Video Content: The QBIC System.*" Computer*, 1995. **28**(9): p. 23-32.
3. Zhao Rong, W. I Grosky, "Bridging the semantic gap in image retrieval." *Distributed Multimedia Databases: Techniques and Applications, TK. Shih(ED.),* Hershey, Pennsylvania,
4. Gudivada Venkat N., Vijay V. Raghavan "Content-based Image Retrieval Systems.*" Computer*, 1995. **28**(9): p. 18-22.
5. CAIM. [cited; Available from: http://caim.uib.no/objective.shtml.
6. Dey Anind K., "Understanding and using context.*" Personal and Ubiquitous Computing Volume, 4 DOI.*, 2001. **5**(1): p. 4-7.
7. Dey A.K. Abowd G.D., "Towards a better understanding of context and context-awareness*". Conference on The What, Who, Where, When and How of Context-awareness, Conference on Human Factors in Computer Systems (CHI2000)*, 2000.
8. Brown P.J, G.J.F. Jones, "Exploiting contextual change in context-aware retrieval." *Proceedings of the 2002 ACM symposium on Applied computing,* Madrid, Spain,
9. Brown P.J, G.J.F. Jones "Context-aware Retrieval: Exploring a New Environment for Information Retrieval and Information Filtering.*" Personal and Ubiquitous Computing*, 2001. **5**(4): p. 253-263.
10. Jones G.J.F., P.J. Brown, *Context-Aware Retrieval for Ubiquitous Computing Environments. Lecture Notes in Computer Science*. 2004, Berlin / Heidelberg: Springer.
11. Allan James (editor), Jay Aslam, Nicholas Belkin, Chris Buckley, Jamie Callan, Bruce Croft, Sue Dumais, Norbert Fuhr, Donna Harman, David J. Harper, Djoerd Hiemstra, Thomas Hofmann, Eduard Hovy, Wessel Kraaij, John Lafferty, Victor Lavrenko, David Lewis, Liz Liddy, R. Manmatha, Andrew McCallum, Jay Ponte, John Prager, Dragomir Radev, Philip Resnik, Stephen Robertson, Roni Rosenfeld, Salim Roukos, Mark Sanderson, Rich Schwartz, Amit Singhal, Alan Smeaton, Howard Turtle, Ellen Voorhees, Ralph Weischedel, Jinxi Xu, ChengXiang Zhai, "Challenges in Information Retrieval and Language Modeling.*" SIGIR Forum*, 2003. **37**(1): p. 31-47.
12. Baeza-Yates Ricardo, Berthier Ribeiro-Neto, *Modern Information Retrieval*. 1999, New York: ACM Press.
13. Pitoura Evaggelia, Bharat Bhargava, "Building information systems for mobile environments." *Proceedings of the third international conference on Information and knowledge management*, Gaithersburg, Maryland, United States,
14. MOVE. [cited; Available from: http://www.moveweb.no/.
15. Schürmann Anders, Anne Staurland Aarbakke, Espen Egeland, Bente Evjemo, Sigmund Akselsen, "Let a picture initiate the dialog! A mobile multimedia service for tourists.*" Telenor R&I Research Note*, 2006. **33/2006**.

16. Cooper Matthew, Jonathan Foote, Andreas Girgensohn, Lynn Wilcox, "Temporal Event Clustering for Digital Photo Collections." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 2005. **1**(3): p. 269-288.
17. Rodden Kerry, Kenneth R. Wood, "How do people manage their digital photographs?" *Proceedings of the SIGCHI conference on Human factors in computing systems*, Ft. Lauderdale, Florida, USA,
18. Loui Alexander C., Mark D. Wood "A software system for automatic albuming of consumer pictures." *Proceedings of the seventh ACM international conference on Multimedia (Part 2)*, Orlando, Florida, United States,
19. Keller I. Stappers Pieter Jan, "Supporting informal collections of digital images: organizing, browsing and sharing". *ACM International Conference Proceeding Series*, 2004. **65**.
20. Kustanowitz Jack., Ben Shneiderman, *Motivating annotation for Personal Digital Photo Libraries: Lowering Barriers While Raising Incentives*. 2005, University: Maryland.
21. Gupta Amarnath, Ramesh Jain, "Visual information retrieval." *Communications of the ACM*, 1997. **40**(5): p. 70-79.
22. Flank Sharon, "Multimedia IR in Context." *Proceedings of the 2001 ACM workshops on Multimedia: multimedia information retrieval MULTIMEDIA '01*, Ottawa, Ontario, Canada,
23. Taylor Chris. *An introduction to metadata*. 1999 [cited; Available from: http://www.itb.hu/fejlesztesek/meta/hgls/core/Background/An_Introduction_to_Metadata.htm.
24. International Organisation for standardisation ISO/IEC JTC1/SC29/WG11 Coding of moving pictures and audio. *MPEG-7*. [cited; Available from: http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm.
25. Dublin Core Metadata Initiative. [cited; Available from: http://dublincore.org/.
26. EXIF-standard. [cited; Available from: http://www.exif.org/.
27. Zhang Chen, Joyce Y. Chai, Rong Jin, "User term feedback in interactive text-based image retrieval." *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, Salvador, Brazil,
28. Keller Ianus. Stappers Pieter Jan, Vroegindeweij Sander, "Supporting informal collections of digital images: organizing, browsing and sharing". *ACM International Conference Proceeding Series*, 2004. **65**.
29. Wilhelm Anita, Yuri Takhteyev, Risto Sarvas, Nancy Van House, Marc Davis, "Photo Annotation on a Camera Phone." *ACM CHI 2004*, Vienna, Austria,
30. Wenyin L., S. Dumais, Y. Sun, H. Zhang, M. Czerwinsky, B. Field, "Semi-Automatic Image Annotation." *In Proc. of Interact: Conference on HCI*,
31. Google. [cited; Available from: http://www.google.com.
32. Yahoo. [cited; Available from: http://uk.yahoo.com/.
33. Altavista. [cited; Available from: http://www.altavista.com.
34. Schank Roger C., Janet L. Kolodner, Gerald DeJong, "Conceptual information retrieval." *Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*, Cambridge, England,
35. Octagon. [cited; Available from: http://users.utu.fi/jukvii/octagon.

36. McDonald Sharon, Ting-Sheng Lai, John Tait, "Evaluating a content based image retrieval system." *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval,* New Orleans, Louisiana, United States,
37. McDonald Sharon, John Tait, "Search strategies in content-based image retrieval." *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval.,* Toronto, Canada,
38. Lim Joo-Hwee, Jean-Pierre Chevallet , Sihem Nouarah Merah, "SnapToTell: Ubiquitous Information Access from Camera." *Mobile Human Computer Interaction with Mobile Devices and Services,* Glasgow, Scotland,
39. imgSEEk. [cited; Available from: http://www.imgseek.net/.
40. Smith John R., Shih-Fu Chang, "VisualSEEk: a fully automated content-based image query system." *Proceedings of the fourth ACM international conference on Multimedia*, Boston, Massachusetts, United States,
41. GIFT The. [cited; Available from: http://www.gnu.org/software/gift/gift.html.
42. Weber Roger, Michael Mlivoncic, "Efficient region-based image retrieval." *Proceedings of the twelfth international conference on Information and knowledge management,* New Orleans, LA, USA,
43. Blobworld. [cited; Available from: http://elib.cs.berkeley.edu/blobworld/.
44. Natsev Apostol, Rajeev Rastogi, Kyuseok Shim, "WALRUS: a similarity retrieval algorithm for image databases." *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, Philadelphia, Pennsylvania, United States,
45. Wang James Z., "SIMPLIcity: a region-based retrieval system for picture libraries and biomedical image databases." *Proceedings of the eighth ACM international conference on Multimedia* Marina del Rey, California, United States,
46. Rui Yong, Thomas S. Huang, Michael Ortega, Sharad Mehrotra, "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval." *IEEE Trans. on Circuits and Systems for Video Technology*, 1998. **8**(5): p. 644--655.
47. CiteSeer. [cited; Available from: http://citeseer.ist.psu.edu.
48. Lycos. [cited; Available from: http://search.lycos.com/.
49. Cox Ingemar J., Matt L. Miller, Thomas P. Minka, Thomas V. Papathomas, Peter N. Yianilos "The Bayesian Image Retrieval System, PicHunter; Theory, Implementation and Psychophysical Experiments." *IEEE Tran. On Image Processing,*, 2000. **9**(1): p. 20-37.
50. Ishikawa Yoshiharu, Ravishankar Subramanya, Christos Faloutsos. "MindReader: Querying databases through multiple examples." in *Proc. 24th Int. Conf. Very Large Data Bases, VLDB*. 1998.
51. Tøndersen Arne-Henrik "Image retrieval based on location and time." Master, *Computer Science*, Tromsø, 2006
52. JeeWiz! [cited; Available from: www.jeewiz.com/ product_architect.html.
53. Dey Anind K., "Understanding and using context." *Personal and Ubiquitous Computing Volume, 4 DOI.*, 2001. **5**(1): p. 4-7.
54. Hofer T., W. Schwinger, M. Pichler, G. Leonharstberger, J. Altmann. "Context-awareness on mobile devices –the hydrogen approach." in *In*

Proceedings of the 36th Annual Hawaii International Conference on System Sciences. 2002.
55. Schmidt Albrecht, Michael Beigl, Hans-W. Gellersen, "There is more to Context than Location." *Computers and Graphics*, 1999. **23**(6): p. 893--901.
56. Schilit B. Adams N., Want R. "Context-aware computing applications". in *Proceedings of the Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, December 1994*. 1994.
57. Kaasinen Eija, "User needs for location-aware mobile services." *Personal and Ubiquitous Computing*, 2003. **7**(1): p. 70-79.
58. Gurrin Cathal, Gareth J.F. Jones, Hyowon Lee, Neil O'Hare, Alan F. Smeaton, Noel Murphy, "Mobile Access to Personal Digital Photograph Achives." *Proceedings of the 7th international conference on Human computer interaction with mobile devices \& services*, Salzburg, Austria,
59. Nokia. [cited; Available from: http://www.nokia.com/lifeblog.
60. Baldauf M. Dustdar S., Rosenberg F., "A Survey on Context-Aware Systems". *International Journal of Ad Hov and Ubiquitous computing*.
61. Maass Henning, "Location-Aware Mobile Applications based on Directory Services." *Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking*, Budapest, Hungary,
62. Toyama Kentaro, Ron Logan, Asta Roseway, "Geographic location tags on digital images." *Proceedings of the eleventh ACM international conference on Multimedia*, Berkeley, CA, USA,
63. Dey Anind K., Gregory D. Abowd, "CybreMinder: A Context-Aware System for Supporting Reminders." *Proceedings of the 2nd international symposium on Handheld and Ubiquitous Computing*, Bristol, UK,
64. Want R., A. Hopper, V. Falcão, J. Gibbons, "The Active Badge Location System." *ACM Transactions on Information Systems*, 1992. **10**(1): p. 91-102.
65. Ciavarella Carmine, Fabio Paterno, "The design of a handheld, location-aware guide for indoor environments." *Personal and Ubiquitous Computing*, 2004. **8**(2): p. 82-91.
66. Cheverst Keith, Nigel Davies, Keith Mitchell, Adrian Friday. "Experiences of Developing and Deploying a ContextAware Tourist Guide: The GUIDE Project." in *Proc. of MOBICOM'2000*. 2000. Boston, USA: ACM Press.
67. Hsi Sherry. "The Electronic Guidebook: A Study of User Experiences using Mobile Web Content in a Museum Setting." in *Wireless and Mobile Technologies in Education*. 2002: IEEE Xplore.
68. Jones G.J.F, P.J. Brown. "The Role of Context in Information Retrieval." in *SIGIR 2004 IRiX Workshop*. 2004. Sheffield, UK.
69. Robertson S., S. Walker, "Threshold setting in adaptive filtering." *Journal of Documentation*, 2000. **6**(3).
70. Flickr. [cited; Available from: http://www.flickr.com.
71. Snapfish. [cited; Available from: http://www.snapfish.com/.
72. Baldauf Matthias, Schahram Dustdar, Florian Rosenberg, "A Survey on Context-Aware Systems." *International Journal of Ad Hoc and Ubiquitous Computing, forthcoming*, 2004.
73. Abowd Gregory D., Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper, Mike Pinkerton, "Cyberguide: A mobile context-aware tour guide." *Wireless Networks*, 1997. **3**(5): p. 421-433.
74. Pauty J. Couderc P., Banâtre M., "Using Context to Navigate Through a Photo Collection". 2005.

75. Pauty Julien, Paul Couderc, Michel Banâtre, "Using Context to Navigate Through a Photo Collection." *Proceedings of the 7th international conference on Human computer interaction with mobile devices \& services*, Salzburg, Austria,
76. Microsoft. [cited; Available from: http://www.microsoft.com/~larryz and http://research.microsoft.com/~larryz/objectrecognition.htm.
77. Lincoln. [cited; Available from: http://lincoln.msresearch.us/Lincoln/Logon.aspx.
78. Technologyreview. [cited; Available from: http://www.technologyreview.com/Biztech/18368.
79. Lire. [cited; Available from: http://www.semanticmetadata.net/lire/.
80. Mahmoud Qusay H. *J2ME and Location-Based Services.* 2004 [cited; Available from: http://developers.sun.com/techtopics/mobility/apis/articles/location/.
81. Lux Mathias, Michael Granitzer, Werner Klieber, "Caliph & Emir: Semantics in Multimedia Retrieval and Annotation", *19th International CODATA Conference*, Berlin, Germany,
82. Lucene. [cited; Available from: http://lucene.apache.org/.
83. Filmweb.no. [cited; Available from: http://www.filmweb.no/.
84. Telenor. [cited; Available from: http://telenormobil.no/tjenester/mms, http://telenormobil.no/tjenester/sms, http://telenormobil.no/tjenester/wap.
85. Wikipedia. [cited; Available from: http://www.wikipedia.org/.
86. PATS. [cited; Available from: http://www.pats.no/.
87. Schürmann Anders, "Interaktive kart på handholdte terminaler som primært brukergrensesnitt for en kontekstsensitiv digital turistguide." Master, *Institutt for Informatikk, Det matematisk-naturvitenskapelige fakultet*, University of Tromsø, 2004
88. Ritschard Mark R., "Thin clients: the key to our success." *Proceedings of the 34th annual ACM SIGUCCS conference on User services*, Edmonton, Alberta, Canada,
89. Akselsen Sigmund, T. A. Bjørnvold, E. Egeland, B. Evjemo, K.J. Grøttum, A. A. Hansen, S. Høseggen, B. E. Munkvold, P. E. Pedersen, A. Schürmann, T. Steen, H. Stikbakke, A.Viken, P. F Ytterstad. "MOVE- a summary of project activities and results." in *Fornebu, Telenor Research and Innovation (R&I R 17/2006)*. 2006.
90. Monsen Svein-Åge, "A marketplace for mobile users in travel and tourism. Implementation of a context aware prototype using ideas from software agency." Master, *Institutt for Informatikk, Det matematisk-naturvitenskapelige fakultet*, Universitetet i Tromsø, 2005
91. Lopatina Irina, "Context-aware Mobile Gaming" Master, *Institutt for Informatikk, Det matematisk-naturvitenskapelige fakultet*, University of Tromsø, 2005
92. The Lofoten Info-Guide. [cited; Available from: http://www.lofoten.info/.
93. Kherfi M. L. Ziou D., Bernardi A., ""Image Retrieval from the World Wide Web: Issues, Techniques, and Systems."". *ACM Comput. Surv.*, 2004. **36**(1): p. 35-67.
94. Viken A., S. Akselsen, B. Evjemo, A.A. Hansen, "Lofotundersøkelsen 2004." *Telenor R&D Scientific Report FoU R*, 2004. **27/2004**.
95. Flognfeldt Thor jr., *Reiselivsgeografi. Innføring i reiselivsgeografisk analyse.* 1995, Oslo: Universitetsforlaget AS.

96. Leiper Neil, "Tourist attracion systems." *Annals of Tourism Research*, 1990. **17**(3): p. 367-384.
97. Lux Mathias. *Lire-Lucene Image REtrieval.* [cited; Available from: http://www.semanticmetadata.net/lire/.
98. Jacobs Chuck Adam Finkelstein, David Salesin. "Fast Multiresolution Image Querying." in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 1995: ACM Press.
99. Maes Pattie. "Concepts and experiments in computational reflection." in *ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages and applications OOPSLA '87*. 1987. Orlando, Florida, United States: ACM Press.
100. MascoT. [cited; Available from: http://seco.asa.cs.uni-frankfurt.de/Seco/mascot/.
101. Eikon. [cited; Available from: http://eikon.recursion.org/.
102. BolinOS. [cited; Available from: http://www.bolinos.com/com/downloads/.
103. ImageSeeker. [cited; Available from: http://corbis.ltutech.com/.
104. Iqbal Qasim, J. K. Aggarwal, "Cires: A System For Content-Based Retrieval In Digital Image Libraries", *Seventh International Conferece on Control, Automation Robotics and Vision*, Singapore,
105. In-Two. [cited; Available from: http://in-two.com/.
106. MUVIS. [cited; Available from: http://www.iva.cs.tut.fi/COST211/MUVIS/muvis.htm.
107. Riya. [cited; Available from: http://www.riya.com/.
108. SCHEMA. [cited; Available from: http://www.schema-ist.org/SCHEMA/.
109. Viper. [cited; Available from: http://viper.unige.ch/demo/php/demo.php.
110. Virage. [cited; Available from: http://www.virage.com/content/products/.
111. Istorama. [cited; Available from: http://uranus.ee.auth.gr/istorama/Home/home_en.html.
112. PictureFinder. [cited; Available from: http://www.tzi.de/bv/projects/picturefinder.
113. SIMBA. [cited; Available from: http://www.aa-lab.cs.uu.nl/cbirsurvey/cbir-survey/node36.html.
114. COMPASS. [cited; Available from: http://compass.itc.it/.
115. Deniman Dave, Tamara Sumner, Lynne Davis, Sonal Bhushan, Jackson Fox, "Merging Metadata and Content-Based Retrieval." *Journal of Digital Information*, 2003. **4**(3).
116. Sony Ericsson. [cited; Available from: http://www.sonyericsson.com/.
117. Eclipse. [cited; Available from: http://www.eclipse.org/.
118. MySQL. [cited; Available from: http://www.mysql.com/.
119. Dimai Alexander. "Assessment of Effectiveness of Content Based Image Retrieval Systems". in *Proceedings of the Third International Conference on Visual Information and Information Systems*. 1999: Springer-Verlag.
120. SmartMobs. [cited; Available from: http://www.smartmobs.com/archive/2006/08/16/google_takes_a_.html.
121. Wang Avery, "The Shazam Music Recognition Service". *Communications of the ACM*, 2006. **49**(8): p. 44-48.
122. Noakes Drew. [cited; Available from: http://www.drewnoakes.com/code/exif/sampleUsage.html.
123. Sesam. [cited; Available from: http://sesam.no/search/?c=p&q=.