

[UiB](#) | [Dept. of Information Science and Media Studies](#)  
[UiTø](#) | [Dept. of Computer Science](#)  
[NTNU](#) | [Dept. of Computer and Information Science](#)  
[Telenor](#) | [Telenor R&I](#)



CAIM  
CONTEXT-AWARE IMAGE MANAGEMENT

# MMIR2

## Mobile Multimedia Image Retrieval

Mathias Hellevang  
September 2<sup>nd</sup> 2008

CAIM-UiB  
TR #4

*The second edition of the CAIM prototype for mobile image retrieval using the Nokia S60 Platform, Java Mobile Edition and Oracle InterMedia*

## Table of contents

Mobile Multimedia Information Retrieval 2 (MMIR2).....	1
1 Introduction.....	3
1.1 Project objectives .....	3
2 Tools used .....	4
3 Design .....	6
3.1 Overview of design .....	6
3.2 The MMIR2 client.....	7
3.3 The MMIR2 server.....	7
3.3.1 The client sends data to the MMIR2 server.....	8
3.3.2 The MMIR2 server stores the data .....	8
3.3.3 The MMIR2 server searches.....	8
3.3.4 The MMIR2 server fetches images from BergenBy DB.....	9
3.3.5 The MMIR2 server sends the images to the client upon request.....	10
3.4 The database .....	10
3.4.1 Structure.....	11
3.4.2 Functions .....	11
4 User manual .....	13
4.1 The menu.....	13
4.1.1 CBIR Search.....	14
4.1.2 CBIR Store .....	18
4.1.3 Browse for image.....	19
5 Future development.....	19
Acknowledgments.....	20
References.....	20
Appendix.....	21
Appendix A – Program code .....	21
Appendix B – Database code .....	21

# 1 Introduction

This report is written to document the MMIR2 (Mobile Multimedia Information Retrieval 2) prototype developed for the CAIM-project (Context-Aware Image Retrieval) during the summer of 2008 by Mathias Hellevang. MMIR2 is the second mobile prototype developed for the CAIM project, and is largely based on the MMIR prototype and the work done by Roe Fyllingsnes and Christian Hartvedt during the summer of 2007.

## 1.1 Project objectives

The projects objectives for the MMIR2-prototype was to expand the MMIR-prototype to include GPS search combined with CBIR (Content Based Image Retrieval) to get a better and more correct result set. In addition to this the project objectives included cosmetic work on the prototype, enhancing the navigation of the result set and full-screen view of pictures from the result set.

A detailed list of objectives follows:

	Requirement	Status summary
1.	Mobile phone image input to MMIR and return of an image result set in a 4-picture array.	This was implemented in MMIR but was broken by the time the development of MMIR2 started.
2.	Attachment of the GPS data from mobile phone to the image so that they can be sent to MMIR for processing.	Instead of attaching data to EXIF, data is stored in database along with picture and other meta data. This makes it easier to process both on server- and client side.
3.	Storage of the mobile phone input image and GPS location in the BergenBy Database.	Done.
4.	Communicate a distance from phone's GPS-location that can be used to select images from the BergenBy DB	Implemented as standard distance of 100 meters as well as selectable distance from settings menu.
5.	Add the name of the primary object in each image in the result set	The image caption is added instead of object name.

6.	Combine GPS and image search capability for MMIR	Done.
7.	Support selection of any of the four images in the result set to make it possible to view in full screen and / or for viewing meta data.	Full screen view implemented. Meta data view is reserved for future development.
8.	<i>Support selection of any of the four images in the result set, to use as feedback to the MMIR search engine.</i>	<i>Reserved for future development.</i>
9.	<i>Add text input for search against the image annotations</i>	<i>Reserved for future development.</i>

As the list describes, some of the objectives were to make sure that the MMIR prototype from 2007 was in a working condition.

## 2 Tools used

**Eclipse 3.4.0:** Eclipse, a graphical Integrated Development Environment (abbreviated IDE) has been praised for its numerous plugins and broad support for different languages. Eclipse was used in the CAIM project because of the support of plugins for J2ME, Tomcat and easy integration with the S60SDK and its emulator.

**EclipseME:** A plug-in for Eclipse, making the development and deployment of J2ME applications easier. With this plug-in you can choose to run the application directly on the mobile phone or on an emulator, through eclipse without having to deploy.

**S60 SDK:** The software development kit from Nokia supporting the S60 phones.

**Oracle SQL Developer:** A free graphical tool for database development. Oracle SQL is great for getting an overview of the database with its graphical view of tables, types, procedures etc. In addition to showing it all in a very helpful manner, SQL Developer has a lot of features useful while

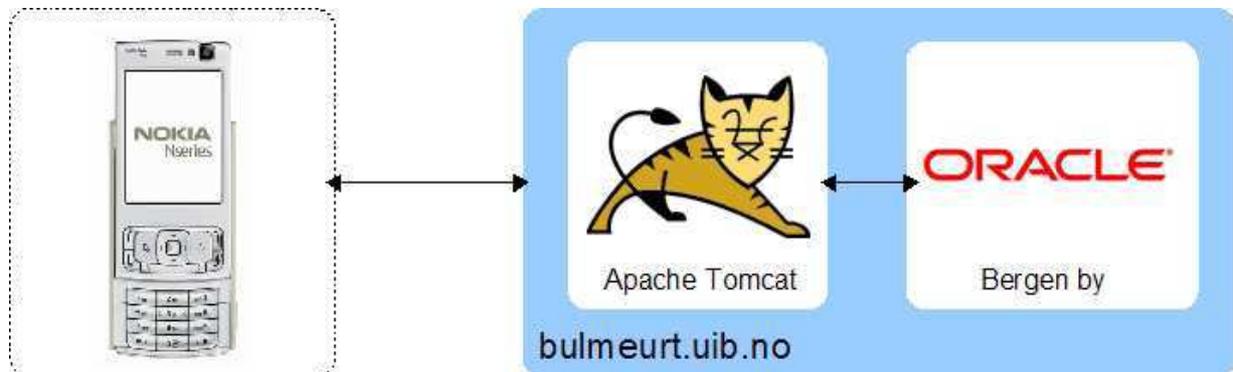
developing and creating databases. SQL Developer runs on Microsoft Windows, Apple OS X and Linux.

## 3 Design

### 3.1 Overview of design

Considering that the project is based on a previous prototype, MMIR, developed in 2007 by Roe Fyllingsnes and Christian Hartvedt, the project uses the same architecture as MMIR. The overall design of the MMIR2 prototype is therefore composed of three parts:

1. The MMIR2 Client, running on a Nokia telephone  
(Developed using S60; Java Platform, Micro Edition)
2. The MMIR2 server running on Bulmeurt.uib.no  
(Developed using the Apache Tomcat servlet API)
3. The MMIR2 BergenBy DB server, also running on Bulmeurt.uib.no  
(Developed using Oracle 10g)



*Figure 1: The MMIR2 prototype. Picture originally developed by Roe Fyllingsnes and Christian Hartvedt for the MMIR prototype.*

### **3.2 The MMIR2 client**

The MMIR2 prototype client has been completely redesigned, trying to make a product more true to the model-view-controller/MVC framework. This was done to make the implementation of new features easier, and the system as a whole more modular. Every GUI-element has been implemented with a separate view and controller.

The client has these main features, represented in the menu with:

**CBIR Search:** Search with photo taken by the phone's internal camera as seed image.

**CBIR Store:** Storage of photo taken by the phone's internal camera.

**Browse Image:** Search with photo from the phone's internal storage as image (Browse)

In addition to this the client has two more options in the menu, namely "**Settings**" and "**About**".

The **settings** function contains options to turn GPS on or off and options regarding the GPS range.

About contains information about the MMIR2 prototype as well as a list of developers.

### **3.3 The MMIR2 server**

The MMIR2 server uses the data delivered from the client to perform queries and inserts on the database. In addition to the database that performs GPS and CBIR queries, most of the MMIR2 logic lies in the MMIR2 server.

In a default scenario the user uses the client to send a GPS-location, a time stamp (containing date, time etc. based on milliseconds since 1970), a seed image he or she wishes to use as search image as well as a range (in meters). The user will then receive four images returned in a 2x2 grid, each of them having a certain degree of relevance to the seed image.

The course of execution proceeds as following:

### 3.3.1 The client sends data to the MMIR2 server

The client calls a method, `doPost`, on the MMIR2 server, and sends image, latitude, longitude, timestamp and a command. The command is a string containing either “store” or “search”, which starts separate processes.

### 3.3.2 The MMIR2 server stores the data

If the command is **store**, the MMIR2 server stores the image, GPS location and time stamp to a temporary table in the database. This is done to provide the project with extra images for the database. Provided they are good, they will be inserted into the main image table and used for future queries.

If the command is **search**, the MMIR2 server will first call the same functions as store, and then proceed to search. In other words, the search command will actually store, and then continue to search. The separate store function makes it possible for the user to go around the city, taking photos and storing them in the database, without having to wait for the search procedures to finish.

### 3.3.3 The MMIR2 server searches

The MMIR2 server sends GPS location and GPS range to the GPS search procedure in the database. The GPS search procedure selects all objects in a given range, that is, all objects in the radius of a circle, where the GPS location denotes the middle of the circle. The GPS search procedure then calls a CBIR search procedure, which uses the primary IDs of all the objects found with the GPS search procedure (i.e.: finds all images which match the seed image, and which belong to the objects selected by the GPS search).

If the client doesn't find a GPS signal, it will return a **null** value. The MMIR2 server therefore contains conditional statements to test for GPS values. If they are not present, the Image Search procedure will proceed to search without GPS, i.e. search with CBIR only.

### 3.3.4 The MMIR2 server fetches images from BergenBy DB

After the GPS and / or CBIR search is finished, the database will store the images found relevant by the search procedures in a temporary table. This is called a result set. The MMIR2 server then proceeds to fetch the images from the **result set**, one by one, until it has them all.

The MMIR2 server is programmed to organize the images into what I have called “**image sets**”. Each result set is composed of four (or in some cases less) images, and the complete result set is stored in several image sets. This makes it easier for the client to handle navigation between different image sets, as the client will only download four images at a time.

```
1 public ArrayList getImageSet() {
2
3     ArrayList images = null;
4     System.out.println("getImageSet() - begin");
5
6     images = new ArrayList();
7     ArrayList imageSet = getTempImages();
8
9     int i = 0;
10    ArrayList set = new ArrayList();
11    while (i < imageSet.size()) {
12        set.add(imageSet.get(i));
13        i++;
14
15        // Hvis størrelsen på set er lik fire, putt set i images og
16 // opprett ny set arrayliste.
17        if (set.size() == 4) {
18            images.add(set);
19            System.out.println("Set full, set.size() == " +
20 set.size());
21            set = new ArrayList();
22        }
23        // Hvis i er lik imageSet.size, putt set i images.
24        else if (i == imageSet.size()) {
25            System.out.println("i == imageSet.size()");
26            images.add(set);
27            System.out.println("images == " + images.size());
28        }
29    }
30
31    System.out.println("CaimDAO: getImageSet() - done");
32    return images;}
33
34
```

*Figure 2: The source code for the organization of the image sets*

### 3.3.5 The MMIR2 server sends the images to the client upon request

When the MMIR2 server is done organizing, it will tell the client that all went ok. The client will then proceed to ask for images, one by one, based on what image set they are in. Typically the client will automatically download image set 0, that is, the first image set.

**Client:** Send me image set 0, image 0

**MMIR2 server:** *Sending image set 0, image 0...*

**Client:** Send me image set 0, image 1

**MMIR2 server:** *Sending image set 0, image 1...*

**Client:** Send me image set 0, image 2

**MMIR2 server:** *Sending image set 0, image 2...*

**Client:** Send me image set 0, image 3

**MMIR2 server:** *Sending image set 0, image 3...*

### 3.4 The database

The database, BergenBy, is an Oracle 10g Object Relational database initially implemented for the MMIR and VISI systems in 2007. The database currently contains 246 images, mainly from Bergen, Norway, but is under constant expansion. The complete database is described by Marius Langøy (2008)<sup>1</sup> and Bjørge Næss (2007)<sup>2</sup>.

It was necessary to implement a few changes to the database, mainly changes in structure and procedures, to accommodate the changes needed for the requirement specification.

---

<sup>1</sup> Langøy, Marius (2008) [http://caim.uib.no/publications/BergenBy\\_report.pdf](http://caim.uib.no/publications/BergenBy_report.pdf)

<sup>2</sup> Næss, Bjørge (2007) - <http://caim.uib.no/publications/VISI-Report.pdf>

### 3.4.1 Structure

The main structural change made to the BergenBy database to accommodate the MMIR2 prototype is a new table, GPSRESULT\_TAB, which is a temporary table used to hold the results from the GPS search during the process of performing a CBIR search (on the results in the table).

```
GPSRESULT_TAB
    Id (Number)
    Score (Number)
```

Except for this new temporary table, the MMIR2 prototype uses the same database structure as the MMIR prototype.

### 3.4.2 Functions

The CAIM\_BERGENBY package has been changed to incorporate GPS search. There is a new procedure in the package, GPS\_SEARCH, which takes three arguments, Latitude, Longitude and Distance. The procedure is used to perform a GPS search, more specific select all objects which are in a distance less than or equal to the distance given, i.e. the given latitude and longitude of the photographer.

The ID of the resulting objects are placed in the table, GPSRESULT\_TAB, along with the distance (in meters) from the objects to the given latitude and longitude of the photographer.

```

1 PROCEDURE GPS_SEARCH(
2   p_Latitude  IN NUMBER,
3   p_Longitude IN NUMBER,
4   p_Distance  IN NUMBER
5 )
6 IS
7 BEGIN
8   FOR iterator IN (
9     SELECT
10    o.ID,
11    ROUND(
12     SDO_GEOM.SDO_DISTANCE(
13      MDSYS.SDO_GEOMETRY(
14       2001,
15       8307,
16       mdsys.sdo_point_type(
17        p_Latitude,
18        p_Longitude,
19        NULL),
20       NULL,
21       NULL),
22     o.Coordinates,
23     20,
24     'unit=M'),
25    2) Score
26 FROM Object o
27 ORDER BY Score ASC
28 )
29 LOOP
30 IF(iterator.Score < p_Distance)
31 THEN
32 INSERT INTO GPSRESULT_TAB VALUES(iterator.ID, iterator.SCORE);
33 END IF;
34 END LOOP;
35 END;
36

```

*Figure 3: The GPS Search procedure*

The GPS Search procedure is a helper function only used by other functions, that is, the GPS Search procedure is only called by the **SERVER\_IMAGE\_SEARCH** procedure. The IDs stored in GPSRESULT\_TAB are used by SERVER\_IMAGE\_SEARCH in a CBIR search that finds all images that looks similar to the image seed **and** are connected to the objects found with the GPS\_SEARCH function.

Most of the procedures in the package CAIM\_BERGENBY have been changed to support searching of GPS. Most of these changes are however minor changes, so they will not be described further here.

## 4 User manual

Before the user starts the application he or she should open the camera lens (if applicable), **if** the user is planning on taking new photos. It is not recommended to open camera lenses after the program has been started.

### 4.1 *The menu*

When the user first starts the MMIR2 Prototype he or she is presented with a menu containing five options. These are:

- CBIR Search
- CBIR Store
- Browse for image
- Setting
- About.

The user must select one of these options with the **left** select-button, or exit the program with the **right** button.

Most of these options (except Settings and About) use some or several of the phone's advanced features like GPS, camera and Internet (Wireless, 3g). Unfortunately most of these options require the user to agree to use the features, and he or she will thus be prompted numerous times to agree to use either one of them.



Figure 4: The MMIR2 Prototype client menu, running on the S60 SDK Emulator

#### 4.1.1 CBIR Search

After agreeing to use GPS-features the screen will be filled with a standard camera frame, showing a copy of what the camera is pointed to. In addition to this there will be a dot in the lower right corner of the screen. This dot will be either red or green, indicating whether the phone has a GPS signal or not, where the green dot indicates signal and red dot indicates no signal.

It is possible to search for pictures while the dot is red, but in most cases this is not recommended. If the phone is unable to get a signal the user can safely perform a search, but this might have an impact on the hit ratio of the search which might be severely degraded without GPS.

To take a picture, press the **center** button. Hold the phone steady while the picture is taken. After agreeing to show the picture on the screen, press either **Search** to search with the image as search image, or **Discard** to discard the picture and take a new one. The discard button will remove the image, and restart the camera feature, i.e. send the user back to *4.1.1 CBIR Search*.

Assuming the user selects Search, the user will have to agree to send the picture, and choose which connection to use. Then the user must wait while the image is uploaded to the server



*Figure 5: The MMIR2 Prototype client upload message, running on the S60 SDK Emulator*

After the upload, the GPS and / or CBIR search will be processed, and the client will download the first four images from the image set.



*Figure 6: The MMIR2 Prototype client download message, running on the S60 SDK Emulator*

The user will be notified of the process with the help of a custom information screen showing what is going on. In the picture above, figure 6, the information screen says “Downloading image” and a number of bytes. The number of bytes will change from image to image, to help show the user that something is happening and that the client isn't stuck.

When all the images are done downloading, the user will be presented with a 2x2 grid containing images sorted by relevance, where the highest rated image is in the top left corner and the lowest rated image is in the bottom right corner. Each image is also presented with its text caption.



Figure 7: The MMIR2 Prototype client search result, running on the S60 SDK Emulator

#### Possible buttons to press:

**Menu:** To get back to the start menu.

**Next:** To download and show the next four images.

**Prev:** To download and show the previous four images.

**Open:** To open the selected image in a full screen view.

#### N.B.:

- Due to the mobile phones limited set of buttons, the user has to press “Options” to get to the buttons Prev, Open and Menu. The Option choice might change sides, or be present at both sides of the screen, depending on what button the user is trying to press as well as what image set the user is viewing.

- The user will only be presented with the **menu** button when he or she is at the first or the last image set. That means that if there are for example four image sets (4), and the user is at the second set (2) and wants to exit or go back to the menu: the menu button will not be accessible before he or she reaches either the first or the last image set. This was actually designed this way intentionally, but in retrospect the author realizes that this is really, really stupid, and should be changed in future projects.

If the user presses **Next**, the client will present the user with an information screen (see figure 6) and download the next four images. If there aren't four images to download, for example if the total images found is 7, the client will only download three more images.

If the user presses **Prev** (which only is possible if the user is viewing an image set that's **not** the first), the client will present the user with an information screen and download the previous four images.

The **open** and **menu** buttons should be self explanatory: open opens the image in full screen, and menu sends the user back to the menu, making it possible to use other features, alter settings or exit the application.

#### 4.1.2 CBIR Store

The CBIR Store option works just as the CBIR Search feature, with the only difference being that CBIR Store will not return an image set. Instead of a **search**-button, CBIR Store will have a button named **store**. The discard button is present like in the CBIR Search feature.

CBIR Store is used to upload and store images in the temporary table on the BergenBy Database, without having to wait for the image sets to be downloaded. CBIR Store is useful for researchers who have a need to store images directly to the database or if the user finds an interesting objective while traveling etc.

The reader should note that both the CBIR Store and the CBIR Search feature will store the seed image, making it possible to use the seed image as an image in the database if the developers should so decide.

### 4.1.3 Browse for image

The browse for image option is basically a CBIR search with the seed image being an image stored on the phone's solid state hard drive instead of being captured by the camera. The feature works just like the Search-feature, except the fact that the GPS function will not be used. For information about how to use this, see *4.1.1 CBIR Search*.

The fact that the CBIR search isn't combined with the GPS search will with a 100% certainty make the search result less precise, but since the image is stored on disk we have no way of knowing where it was taken and it is thus impossible to improve the search with GPS data.

## 5 Future development

The MMIR2 prototype is without a doubt a prototype, and should under no circumstances be considered a finished product. However, I still think there are improvements to be made, and they are as following:

- More meta data about the objects: at the current time the only meta data about the objects are their name, and that does not suffice. Considering that the MMIR2 prototype is meant to be a program tourists can use to get information about buildings in Bergen, the program should really have some way of presenting said information. At the time of writing the only information the user is given about the selected objects are their names. History, date of building, historical events and current events should be presented.

- The program should probably implement different presentations for image sets other than the 2x2 grid.
- Settings should be saved to hard drive or database. That way the user doesn't have to turn the GPS feature of manually every time he/she wishes to use it. The settings for range should also be saved – at the time of writing they are hard coded to 100 meters and the user needs to select a new range for every time he starts the program if he for example is further away.
- The view of selected images from the result set has a wrong aspect ratio. This should be fixed! The same goes for the presentation of images taken by the camera.
- The menu button should really be accessible from all the image sets, not just the first and last (in the result set view mode).

## Acknowledgments

Roe Fyllingsnes, graduate student at the University of Bergen and developer of the first MMIR prototype, has been of great help during the development of the MMIR2 prototype. Mr Fyllingsnes is working on the MMIR2 prototype for his master thesis, and has been helping the developers on the CAIM project during the summer of 2008. Without his expertise and thorough knowledge of the J2ME SDK the MMIR2 prototype would not have been nearly as complete. Thank you, Roe.

## References

Næss, Bjørge. (Sept. 2007). *The VISI Prototype for Image Retrieval*. CAIM-TR-1, Dept. of Information and Media Sciences, Univ. of Bergen.

Fyllingsnes, Roe. and Hartvedt, Christian. (Sept. 2007). *The MMIR Prototype for Mobile Image Retrieval*. CAIM-TR-2, Dept. of Information and Media Sciences, Univ. of Bergen.

Langøy, Marius. (Sept, 2008) *BergenBy Database and Metadata Editor*. CAIM-TR-5, Dept. of Information and Media Sciences, Univ. of Bergen.

## Appendix

### ***Appendix A – Program code***

Due to copyright and space consideration the source code for the MMIR2 prototype (The client and the servlet) will not be included in this report. The source code will however be available on CD-ROM. CAIM is free to distribute the source code to CAIM affiliates via the medium of their choice at their leisure, so if you are in need of this source code – contact CAIM.

### ***Appendix B – Database code***

The database structure used by MMIR2 is documented by Marius Langøy (2008), and as such I will not dive further into the structure here. The MMIR2 Prototype uses an Oracle package, CAIM\_BERGENBY, made during previous CAIM projects, and I have had to modify this package to suit the new features. The resulting database package is available upon request to <mailto:joan.nordbotten@infomedia.uib.no>.