M2S and CAIR Web based Photo Management Solution in Mobile Environment.

KIRTI RAHEJA
STUDENT
M.D. UNIVERSITY ROHTAK, INDIA
kituraheja@gmail.com

Abstract – 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, 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.

1. Introduction

There exist several web-based photo management systems such as Flickr and Snapfish 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.

1.1 Web-based photo management solutions

There exist several web-based photo management systems such as Flickr and Snapfish. 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.
The Cyberguide project 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 “Whopainted 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 1: Traditional information retrieval scenario.

Figure 2: Map and information interface 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 3: Shows a broadcast message to the individual Cyberguide units

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 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.

1.3 The Photo Navigator System

Pauty et al. proposes a system that makes users able to navigate through a photo collection using context. Photo Navigator makes use of the user's 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.

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.

![Image](image_url)

**Figure 5:** Shows how a user can go from physical to virtual navigation.

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 a 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 its surroundings from The Second World War and how it looked like then. 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 with any 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. 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.

### 1.4 Lincoln

Microsoft has developed a research prototype of a service they call Lincoln. 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 argues 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.
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. 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 whole image. This is because they consider the odds that there exist many images with the same three data sets are small. 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. Currently, Lincoln can only be downloaded for free using Internet 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 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). 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 weblink 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 moviereviews 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. 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 have 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 evolution 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 phonenumber. 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 stable 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 serverside until the user is in range again. People are also more used to MMS and are familiar with how to wrap and send MMS’.

1.5 SnapToTell

Lim et al. proposes a service called SnapToTell (also called Snap2Tell).

The proposal makes users able to retrieve information about sights by using their mobilephone. 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.

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.

Figure 8: An application scenario with SnapToTell.
1.6 Mobile Access to Personal Digital Photograph Archives using context
Gurrin et al. 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.

1.7 CONCLUSION
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 explicitly 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.

1.8 REFRENCES
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.
3. Zhao Rong, W. I Grosky, "Bridging the semantic gap in image retrieval." Distributed Multimedia Databases: Techniques and Applications, TK. Shih(ED.), Hershey, Pennsylvania,


