PlaceMemo: A Prototype of a Context-aware Information System

by

Anton Gustafsson

Abstract

This master thesis discusses issues concerning the construction of the PlaceMemo prototype. The PlaceMemo prototype is an application designed based on ethnographic studies, to support road inspectors in their work. This involves tasks such as managing, collecting, storing, employing and distributing detailed information in a vast working area were the workers often finds themselves isolated from their colleagues.

The ethnography resulted in the design ideas of a location dependent information system implemented on a mobile handheld Pocket PC device. Unlike most others location dependent information systems PlaceMemo let the users themselves provide the information stored in the system. The user adds information to the system in the form of voice messages. The voice message is appended with positioning data and thereby becomes a place memo. The messages added can later be accessed either through the users location, a GPS positioning system is used to augment the users environment with vocal information, or via interaction with a visual information space.

The task of building the PlaceMemo application in accordance with ethnographic findings and initial ideas consisted of

- Assembling the appropriate hardware platform,
- Implementing system functionality to retrieve, process, search and store both voice and GPS data.
- Creating a visual information space representing the collected information.
- And to create means of interaction for both ultra mobile and stationary situations using vocal, visual and motoric means.

The visualization was done as a 3-dimensional map metaphor resembling the cartographic maps used by the road inspectors. Information is accessed through position and date properties. The travelled path is plotted and used as a visual reference in the map. Effects caused by positioning errors and receptions failures are limited by the use of interpolation and error perimeters. Searching is made efficient through setting up a safe perimeter. Interaction while driving is done through tactile and vocal means.

Index

1 Introduction

In today's technology-imbued society our lives becomes increasingly dependent on the aids that information technology provides. This applies to when we are at home as well as in work for most of us. Computers can be found, to at least some extent, in most professions. The driving force behind this is often the need for companies to increase efficiency and lower costs in an environment that now not unusually demand more information to be handled than before.

As the computers and information technology tends to advance towards smaller and more flexible devices, a trend fuelled by the mobile phone industry's constant strive towards diminutiveness, new areas and situations are revealed as suitable for information technological appliances. These new situations however needs to be fully understood or the efforts being made to make a work task more efficient can sometimes prove to be misdirected.

At one of the Interactive Institute's research studios, the mobility studio, there is one such new situation in particular that is found to be of interest. The situation, as the studio name implies, includes information technology used in a mobile situation. Although even more interesting is thought to be the certain situation when a person is in a mobile setting collaborating with others. In order to gain knowledge about this specific situation information is often collected through ethnographic fieldwork. Occasionally, as in a fieldwork carried out on road inspectors at the Swedish road administration office, the study inspires to new ideas for design.

The study on road inspectors offered researchers an opportunity to study and evaluate an existing computerized information system. A number of findings were done. The existing solution, it became evident, was designed from an organizational point of view and did not provide support for the kind of mobile working situation the road inspectors were exposed to. The solution were thought to be an complementary system supporting the individual in the tasks of remembering and collaborating with colleagues while constantly being on the move.

The system has been named PlaceMemo. Among other things PlaceMemo aims to support tasks such as managing, collecting, storing, distributing and employing detailed information in vast working areas. The concept and the ethnographic study was originally described in the article *PlaceMemo - Supporting Mobile Articulation in a Vast Working Area Through Position Based Information* [1] by Oskar Juhlin and Mattias Esbjörnsson.

1.1 Thesis outline

The intent of this master thesis is to interpret and transform the PlaceMemo idea into a functional prototype. This includes, based on the provided ethnography, to determine the graphical, physical and technical form for these ideas as well as contribute with own ideas where existing design criteria needs to be completed. It also means adapting the solution to fit the limitations within mobile hardware used. Although the focus of this master thesis will primarily be on technical details concerning the implementation, the intention is also to make connections to the ethnographic background material and related research areas such as context awareness and mobile computing visible. The anything but obvious path from ethnography to design will naturally be a concern.

The application built is a prototype. A prototype is often seen as the instance that will serve as basis for later stages. It is a probe for collecting information. From the system designer's view, the construction of this master thesis can therefore be viewed as the completion of the first step in an iterative ethnographic design process. The information retained from prototyping will be used as the starting point for a second revolution. With this notion experiences made during the development will be described together with suggestions for future improvements to the application.

Since the PlaceMemo prototype is part of a research project, the information retained from evaluation will also serve another purpose. From the researcher's point of view evaluation of the prototype is intended for helping us make conclusions regarding the quality of the more general concept behind the design. These conclusions will in turn generate knowledge in how to design future information technological systems in similar circumstances.

Information retained from the evaluation following this master thesis will reveal flaws in design on all levels. This master thesis will therefore be concluded with a brief discussion concerning how to distinguish results effected by the implementation form or limitations imposed upon the prototype from those in relation to the overall design concept.

2 PlaceMemo the idea

A place memo is a voice memo appended with information about its position. This construction is the most fundamental part of the PlaceMemo[1] concept. The design is based on the notion that entering information in the unrestrained and effortless way that the making of a voice memo forms, will support the user in simultaneously performing other tasks such as moving through, observing and analyzing his or her surroundings. By a simple push of a button both the location and any information about this location can be stored, with the minimum of effort by the user.

The information entered into the system will through GPS positioning be used to augment the user's environment with vocal information. Once revisited, the approach to a location with previously stored information will trigger playback of the voice memo associated with that specific location. In this way a place memo is thought to act as a reminder, simultaneously providing the user with essential information about the task at hand. In mobile situations the user's location is used as the primary mean of interaction.

During non mobile situations or situations were the user can afford to provide a higher level of interaction the PlaceMemo concept suggests a visual mode of interaction, enabling planning as well as information managing. As part of managing we include distribution of place memos between devices. The original idea suggested that this should be accomplished by the ability to send place memos using SMTP, a protocol normally used for sending emails. In this prototype however, distribution has been left out due to time constraints.

2.1 Background

The outlines for the PlaceMemo prototype came to take its form in the summer of 2000 during which two of the studio employees (Mattias Esbjörnsson and Oskar

[1] While PlaceMemo refer to the application built, place memo written as two words will refer to an individual voice memo appended with positioning data.

Juhlin) were conducting their ethnographic field research at the Swedish national road administration office. It was concluded that the road inspectors experience a very unique working situation. Analysis of the ethnographic material resulted in the PlaceMemo concept. It was suggested that a prototype were to be built and tested on the road workers in order to evaluate the design suggestions. The work done so far was summarized in a paper by Liselott Brunnberg and Mattias Esbjörnsson, *PlaceMemo: Using GPS and Mobile Computers to Augment the Roads* [2]. During this time a very basic prototype was also developed by Liselott Brunnberg and Mattias Esbjörnsson. Using a laptop connected to a GPS-receiver this prototype was used to test the very basic technical concept of location linked voice messages. The results were inspiring and it was decided that a full-scale prototype on a more appropriate platform suitable for testing on actual users should be built. A full paper on the subject was published in 2002 by Mattias Esbjörnsson and Oskar Juhlin [1].

The prototype was from the beginning meant to be brought back to the original setting for evaluation. But since the collaboration with the Swedish national road administration had ended this was no longer possible. The PlaceMemo project has instead been continued in collaboration with Cleanosol, a company that performs maintenance, inspection and painting of the white markings on the roads. The conditions for some of the workers at Cleanosol are very similar to those at the Swedish national road administration. During the construction of the second prototype complementary interviews have been made to investigate the new situation.

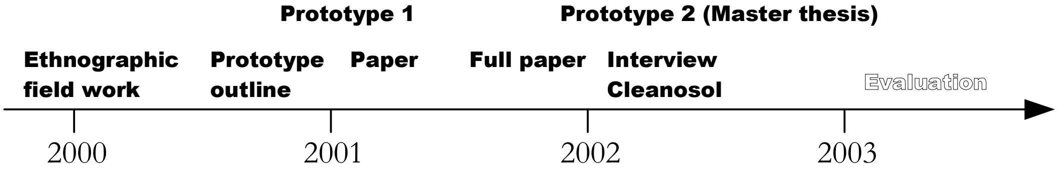


Figure 1. The PlaceMemo project in its whole.

2.2 The field study

During the field study at the Swedish road administration office, Mattias Esbjörnsson and Oskar Juhlin found that working conditions for the road inspectors are both highly mobile and often demands coordination and collaboration. The situation is also distinguished in the way that the inspectors cover very large areas and most of the time find themselves isolated from their colleagues.

Figure 2. A road inspector.

2.2.1 Working on the road

During a normal day the road inspector might travel up to 250 km on a 7 hour predetermined tour. During this time he is constantly on the look out for things that needs to be repaired or attended to. Among many things this can involve dead animals on the roadway, abandoned cars, missing road signs, broken fences or potholes in the road. To aid the road inspector their mobile offices are equipped with a cell phone, a UHF communications radio, maps, post-it notes and a Prodata unit. An ordinary FM radio is also used to monitor public traffic reports.

In most cases where communication with colleagues or authorities is necessary the cell phone is used. The communications radio is rarely used at all. Identified defects can be dealt with on the spot, postponed to a later occasion or delegated to a college or public authorities with better capabilities to handle the problem.

2.2.2 Prodata

Figure 3. The mobile office.

The Prodata system currently used, is a tool for writing formal reports. To make a report predetermined defect codes are used. A dead animal has one code while an abandoned car has another. A list of the codes can be found in the Prodata manual. The inspectors often need to consult this manual in order to get the correct code. When entering a report there is also a possibility to add extra textual information. The system stores the position coordinates, obtained through GPS, together with the report in a database. The database is not accessible for the user through the ProData unit, so once entered the information is lost in the road worker's perspective. The database is later updated against a central database to be used as a basis for debiting the orderer. The road inspector can only access previously stored information when in office.

2.2.3 Deficiencies in design

The research group concluded that the Prodata system was designed from the notion that when a fault is spotted the inspector stops the truck, does a formal on the site report on the Prodata system and there after takes appropriate measures to remove the problem, a sequence consisting of identification, report and repair. In reality there are many circumstances that causes the inspector to deviate from this ideal line of actions. Three common conditions that often cause repair or identification to be postponed are listed here.

- **Inadequate information** – The inspector is unable to make the decision if an object needs attention or not. He might for example see a car parked by the road. It might be abandoned or it could be that the owner simply has run out of gas and went for more. He decides to make a note. If it's still there the next time he will report it and have it removed. In other situations the inspector might suspect that a road sign is missing but he needs to investigate this further in order to be certain.

- **Lacking tools** – The inspector sees an unlocked safety fence gate. Since he doesn't have a padlock he makes a note to bring one the next time.

- **Difficult traffic situation** – The inspector might be unable to stop when the traffic is really heavy or he might have spotted the problem too late in time to stop.

The ProData system requires the user to be on the spot when entering a report. Consequently the vehicle has to be stopped in order to do so. The road inspectors often want to avoid stops since they are both time-consuming and hazardous. In dense urban traffic the situation might also prevent you from stopping. In cases when the repair was postponed it was noted that reporting to the Prodata system also often was. Inspectors were often uncertain about, if an item already had been reported or not. As a result a duplicate reports were sometimes made while in other cases non at all. Prodata system in the end forced road inspectors to entirely rely on their own good memory, notes on a piece of paper or post-its in order to efficiently conduct their work.

2.2.4 Post-its

Post-it notes in contrast to the reports entered into Prodata have the advantage of being accessible all the time. If the road inspector later changes his mind about reporting an assumed abandoned car to the police, information such as the registration number of the car etc. will be at hand. The disadvantages with post-its are that it is hard to describe the position of an identified problem. It's also hard to organize and utilize large amounts of information stored in this way. They do however posse's properties that makes them ideal for face-to-face delegation of work between colleagues.

2.2.5 Conclusions made in the field study

To conclude some of the findings by the ethnographic fieldworkers, the following can be said.

- Inspectors often forgot things they were supposed to do, due to the large information volumes. Passing an objected in need of repair often made the road inspectors recall previously forgotten jobs. Items in the environment acted as reminders for tasks to be done. Once an object was spotted however the inspectors often had little or no time to stop.

- The inspection areas have to be managed individually since each individual had to rely on his own memory in order to remember tasks to be done and furthermore which of those tasks that had been reported into the Prodata system or not. This was concluded to be the prime barrier for collaboration between road inspectors.

- The Prodata system is designed from an organizational point of view. It does not only fail to support the road inspectors in their work, it even aggravates it in some cases. One of the effects from this is the poor quality of the reports contained within the Prodata report database. In the end the system's inability to support the users causes it to fail its purpose even from the organizational point of view.

From the conclusions made from the ethnography the PlaceMemo idea was developed to provide a complement to the existing system, if successful possible also to be integrated into the existing system.

2.3 Ethnography as a basis for design

The method of using ethnography as a way to do research has become increasingly common. This method has its roots in field of anthropology were it has long been used to gain insight in everyday life of people living in societies very different from ours. The technique was adopted in systems design in the 1980s after it was discovered that simply observing a person and an artefact in a closed laboratory environment wasn't enough. System designs often failed because the design paid insufficient attention to the overall context of the work [3]. Today ethnography is very often practised in CSCW [2]since it is a brilliant tool for capturing the social context of a situation [4]. At the mobility studio the ethnographic method fit requirements resulting from the often social and collaborative focus in the research.

Conducting ethnographic survey, basically means to observe and document activities and research subjects in the natural setting in which they occur. By doing so the ethnographer is able to focus on what people actually do and not on their accounts of behaviour. The ethnographer is generally expected to be objective, free from any preconceived notions and minimize his own influence on the situation as much as possible. To remain objective is a big challenge for the ethnographer.

From a system designer's point of view the challenge is to utilize the ethnography and transform it into a design. The relation between ethnography and design is a commonly discussed issue. The output of ethnographic research is often both extensive and discursive and rarely fits nicely into a system designer's charts and diagrams. The information has to be sorted and interpreted; something that is often done on pure intuition.

Many times the problem of translating ethnographic findings into design is described as being the result of difficulties in interdisciplinary collaboration. It is thought that different vocabularies and views of the problems coming from the different disciplines makes it hard to communicate and convey ideas among the different actors in an interdisciplinary work [5 , Ch 1].

The ethnographer needs to translate field studies into a form accessible to the designer. This is often, as in this project, done in the form of an article. But as Y. Rogers and V. Bellotti [6] points out it is very important to supplement the written material with an active dialog between the ethnographer and the designer and to not exclusively rely on written documents. Two-way communication often tends to be the most efficient way of conveying ideas. In the dialog between ethnographer and designer, Rogers and V. Bellotti advocates the use of scenarios from the fieldwork. This idea is taken even future by Andy Crabtree et al. [7] who consider scenarios to be something of the common language between the parts involved in an interdisciplinary work.

In this project the interdisciplinary work was less of a problem. A far more difficult problem has instead been to handle the ethnography's inability to predict the future. An ethnography only describes, "what is" and not "what will become". Once new solutions are introduced into the equation it tells us nothing about the outcome. This is particularly a problem, when the design implemented creates completely new situations for the user. In the PlaceMemo project for instance, we find support for design decisions concerning placing a memo to a larger extent, than we do for decisions concerning the part involving the visualization. While placing a memo occurs in an already existent situation, where it replaces similar actions with the Prodata system or Post-its, the visualization creates an entirely new situation.

In either case fictive scenarios can be used as a tool to further analyze design criteria concerning low-level implementation details. The creation of these fictive scenarios is done based on findings in the ethnography when ever possible. A method similar to the use of analogies for creating scenarios as suggested by John M Carroll [5 Ch. 6] in his method of artefacts and scenarios. In cases where analogies are inefficient or can't be found we turn to cognitive modelling or other known theories of human activity.

[2] Computer Supported Collborative Work.

During the PlaceMemo project scenarios have been a redundant element, both to be used when communicating but also as a tool for testing and shaping the prototype (the artefact), as suggested by Carrol. The interdisciplinary orientation of the people involved has made the communication very easy. The roll of designing has been distributed between ethnographers and engineer. As the ethnographers provided the general ideas, the engineer has through an ongoing discussion with the ethnographers been able to work out the details. The later stages of the PlaceMemo design process could be compared to the kind of participatory design with the ethnographer acting as the stand in for the user, proposed by Bentley [5 Ch. 8].

The ultimate key however, to insure success of the prototype, must be attributed to the constant small-scale testing present during the entire development process. By continuously alternating building with testing, in an empirical or kind of trial and error method if you like, flaws in the implementation could on an early stage be detected and functionality improved.

2.4 Related works

In order to relate to the PlaceMemo prototype we begin by defining it. As PlaceMemo utilized the users current position in order to simplify interaction it can be recognized as a context-aware application [8]. The fact that it provides several modes of interaction to match the user's abilities during the different situations in which it will be used, furthermore implies its context aware nature [9]. Since PlaceMemo handles information it can consequently be defined as a context-aware location-dependent information system. In another sense PlaceMemo could also be described as an AR-application[3]. As the user moves the application will continuously augment the user's environment with voice memos. It does however differ vastly from what perhaps commonly is thought of as augmented reality systems while it doesn't make use of any fancy graphical equipment such as semi transparent VR-helmets or comparable devices.

In contrast to most location based information systems PlaceMemo differs in the aspect that the information stored in the system is provided by the user solely. The otherwise more common approach in location based information systems is to provide only fixed predefined and supervised information. A large group of systems, among them Cyberguide [10] are designed as electronic guides. A few systems however seek to combine both predefined and user added information. An example of this would be Geonotes [11], an application enabling users to post graffiti like electronic messages among each other.

A message when posted in Geonotes becomes connected to the locations at which it was posted. It can later be received by anyone passing the area equipped with a Geonotes device. The designers behind Geonotes stresses the importance of letting the users participating in forming the information space. They also point out that in order to make this free information space feasible on a larger scale, extensive information filtering will be needed.

While Geonotes often contains public messages sent from one person to a random number of unknown individuals, PlaceMemo mainly contains private memos intended for the user's own purpose. The PlaceMemo concept however, just as Geonotes, does include the ability to post messages to other users. We can imagine

[3] Augmented reality

Geonotes as a virtual information space common to all users augmenting an area of the world. Filtering in this construction is necessary to adapt the augmented world to the specific user's context. PlaceMemo on the other hand could be seen as a system maintaining a personal virtual information space for each user, designed especially for the user's private context. Information elements in the spaces are occasionally, and when so always deliberately, copied from one user space, into another's. Since sharing of memos is done in a considerably more directed fashion, filtering never really becomes a problem. The sender will always provide the information on the notion that it is of importance to the receiver. The relevance of the information can as a result basically always be assumed to be high for the receiving part.

Technically the PlaceMemo and Geonotes prototypes appears to be even more different. While Geonotes has gained a server-client architecture, resulting from the distribution needs PlaceMemo uses peer-to-peer communication. Information in Geonotes are kept on a central sever unit while place memo messages will reside locally on each device.

Another location dependent information system, that perhaps resembles PlaceMemo to an even larger extent is ComMotion [12]. The application monitors the user's behaviour in order to learn frequently visited locations. It then associates to do lists to those positions. Just as PlaceMemo the Commotion system uses GPS positioning. It also shares the concept of providing so called just in time information deliveries as reminders for tasks to be done at the current location. Furthermore it utilizes multiple modalities (both vocal and visual) when interacting with the user in a similar fashion that PlaceMemo does.

A difference between comMotion and PlaceMemo is that ComMotion is built on the notion that there are certain locations that are more likely to be associated with information. Since this is not true in the road-inspector's situation, the PlaceMemo application has been designed to support arbitrary placing of information on any position. The PlaceMemo prototype is also more specifically adapted to function within the ultra mobile situation that occurs while driving and working at the same time. It has for this purpose adopted a minimalist approach and relies entirely on the traditional voice memo as the information carrier. ComMotion on the other hand makes use of both speech recognition and text messages.

3 The prototype

The PlaceMemo prototype will be used in two different contexts, one while placing and triggering and the other while visualizing and managing place memos. Placing and triggering represents functionality used as the user are working on the road. This situation is characterized by a low bandwidth and a non-visual mode of interaction. The situation is carefully described in the ethnographic work by Oskar Juhlin and Mattias Esbjörnsson [1].

In the other situation we imagine the user to be more focused on the device. Interaction can in this case be allowed to be far more visually demanding. The situation is thought to occur previous to an inspection, during a break, or once the inspection is completed. The primary objective for the user in this context will be to remotely access information regarding a certain area. This is done by utilizing the available hardware, in order to produce a representation of the collected memos. The user will use the information in order to plan ahead and equip himself with appropriate tools needed for the inspection. He will during these situations also be able delegate work to colleagues and possible also to translate place memos into formal reports on co-existing systems like Prodata.

The implementation of the PlaceMemo prototype has been divided according to the two, above mentioned, contexts both in the system design and in the layout for this report. A brief description of the selected hardware platform will therefore be followed by a part describing the two actions of placing and triggering memos. Following this section will then be a description of the components involving visualization and management of place memos. In the visualization and management section however, the part involving the distribution of PlaceMemos has been excluded since it is yet to be completed.

3.1 The Platform

In order to provide a platform for the prototype we utilize existing hardware components. In accordance with the original idea the main component of the platform will consist of a handheld computer. An initial step of this master thesis project was to determine which kind of handheld and what additional peripheral devices, needed to provide necessary functionality and to physically put the given design ideas into practice.

3.1.1 Pocket PC devices

In the group classified as handheld computers we find several standards of devices defined. One such is the Pocket PC, a standard provided by Microsoft. The Pocket PC device is distinguished by its touch screen display together with a few hardware buttons for user interaction. This device aims to be more like a stationary PC when it comes to functionality and performance yet remaining the small size and mobile features of a PDA[4]. Many of the new models provide high-resolution color screens and built in features such as Bluetooth and Wireless LAN.

3.1.2 Symbol vs. IPAQ

There are several different brands of Pocket PCs currently competing on the market. Manufactures such as Compaq, Symbol, Toshiba and Siemens all have their own set of Pocket PC models. In this project devices from Compaq (HP) and Symbol were tested and used. The Pocket PCs from Symbol are designed to meet the more demanding environment that industrial use sometimes involves. This is the reason why it in the beginning looked as if the Symbol units would be the devices best suited for the PlaceMemo prototype, considering its operating conditions.

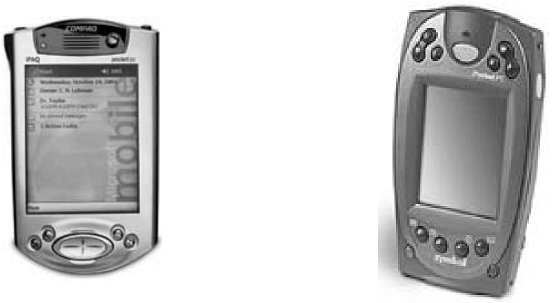


Figure 4. Pocket PC devices HP (left) and symbol (right).

[4] Personal digital assistant

Unfortunately the Symbol units suffered from a couple of drawbacks that made them hard to use. The main problem with these units was poor speaker quality that made it very difficult to listen to voice memos in noisy environments. Some of the HP models also had this problem but since the HP devices as a standard are equipped with an external headphone connector this problem could be avoided by using headphones or connecting the device to the car's speaker system.

The Symbol machines in addition also lacked standard buses or expansion slots such as PC Card, USB or Bluetooth to be used for connecting GPS equipment or additional memory. In addition none of the Symbol units available at the time of the project where equipped with a colour display.

3.1.3 GPS receivers

The Pocket PC platform provides necessary hardware for both playback and recording. It also provides us with a somewhat limited storage capacity. A third crucial component however is missing. In order to create a PlaceMemo, positioning-data obtained through a GPS receiver is required. The GPS receiver will in the future hopefully be an integrated part of the handheld computers. Currently such a receiver has to be connected externally. In this project we have tested tree different receivers[5] in configuration with an HP Ipaq Pocket PC. The most favored configuration became the use of the Bluetooth-equipped HP Ipaq 5450 in combination with a Bluetooth GPS-receiver (BTGPS). This solution requires no cable connection between the receiver and the application platform. The Bluetooth GPS-receiver from EMTAC also provides a better positioning accuracy than other receivers tested [29]. In addition the built in rechargeable battery makes external power cables or connections to the Ipaq device's battery unnecessary.

3.1.4 The operating system

The Pocket PC operating system present on the Pocket PC device is basically the Windows CE operating system with a few alterations. On this platform application-programming interfaces (API:s) for both recording and compressing sound can be found. The Pocket PC operating system also provides basic means for the creation of databases. The database functionality however later proved to be rather unstable and in the end became responsible for some of the greatest shortcoming in the prototype (more about this under the Future work section).

A disadvantage concerning the operating system is the limited graphical interface library provided. The interfaces implemented on Pocket PC are many times down scaled replications of its desktop counterpart. These tends to supports the mobile situation very poorly. The standard graphical library has for this reason as far as possible been avoided. The implementation instead uses the Pocket PC:s game API (GAPI) which is primarily intended for game development. The game API supports only the most basic low-level graphic functionality. It does however give you complete freedom to design your own interfaces and to use existing hardware buttons on the device at will.

[5] Redknows, EMTAC, RoyalTek

3.2 Placing a memo

In order to place a memo there are mainly four kinds of functionality needed.

- The means to interact with the user who initiates the placing of a memo,
- the ability to perform an accurate positioning,
- means to create the voice memo,
- and finally a way to store the information.

3.2.1 User interaction

In order to initiate the placing of a memo the user presses a button. The situation in which this occurs is, as mentioned before, often while the user is driving his vehicle and simultaneously analyzing his surroundings. Analyzing as well as driving requires that the driver is visually focused on his environment. Visual focus is also necessary since the placing of the memo has to be done the instant the vehicle passes the defect identified, in order to get an exact geographical position. Consequently the interaction with the device at this point must be designed so that it can be preformed without, or with limited, use of the user's eyesight.

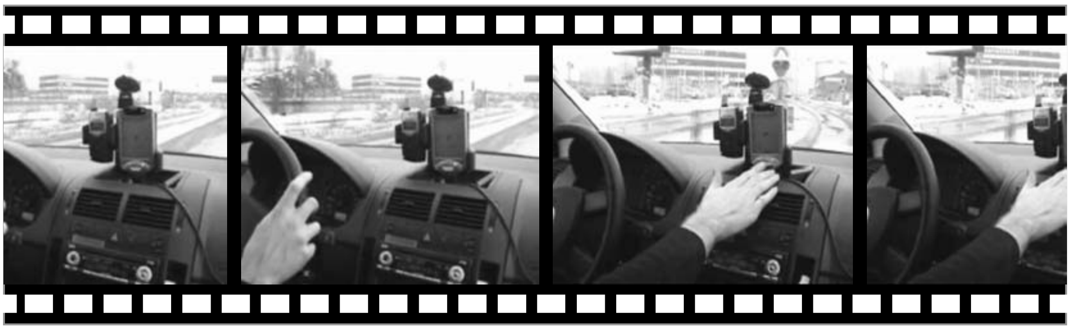


Figure 5. The video sequence shows the placing of a memo using a hardware button on the device. Bumps in the road cause the user to use the instrument panel as support when aiming at the button. The user lets the finger rest on the button until the right moment for pressing it occurs.

This immediately rules out the use of logical buttons defined on the Pocket PC:s touch screen. A touch screen lacks the distinguishable form and tactile definition of area that a hardware button possesses. It also lacks the motoric feedback present in the construction of a hardware button. In this particular case having a logical button occupying the whole screen area would perhaps solve the first problem since the touch screen is surrounded by a frame that defines its area. Feedback could be generated through audio. However, at the same time as the fingertip would find this area it would trigger the placing of a memo. During preliminary tests it was discovered that the user is likely to first place a finger on the button in order to prepare for the right moment to occur before pressing it. A hardware button preferably placed close to where the user normally keeps his hands (usually in close proximity to the steering wheel) would in this case seem to be a better choice.

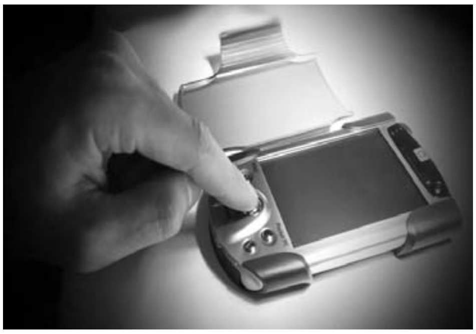


Figure 6. Using an existing hardware button.

The PlaceMemo application currently provides two different solutions to the "button" problem. The first solution uses a hardware key present on the Pocket PC device. The problem with this alternative is that the buttons on the Pocket PC device generally are hard to use since they are both small and inset mounted. The user is also likely to press the wrong key since they are all situated close to each other. By mounting the device just below the windshield the use of periphery vision can be made in order to operate the device while focusing on objects outside the cabin.

The second solution makes use of an external hardware button connected to one of the Pocket PC:s communication ports. The choice of this button can be carefully adapted to fit the perceptible needs. It can furthermore be mounted directly on the steering wheel and by doing so, gain the advantage of using the steering wheel itself as a tactile reference for finding the button. The drawback is of course the cable connection required between device and button.

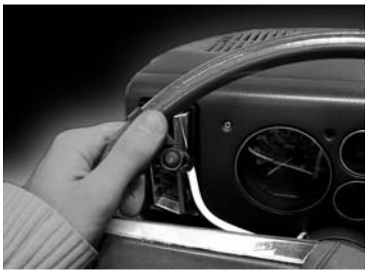


Figure 7. The button mounted on the steering wheel.

When using the PlaceMemo application outside the cabin the second solution can obviously not be used and the cable will have to be disconnected. The first alternative using a hardware button present on the device will in other words always be necessary. Hopefully the advancement of cheap Bluetooth circuits will be able to provide a solution to the discomfort of unplugging the device while moving out on foot. In future cars we could also imagine the device to be able to communicate with the interfaces of cars and in this way make use of existing buttons on the steering wheel or instrument panel.

Once the button is pressed the PlaceMemo uses a complementary acoustic feedback to signal that the recording of a voice memo has begun. Just to be on the safe side the entire PDA screen also turns red to indicate this. An acoustic feedback is produced in the form of an answering machine metaphor. First a message saying, "Recording commences" is played and then a sharp beep is sounded to indicate this.

3.2.2 Positioning

In response to the placing of a memo the system immediately needs to perform positioning in order to determine the current position. A seemingly straightforward problem, but once analyzed this problem in fact becomes rather complex. To understand why we need to know some of the basics about GPS-positioning [13] [14].

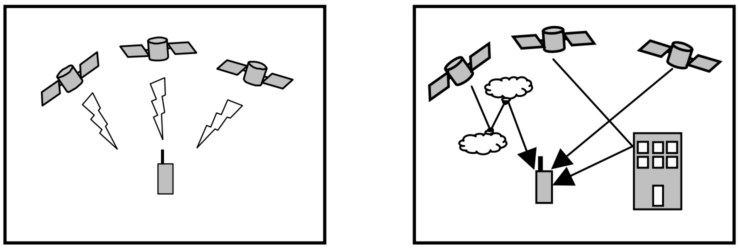


Figure 8. Positioning in the GPS system is achieved through triangulation. The signal delay between the receiver and a minimum of three satellites are used calculate the distance between satellite and receiver (left). Errors are mostly caused by signals being reflected causing the path it travels to become longer (right).

A well-known issue with GPS or any other positioning method for that matter is the accuracy. Interference from object and atmosphere causes signal errors that in turn causes positioning faults. While compensating for these faults, any indication of the magnitude of the is of great value. When using triangulation, as is done in the GPS-system the magnitude of the positioning errors are dependent on the magnitude of the signal error in combination with the angle to the satellites. An acute angle will amplify the positioning error while an obtuse will reduce it compared to the original signal error [13][14]. There is currently no way to determine the current accuracy in the GPS system without using additional reference systems. The angle to the satellites however can in most cases be used as a clue for estimating the positioning error. To do this an assessment of the average signal error is needed.

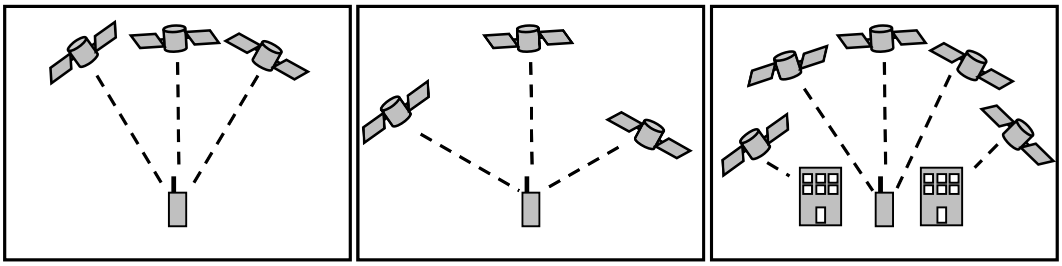


Figure 9. An acute angle between satellites causes small signal errors to become large positioning errors (left). An obtuse angle does the opposite (middle). Why this is a problem particularly in urban areas, is illustrated by the rightmost picture.

Accuracy in GPS-positioning is sometimes improved using statistical methods. In this case however such methods are not applicable since positioning needs to be done instantly. The conditions in the ionosphere, which are the largest contributors to signal errors change very slowly meaning that the errors correlate during short periods of time. Data will consequently need to be sampled for several days on the same spot in order to improve the position accuracy.

During certain conditions positioning is not possible at all and periods of reception failures will be a reoccurring element. The around 24 satellites in the GPS system constantly transmits positioning signals in the form of microwaves (a radio signal with very high frequency). Since these signals don't easily penetrate solid objects reception might be cut off if the path between satellite and receiver is blocked. The receiver needs at least three satellites to be able to perform a fix of the position. With a greater number positioning of course becomes more precise.

Another issue concerning the positioning is that the receivers, intended for navigational purposes, only deliver a position each second. When traveling at a high speed, say 110 km/h, this means that positioning occurs only once on each 30 m interval during optimal circumstances. When the user marks a position (P) by pressing the button the system can either use the position received before the mark (R1) or wait for the next one (R2). In doing so additional error in the position is introduced. If traveling at 110 km/h this error will be somewhere between 0-30 m granted that the time between the receptions are no more than 1 second. When reception is bad there is of course a chance that the intervals between receptions are longer and the error will become even larger.

In order to avoid additional errors caused by the positioning and to limit the effects of short reception failures PlaceMemo uses the average speed between the two receptions points (R1 and R2) along with both positions to interpolate position P (se image below).

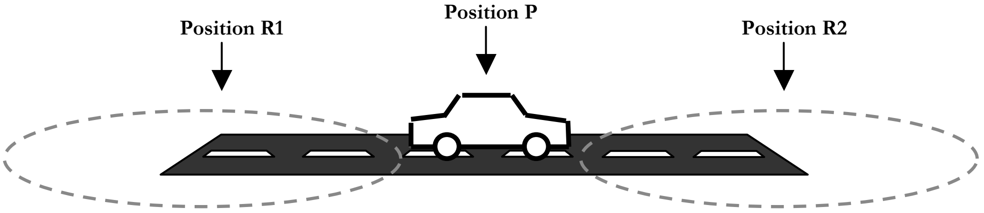


Figure 10. The user indicates that a PlaceMemo should be placed by pressing a button. This occurs when the vehicle is at position P in the figure. The points R1 and R2 indicates the vehicles position when positioning data was received. This data can point to any location within the error perimeter (orange marked area*).*

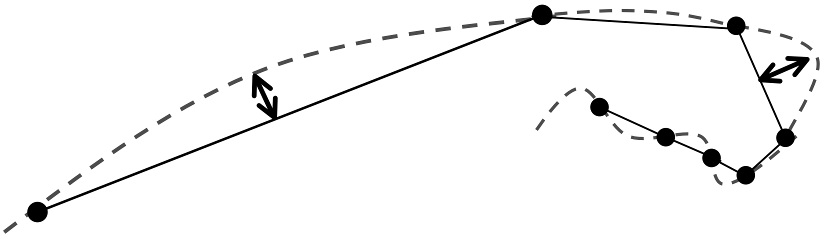


Figure 11. The path travelled is marked by the dotted line. Each point indicates a received position. Obviously the assumption that we travel in a straight path between reception points is not entirely true.

When doing so it is assumed that we travel from R1 to R2 in a straight path. This assumption is not true in every case, which is illustrated in figure 11. Using the heading of the vehicle at each point, a better approximation would perhaps be possible. But since a car traveling at 110 km/h is unlikely to make a rapid turn (law of momentum), it will deviate only partly from the straight path. If it later slows down and makes a more rapid turn the slow speed will make the distance between reception points so small that the deviation from the path again will be small.

It is possible that we, when performing positioning in order to place a memo, also will be forced to account for user behavior. When tested in a simulation with a movie of the traveled road projected before the user it became evident that the users often marked an item when it appeared just in front of the vehicle later causing the memos to be slightly delayed when arriving from the opposite direction. While this of course could be ascribed to a narrow camera lens causing the user to misjudge the vehicle's position it also implicates the possibility of an offset while placing memos under non-simulated circumstances. Higher velocities often cause drivers to gain so-called tunnel vision. Since tunnel vision in some aspects is very similar to a narrow camera vision we could expect the same behavior to occur while driving and furthermore that the offset in placing will be correlated to the current speed.

3.2.3 Recording the memo

At the same time as the position is secured, recording of the voice memo starts. Since storing audio is very expensive in terms of memory the voice memos have to be compressed before they can be saved. The compression coder/decoder used, is the platform-included Microsoft's GSM codec [15]. This codec, based on the same algorithm that is used for compressing speech in the cellular GSM network, utilizes common characteristics of speech to achieve an exceedingly high compression rate. Since the algorithm is very specialized it will not work well when used to compress sounds other than speech. Any background noise lacking the characteristics of speech will therefore affect the performance negatively or be filtered away. Data is compressed to about 2 kB/s.

3.2.4 Storing the information

The voice memo data is stored as a wave file [16]. Information about the voice memo such as length and filename is stored along with the positioning data in a database once the recording is completed. Originally the idea was to include the positioning data under an info tag in the wave file as well. In this way the file would become a complete PlaceMemo message by it self, still remaining properties of an regular sound file. Such a solution would have made it easy to distribute, play or backup PlaceMemo messages and to do so even without the use of the PlaceMemo application. Since this fell under part three of the project (distribution) it is not yet implemented.

3.3 Triggering a PlaceMemo

In accordance with the design criterias, a previously marked location once again is approached, will trigger the playback of the voice memo associated with the location. This playback is announced, by two loud beeps, to assure that the user's attention is gained. The playing is started before arriving at the marked location in such way that the memo will have finished playing, as the location is reached.

3.3.1 Determining an error perimeter

In order for the system to decide when a PlaceMemo should be triggered, it needs to compare the PlaceMemos in the database to the current location. When doing so, positioning errors in both our present and the previously marked location has to be accounted for. The GPS equipment will due to positioning errors never give us the exact same coordinate as we revisit a location. To determine when two positions are to be treated as concurring, an error perimeter or threshold value is used. If two positions are within a certain distance from each other we will treat them as equal. Setting the threshold value low will result in place memos being left out as they are passed, setting it high will on the other hand result in inaccurate triggering.

In this implementation a fixed error value is used. The total error perimeter (E_{tot}) will, when accounting for both previous and current errors, be twice this value. Since accuracy to some extent is related to the GPS equipment used the user can override default error value within the configuration file. A better more complex solution would involve individually approximated error perimeters for each location based on current PDOP value (the angle to the satellites upon reception, se GPS section) and the PDOP when the location was placed. This method would need the PDOP values to be stored together with the coordinates of the place memo. It would also cause more extensive calculations in order to determine the error perimeters. Since we are building a prototype we will settle with the simple solution at this time, but triggering of the memos can not be based on the error perimeter alone.

3.3.2 Determining the trigger perimeter

We want the playback of the place memo to act as a reminder. We also want it to provide information to the user in order for him to decide which measures to take. To base the triggering of a memo purely on the error perimeter would generate a solution that would begin playback the instant the location was passed. This is often too late since a user, traveling at high speed, will be have passed the location by several hundred meters before the message is completely received.

One approach would be to expand the error perimeter to cause triggering to occur earlier. This very static solution however does not appeal to our context aware thinking. At high speed for example the triggering perimeter might still be too small for the user to be able to take actions. At slow speeds it might be to large, resulting in the danger of losing the context of the message that the surrounding reality is supposed to provide.

In order to give the user time to act in all situations we have besides the error perimeter based the triggering algorithm on speed and message length as well. In the current implementation these two properties are used to approximate the distance traveled as the message is played. We use the instantaneous speed as an approximation since it's not possible to predict the mean speed during the message playback. By adding this distance to the total error perimeter we get a new perimeter (T_{perim}), which we refer to as the triggering perimeter. Given that the speed doesn't change during the playback this perimeter when used for triggering will cause the memo to finish just before the associated position is reached.

$$T_{perim} = E_{tot} + (\mathrm{MessageLength} * \mathrm{Velocity})$$

The trigger perimeter will as a consequence of the above formula be different for every place memo since the length of the voice message will differ. Furthermore it will also differ between each search since the speed is likely to have changed. The above formula is the same one used in the Backseatgameing prototype described in the article Movement and spatiality in a gaming situation – boosting mobile computer games with the highway experience by Liselott Brunnbergs and Oskar Juhlin [17].

At one point the idea was to also include the approximated distance the vehicle would need in order to stop, into the triggering perimeter. By doing so the message could be made to finish in time for the user to be able to make a safe stop. The distance needed for breaking is proportional to the kinetic energy of the car, which in turn is proportional to the square of the current velocity.

$$K.E = \frac{m \times v^2}{2}$$

The breaking distance, it follows, will be a square function of the velocity. As a consequence the triggering distance will increase very rapidly as we approach higher velocities. This in turn means that the probability of getting erroneous triggering of memos residing on parallel roads will increase since the triggering perimeter will sometimes become very large. To avoid this an idea about restricting the triggering of place memos to the direction of the vehicle, was developed. The triggering perimeter becomes a triggering area or triggering sector.

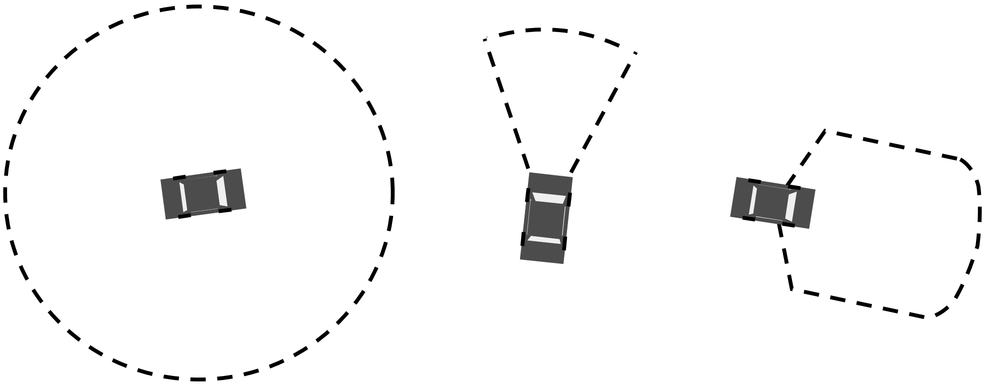


Figure 12. Triggering perimeter, triggering sector or triggering area. While using a triggering perimeter no considerations regarding the vehicle's direction needs to be made.

The idea of using pre-recorded samples to give the user a clue about the direction of the location marked could also prove useful in regard to this problem. The user could benefit from getting information about place memos in nearby areas given that additional information makes it apparent that the memo currently played doesn't reside on the current road.

With the uncertainty already in the calculations regarding the velocity, resulting from the use of the instantaneous speed, in addition to uncertainties about the user's behavior during triggering it is hard to tell if improving the triggering algorithm in this way will yield any benefits over the current solution. As mentioned earlier the length traveled while playing the message is approximated with the notion that the user keeps a constant speed. A likely response to a place memo being triggered however would be for the user to slow down in order to prepare for the stop. It's also likely that the user is able to make a decision of stopping before a message is completely received and as a result will have plenty of time to stop the vehicle. As a result it was decided that no further efforts were to be invested into this problem at this point until further investigations about the user's behavior in these situations are made.

When triggering a place memo the same difficulties regarding positioning exists as when placing a memo. There is however an important difference that prevents us from applying the same solutions to these problems. When triggering, positions can't be interpolated between reception points. This is obvious since, should we wait for reception R2 (in figure 13) to perform this interpolation we will already have passed the location. The problem could instead be solved by using a dead-reckoning algorithm as will be briefly discussed under section 3.3.6, however the current prototype simply adds one second to the voice message length in order to compensate for the random delays caused by the delay between receptions during optimal conditions with no positioning failures.

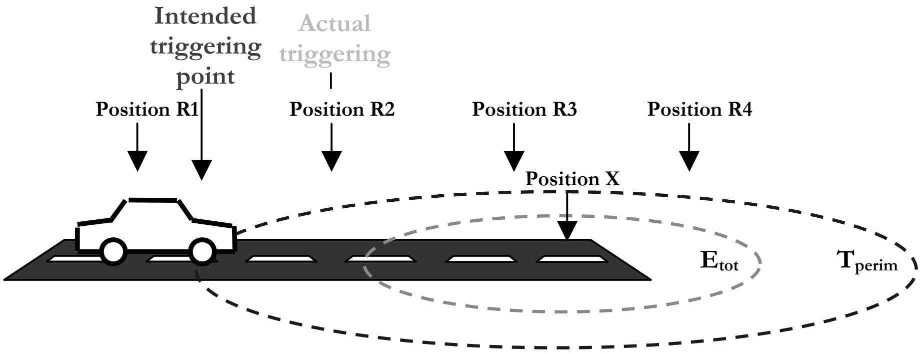


Figure 13. Since interpolation can't be done during triggering the memo will not start until the first reception within the triggering perimeter.

To avoid memos being trigged one and one in situations where the vehicle will be present inside the triggering perimeter of an object over a longer period the system also stores the time when the last triggering of a particular memo occurred. A default setting, configurable within the configuration file, then disables the memo from being played for the next 30 minutes.

3.3.3 Establishing a safe perimeter

Searching through the database each time our position or speed changes and to perform calculations on every position stored will consume an exceedingly large amount of system resources. To avoid this the application establishes a safe perimeter for every search. The safe perimeter (D in figure 14) is the distance from our current position to the nearest triggering perimeter of a place memo. In other words it is the longest distance we can move from the current point without any chance of triggering to occur. The triggering perimeter is, for all place memos dependent on the current speed so the safe perimeter consequently has to be defined as a function of the current speed.

An initial search is done in order to establish the safe perimeter. After that the perimeter can be recalculated using only the speed. The safe perimeter is continuously compared to the distance moved from the point where it was established. As long as we do not proceed outside the safe perimeter no additional database searches will be necessary.

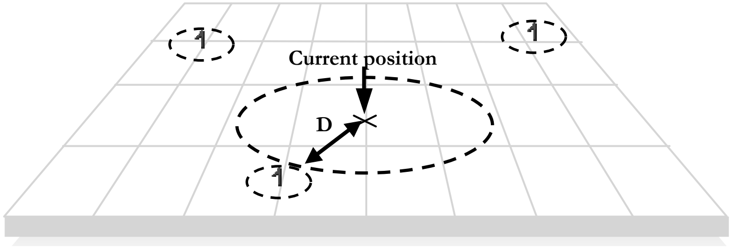
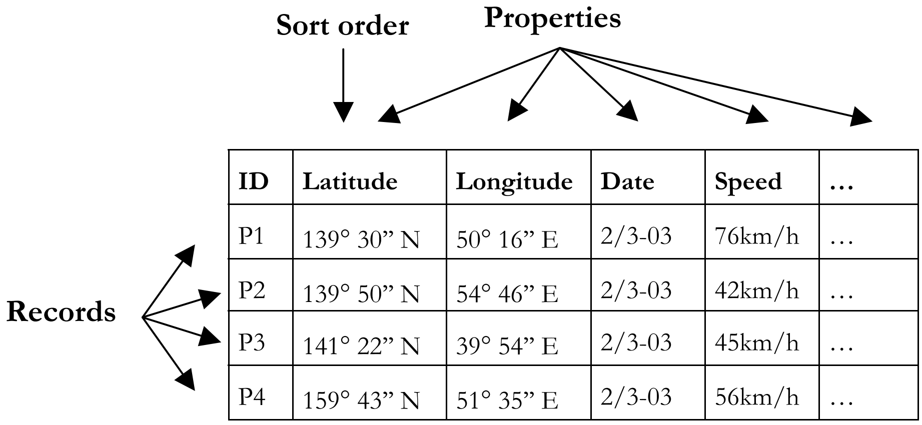


Figure 14. A safe perimeter is established as long as we reside inside this area, no additional database search is necessary as there is no chance for the place memos being trigged. The size of the safe area decreases as speed increases and vice versa, since its inversely dependent on the triggering distance of the

3.3.4 The database search algorithm

In order to further improve performance the database search algorithm needs to be designed to be as efficient as possible. The database in the Pocket PC OS API used in this prototype supports basic functionality such as sorting and searching of records with respect to a certain property. One difficulty that arises is that position, which is the search property used in this case, is a two dimensional property. This means that we have to use two properties in a record in order to describe it, one for longitude and one for latitude. Since the database functionality only supports one property to be searched at a time, we need to construct our own search algorithm.



ID	Latitude	Longitude	Date	Speed	…
P1	139° 30” N	50° 16” E	2/3-03	76km/h	…
P2	139° 50” N	54° 46” E	2/3-03	42km/h	…
P3	141° 22” N	39° 54” E	2/3-03	45km/h	…
P4	159° 43” N	51° 35” E	2/3-03	56km/h	…

Figure 15. The database can be described as a table where each row forms a record containing properties.

The database is created with the latitude as sort order. The reason why we choose the latitude and not the longitude is because longitudes converge towards the poles. A numerical distance latitude-wise, can be said to correspond to a fixed distance on the globe surface, with exception for some deviations emerging from the earth's ellipsoidal shape. For longitudes we need to consider the latitude on which the longitude distance resides before concluding anything about the distance in units of meters it corresponds to. Why this is important will become evident once the algorithm is described.

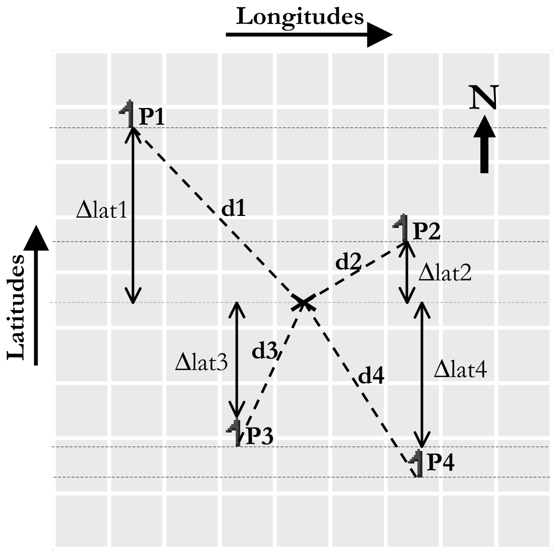


Figure 16. The database objects spatially represented. X is the current position.

In the picture above we can se the four records listed in the database table plotted into a 2-dimesional map. As our database search for the currently closest place memo begins, we use default functionality within the database to set our seek pointer onto the record with the largest latitude value smaller than our current latitude. Our current position is marked by the X in the figure. The seek pointer will in this example after the operation point to record P3 in figure. The algorithm continues to search the records with the closest latitude values on either side of record P3 calculating the distance to each coordinate. Once the distance latitude-wise exceeds the shortest distance found so far the search is completed.

In the above example P2 will be the closest position. Even if we imagine there to be many more records in the database containing latitude values larger or smaller than the ones in the example we are able to conclude that P2 is the closest and thereby end the search. This conclusion can be made since the latitude difference between the current position and P1 or P4, lat1 and lat4 alone exceeds the distance to P2 (d2).

The distance to the closest place memo is, after the search is finished, used to set the safe perimeter or decide if the memo is to be trigged as described in previous sections (3.3.2).

3.3.5 Extended functionality

As a test for investigating the possibilities to provide additional information, besides the information contained in the voice message itself, a feature to provide information about the age of the place memo encountered was included into the design. This function can be enabled by setting the appropriate flag in the configuration file. While enabled this function will use prerecorded samples in order to compile a sentence that announces the date of when the memo originally was placed. Each time a place memo is trigged it will automatically be follows by its date. The feature was implemented on the notion that a road inspector's dynamic environment will make the place memo's age a very important property in determining its current significance.

Other possible features, considered includes samples to indicate the direction of the marked location, indication of the direction traveled once the location was marked, the occasional annunciation of distance to enclosing place memos in order to provide the user with an overview of the current situation. In future versions of PlaceMemo the possible use of a speech synthesizer in order in to provide the user with additional information while driving seems likely.

3.3.6 How the solution meets the requirements

There are a number of cases in which we could suspect that placing and appropriate triggering of place memos with the current solution will fail. For instance when placing memos in road tunnels, on roads parallel to others or on top of each other, when place memos are placed very close or when a place memo is encountered while a new one is being recorded.

The failure to handle these special cases can all, be traced to one or several of the following three shortcomings:

- failing GPS-reception
- inaccurate positioning
- inadequate algorithm

If a place memo is encountered while another place memo already is playing or a new one is being recorded the encountered message is put on hold until the current operation finishes. While the memo still gets delivered it will often fail to be delivered in time. An idea for solving this issue would be to improve the algorithm so that the situation with multiple place memos in close proximity to each other could be detected and triggering scheduled to occur earlier in order for all memos to have time to play. When a memo is being recorded the operation could be temporarily paused in order for encountered messages to be played. This solution would of course require additional system messages to be played to inform the user of the temporarily suspended recording. Recording could be resumed shortly after the playback finishes.

When it comes to receptions failures the current implementation provide functionality for dealing with short failures during the placement phase. In the example in figure 17 the GPS receiver gets its last reception shortly before entering the tunnel. It regains the signal again once clear of the tunnel. Functionality for placing of a memo is disabled by the system after 10 seconds without GPS-signal. This means that after entering a tunnel or any other obstacle interfering with reception we still got the ability to place a memo for a period of time. During reception failures the system will, just as it does in normal cases, wait for the next reception and use is together with the previously known location to interpolate the current position. If the tunnel path doesn't follow a straight line or the vehicle changes its speed during the reception failure, the approximation as explained earlier will be less accurate.

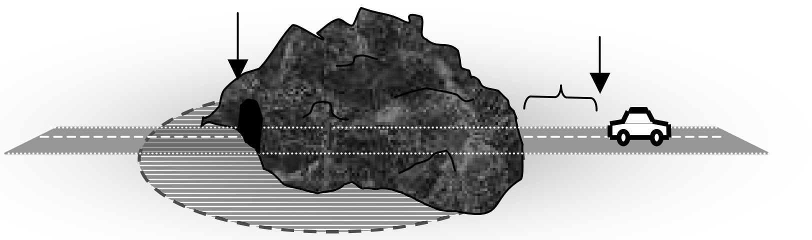


Figure 17. The figure shows a car passing through a road tunnel, placing a PlaceMemo inside the tunnel. The left arrow indicates where the GPS signal is lost, the right where it is regained. A delay for the GPS-receivers to lock on to the signal will normally be needed when recovering from a reception failure.

In order for a PlaceMemo inside an area without reception to be trigged the triggering perimeter has to stretch outside the area obscured from GPS reception. In the figure above (fig 17.) an approach from the opposite direction (from right to left) would fail to trigger the PlaceMemo (marked by the red flag). The triggering would instead occur once leaving the tunnel on the left side. An approach from the left on the other side would trigger the PlaceMemo correctly since the triggering perimeter stretches outside the tunnel opening.

The implemented algorithm is only intended to support short reception failures. Still it could during the right conditions support the placing of memos in more complex situations. To improve functionality of the application further, the use of a dead-reckoning algorithm is suggested. A dead-reckoning algorithm assumes that the vehicle continues with the same speed and distance until information saying otherwise is received. Using this algorithm to determine if a triggering should occur would in the example illustrated above cause correct triggering to occur even as we approach from the right.

A more explicit approach to the tunnel problem would be to store information about any unusually long reception failure together with the place memo placed (a somewhat comMotion inspired solution). Once the place once again is approached the system can be programmed to anticipate the signal loss. The system could, as this occurs, assume that we have entered the tunnel and approximate the time to when the triggering perimeter is reached given the speed at which we entered the tunnel. This algorithm would of course still fail in complex tunnel systems where the road divides and there are more than two exits or entrances.

Regardless of possible improvements to the current prototype the current implementation for triggering has proved to work exceedingly well. It has been subjected to a series of functionality tests performed both in the central parts of Stockholm as and in the more rural parts in the surrounding of Umeå. It should of course be noted that these tests has not been performed with actual users but rather people possessing basic knowledge about the capabilities of this prototype.

3.4 The graphical interface

In the ethnography, a need for users to be able to plan ahead was identified. An interface capable of satisfying this need was therefore requested. The user is believed to use this interface, to among other things determine what kind of equipment to bring on the inspection. The interface will consequently need to support users in finding and utilizing information stored in the system regarding certain area, without the use of the real world as context.

The interface, has besides planning, also been assigned functionality for tasks such as managing and distributing place memos. Managing in its current form basically consist of deleting old memos no longer in question. In an imagined future version it might however be expanded to also include tasks such as the translation of the informal place memos to more formal text based reports, possibilities to add additional memos to a location, appending or updating existing memos with new information and abilities to flag important place memos. Distributing of memos will occur as work is delegated.

The visual mode of interaction is anticipated to be used in all situations were the user, can afford to provide a higher level of interaction. This does obviously not include when the user is driving. The predicted situations for this mode of interaction to be used will be:

- At the station prior to an inspection, where it is believed be used for planning.
- During a break along the road, where the user will delegate work, check on incoming memos or possible re-plan the inspection.
- Once returning after the inspection. Where removal of memos settled during the inspection and delegation of work is likely to occur.

During testing of the prototype, it has become apparent that the visualization to some extent also could prove useful while driving in order to get a quick overview of approaching memos.

PlaceMemo's graphical interface became a very large component of the project. The reason for this is the very limited support for graphics on the Pocket PC platform, earlier mentioned under the platform section. Existing GUI-features[6] on Pocket PC/Windows CE are in many aspects unusable since they don't reflect the mobile nature of the device. Instead GapiDraw [18] a library intended primarily for game development had to be used, in order to create the visualization. Since this library provides us with only the most basic graphical functionality, a lot of effort had to be spent on implementing additional functionality needed[7].

3.4.1 Representing a PlaceMemo

The easiest and perhaps also most common way to visually represent a number of objects and its properties in computer graphics is in the form of a list. Windows CE

[6] Graphical user interface.

[7] The GapiDraw toolkit by Johan Sanneblad at the Viktoria Institute, is occasionally expanded to include additional functionality. We here refer to the state of this graphics library at the time of development of this prototype. http://www.gapidraw.com/

supports a number of GUI lists intended for purposes such as listing files. Listing memos unlike listing files however is difficult since there is no textual identification connected to a place memo. A place memo is unlike a file never named since entering textual input on the Pocket PC is not applicable while driving. Displaying a list of unnamed memos would make it very hard for the user to tell them apart. In order to be assured of having all the information needed a user would be forced to listen through each and every memo in the whole system. While speech might be fast for the author it is often slow and tedious for the listener [19]. The sequential properties of speech makes it difficult to scan through and search for information in the way you normally would do with printed text.

To minimize the number of memos that the user needs to access, we form a more characterized representation of a PlaceMemo by making use of known properties such as it's date, length and position. This will enable the user to discard some of the memos in certain situations. The user might for example be looking for information about an object he knew he encountered a few days ago he then uses the date as a reference and searches messages marked with a recent dates. A list with PlaceMemos marked with date could be sorted to make this procedure even easier. Since the road inspectors environment are very dynamic it is likely to assume that the date overall will be a very important property for estimating the current significance of a memo. The length of a memo might, in this sense, maybe not be as good, still it could provides us with a clue to the complexity and the amount of information in the message. The listing constitutes one part of the solution but to further enable browsing of the PlaceMemo information space additional aids has been provided.

As the ethnography explains inspectors travel in predetermined routes. The user planning an inspection will in other words prior to the tour know which areas he will be traveling through. It is reasonable to assume, that the place memos interesting in the road inspectors view prior to the inspection, will all originate from the planned inspection area. With this in mind it looks like if the best property for initially representing, sorting and identifying a memo would be its position or the region in which it lies. By selecting an area of interest the user should be able to access memos within a specific area. As the position is not represented particularly well in a list, simply displaying longitude and latitude values is not very informative, we will need to consider some form of multidimensional spatial representation that better can visualize the 2-dimensional properties of the position.

Common forms of spatially represented information are maps. When referring to the term map most of us first of all think of cartographic maps. But the term map is not well determined [20] and there are several other forms of visualisations other than cartographic maps that also can be defined as maps. For instance the connected nodes displaying how to travel by underground train from one point to another. A map could even involve descriptions of non-spatial entities, like a mind map. In this case however we will for certain use some form of spatial map since we wishes to visualize the location of the place memos. Having a map displaying the spatial relation of the PlaceMemos on the touch screen would enable us to construct a system were the user easily could encircle the areas of interest.

The question is then how this spatial representation should be designed considering both the user's situation as well as limitations in the hardware. It might not be feasible and within the scope of this master thesis to include a complete cartographic map database in the solution. The small screen and limited recourses of the mobile device furthermore makes is difficult to display detailed cartographic information [21]. In a map more like the underground scheme, distances could be abstracted away to create more compact representation, better suited for the small screen of a Pocket PC. A schematic map like this will with the absence of details, also cause place memos appear more distinguishable.

Considering the ethnography once again it can be assumed, that the road inspectors are familiar with cartographic roadmaps, since they are a part of their equipment. An idea would be to make the spatial representation similar in form to those that already exist. The users would then have a much easier task in recognizing and understanding the system. It would also be easier to use existing maps in conjunction with the new system, to mentally transfer information from the cartographic map onto the system and vice versa. By displaying the cartographic information directly on the PlaceMemo system the existing cartographic maps could perhaps even be abolished. A further argument for providing a detailed map would be that no real world context will be present during planning. Since the placememos are created in situ, they might not make much sense outside their context. A detailed roadmap could in this regard, provide the clues for regaining the context to a certain extent (it could for example be seen that a place memo is located in a road junction).

3.4.2 The virtual information space

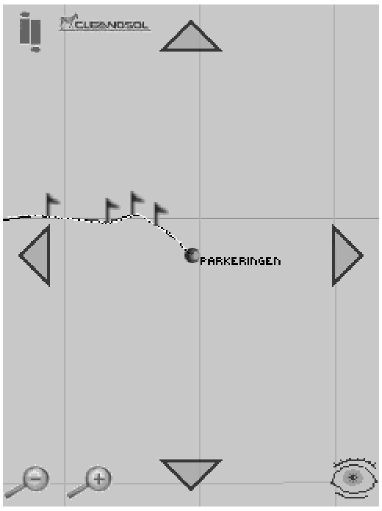


Figure 18. The graphical map interface.

Nevertheless visualization in PlaceMemo in its current form lacks any form of cartographic information other than the information put there by the user. The decision to not provide cartographic data was done since the memory was better needed for storing voice memos. Implementing the functionality needed to display this information would in addition make the task of completing the prototype during given time frame difficult.

The idea instead became to construct a meaningful information space, using information stored in the system in addition with a grid net of longitude and latitude lines acting as a reference for the ground. This reference is important since panning and zooming will not produce any visual feedback unless there is an object in the view. Views without objects will, without the grid net, appear as a blank. The grid will contribute to the illusion of movement and also give us clues about current scale. When viewing the earth from distance the grid net will outline the spherical shape of the earth and cause it appear as a 3-dimensional object.

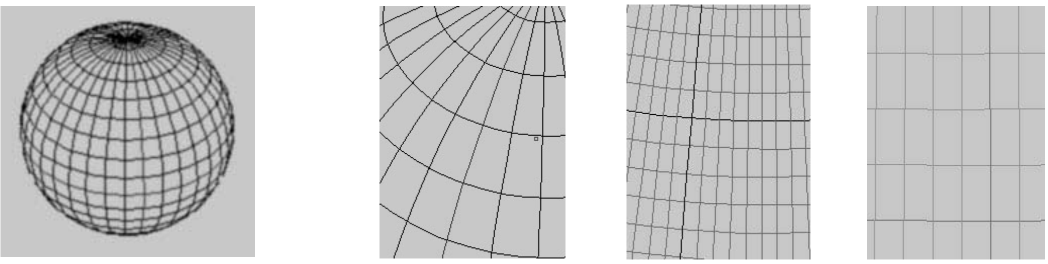


Figure 19. The grid net in on four different zoom levels.

A place memo in this 3-dimensional visualization is depicted as flags, colored in accordance with its age. To further facilitate the interpretation of the map, we utilize the path traveled during placement of the PlaceMemos. This path provides us with information about the order in which the memos was laid. It also shows to us the shape or the trail of the roads used, information that can be used to fit the image onto a roadmap.

At an interview with the intended test persons, where the map design was discussed, the need for further aids to locate the view was envisaged. As a result support for reference points were added to the implementation. A reference point is a location marked with a dot and a text string. These points are meant to be entered by the user just as PlaceMemos. In opposition to a PlaceMemo however a reference point is regarded as static information. A reference point can be added during an inspection tour but is named at a later occasion when not driving.

3.4.3 Geographical information in computers

The geographical information when used in computer graphics for creating maps can be stored as an image of a previously done projection in the same way a plotted map stores the information on a paper. The map projection, which later will be described, is the task of transferring the 3-dimentinal surface of the earth onto the 2-dimensional surface of a paper or computer screen. While saving us the work of calculating the projection, this method still allow us to pan and zoom our map image to some extent. We will however be constrained to the same level of detail regardless of zoom level. The method will also prevent us from changing the map projection.

Another way to store the geographical information is as points and vectors in a map database. Information stored in such way could be visualized using alternating projection methods as the projection is done the instant the map image is to be displayed. Zooming becomes less restricted since vector graphics remains sharp regardless of zoom level. Map databases of this kind can furthermore be implemented in layers. Each layer will then contain a different level of detail enabling us to produce a map view, with the appropriate amount of information in relation to the current zoom level or situation. While displaying this kind of map each coordinate and vector has to be projected from coordinates defined within a certain geodetic datum, into coordinates used to plot the vector onto the 2-dimensional screen.

In PlaceMemo each memo is stored with its position. Consequently we will be using the second approach and perform the projection once the memo is to be displayed. The same applies to the traveled paths, reference points, and grid net displayed. The databases used to store these objects, currently only supports one layer of detail. The implementation of more layers would of course help creating a better overview when a large area is contained inside the view. This will need to be implemented as an automated ongoing process since information in PlaceMemo is constantly being added and removed. The more abstracted layer could for example be created by adding place memos close enough to be represented by one single point. The point could then be sized and displayed with a number, in accordance with the number of place memos it represents. As place memos are removed the number is decreased. As the counter becomes zero a point would be deleted.

3.4.4 Geodetic datum's and map projections

The geodetic datum, it can be said, defines the relationship between a point on the globe and its coordinate. In 1687 Newton suggested [22] that the earth due to the centripetal force must be flattened by the poles and not spherical as believed until then [23]. The shape of the earth have since then been a subject for much debate.

Today we know that the earth possess a highly irregular shape. To completely model the true shape of the earth becomes very complex so instead we often apply approximated models of the earth's surface, suiting our current needs for precision. A system consisting of one such approximate earth model together with rules defining reference planes and the origin of a coordinate system is referred to as a terrestrial reference system or a geodetic datum [24].

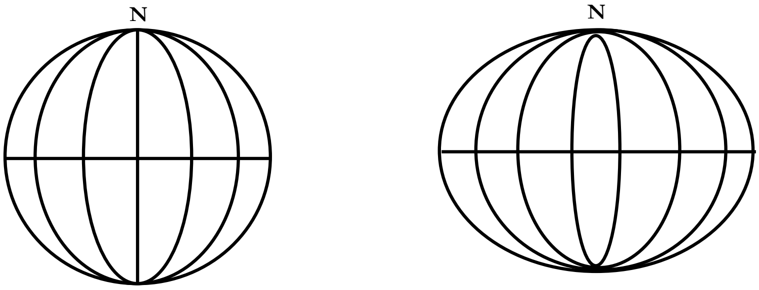


Figure 20. Some geodetic datum's uses a spherical model of the earth (left), more commonly however an ellipsoidal (right) model is used.

A place on the globe can obviously have several different coordinate values depending on which geodetic datum currently being used. Using the wrong datum without converting coordinates can of course result in positioning errors. While some datums specialize in mapping a certain area (country or region) others have been constructed to provide an adequate accuracy over the entire globe. One such datum is the WGS84 (World Geodetic System 1984) currently used by the GPS-system.

Most datums including the WGS84 approximate the earth as an ellipsoid. There are however, still spherical model in use. Spherical models results in simpler calculations when, for example, calculating the distance between two points. In return we have to tolerate less accuracy, especially when two points are far apart. Over short distances, the degrade in precision is considerably less. Consequently the spherical model is best used when performing fast calculations on small distances where high precision is not crucial. For really small distances, under 10km, we could get acceptable results even by using a flat earth model.

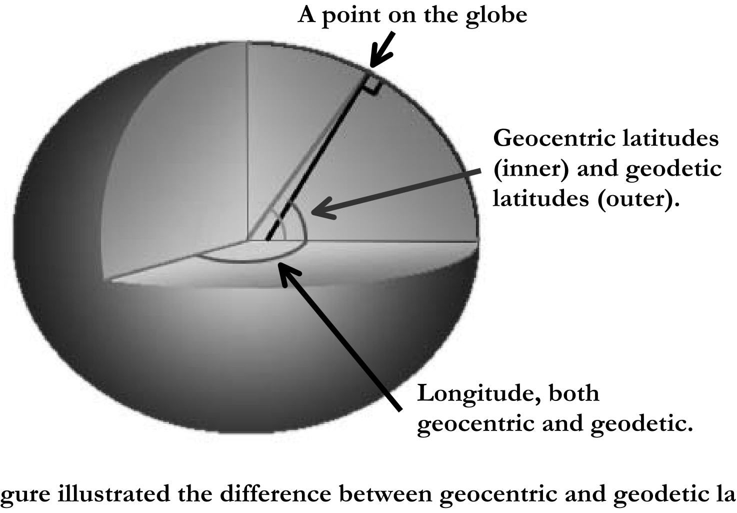

A point on the globe
Geocentric latitudes (inner) and geodetic latitudes (outer).
Longitude, both geocentric and geodetic.

Figure 21. The figure illustrated the difference between geocentric and geodetic latitudes.

The most common way to describe a point on the globe is through the geodetic coordinate system. There are however, even though rarely used, other types of coordinate systems such as the geocentric system. In a spherical model the geodetic and geocentric system are the same. In both systems coordinates are given in two angles named longitude and latitude. The longitude of a point is the angle between a reference plane and a plane passing through the point, both planes being perpendicular to the equatorial plane. The latitude is for the geocentric system, the smallest angle between the equatorial plane and a line drawn from the point to the centre of the earth. The geodetic latitude on the other hand is the smallest angle between the equatorial plane and a line drawn through the point perpendicular to the ellipsoid. This is best understood by viewing the figure above. As you can see this causes the geocentric and geodetic latitudes to deviate in non-spherical earth models.

In order to create a 2-dimensional map of the 3-dimensional globe, cartographers use map projections. There are an infinite number of map projections so we will just mention a few. The three most common types of projections are planar, cylindrical and conic projections [25]. The picture below gives you a notion of how these projections work.

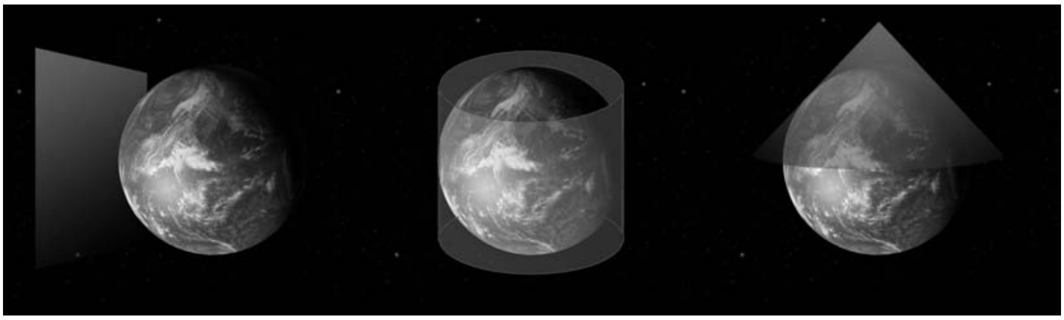


Figure 22. Here three common types if map projections are illustrated. To the left planar projection, in the middle cylindrical and to the right conic projections.

The 2-dimensional surface that the image will be projected upon can be thought of as a sheet being wrapped around the globe or positioned next to it in order to better understand the projections. As all projections distort the globe in some way there is never a best projection suitable for all situations. While some preserve properties such as direction or area others maintain the continental contours or shapes. Projections can be designed to distort the globe less over a certain area or along a certain longitude or latitude. The choice or design of projection ultimately comes down to an issue of determining which properties to favor.

The most natural map projections, i.e. the projections most similar to the projection that would occur on our retina while viewing the earth from a great distance, are so called perspective or orthographic planar projections.

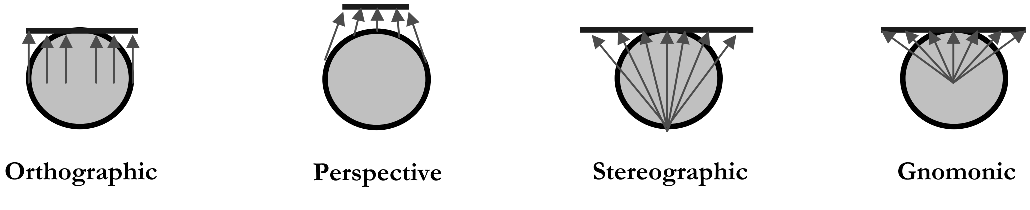


Figure 23. Four types of azimuthally planar projections.

While producing an image that is perceived as 3-dimensional, these projections has the ability to create a very realistic and undistorted image of the earth in the mind of the viewer (the image of the globe in figure 22 is an example of perspective planar projection). On printed cartographic maps perspective or orthographic projections are uncommon due to the difficulty to display a coherent image of the whole world using these methods. The images produced by these projections will only show one side of the earth. Furthermore the true distances and the distances on the projection area does not correspond well.

In computer graphics or in geographical information systems (GIS) the ability to display a coherent image of the whole world or a region is possible even though we use a perspective or an orthographic projection. This is possible since the user can be given the ability to change the view. If the user needs to determine a distance between two points this can be done automatically by the computer rather than having the user measuring the distance on the screen.

3.4.5 The visualization

In the PlaceMemo prototype the projection used is planar perspective. This means that the picture of earth is depicted onto a plane using a visual point (se figure 24). This projection can also be thought of as being the picture projected onto the photographic film in a camera. The distance between the plane and the visual point (the eye in the picture) corresponds to the angle of the lens or objective used in the camera. By making the distance small we could gain a fish eye perspective, the kind of view you experience while looking through the peephole of a door. By making the distance large the effect would be that of looking through a zoom objective. If the viewpoint to plane distance gets large enough the projection will ultimately become an orthographic projection. Although the camera's position, direction or lens angle can be adjusted freely in this solution the user will, as we will discover later, in most cases be restricted to an azimuthal view of the earth.

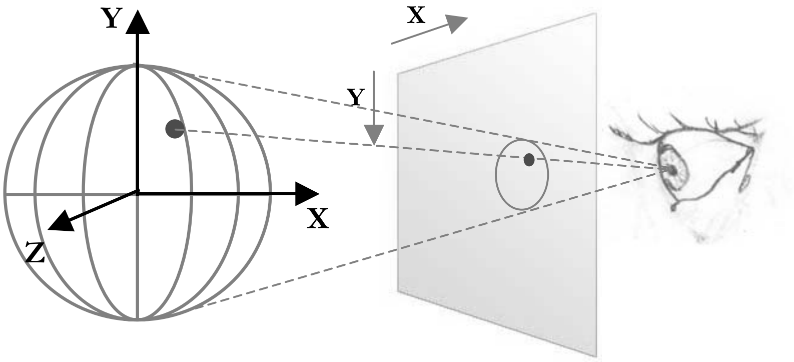


Figure 24. Imagine the screen as a plane between the users eye and the virtual 3-dimetional model. The indicated 3-dimensional point projected upon the screen by calculating the intersection point of the screen-plane and the line connecting the point and the point where we imagine users eye to be.

The projection algorithm is done in the following way. First of all geodetic coordinates are transformed into an earth centered, earth fixed Cartesian coordinate system. The axes of this coordinate system are marked X,Y and Z in the figure above.

The transformation formula used is the following [24].

$$X = (N + h) \times \cos\phi \times \cos\lambda$$

$$Y = (N + h) \times \cos\phi \times \sin\lambda$$

$$Z = (N\frac{b^2}{a^2} + h) \times \sin\phi$$

λ = Longitude

ϕ = Geodetic latitude

a = semi-major axis of earth model used

b = semi-minor axis of earth model used

N = Radius of curvature in prime vertical

h = height above ellipsoid

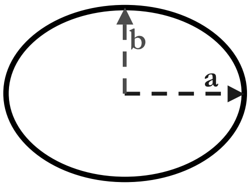


Figure 25. Semi-major and semi-minor axis.

The N is the radius of the ellipse at the point we are transforming. This value is dependent on the current latitude and is given by the formula:

$$N = \frac{a}{\sqrt{1 - e^2 \times \sin^2\phi}}$$

where,

$$e^2 = 2\left(\frac{a-b}{a}\right) - \left(\frac{a-b}{a}\right)^2$$

The coordinates received from the GPS-receiver will be using the WSG-84 datum. This datum uses an ellipsoidal earth model. In this visualization however a spherical approximation of the earth will be used. This simplifies calculations considerable, while only having a marginal effect of the resulting visualization.

Using this simplification, formulas 4 and 5 can be disregarded since the radius of a sphere per definition is constant regardless of latitude. We also ignore the height of objects which means that h=0 are used for all cases. The formulas 1,2 and 3 can as a result be transformed into:

$$X = r \times \cos\phi \times \cos\lambda$$

$$Y = r \times \cos\phi \times \sin\lambda$$

$$Z = r \times \frac{b^2}{a^2} \sin\phi$$

where the scalar *r* is the mean earth radius. The next step in the projection involves transforming the coordinates from the earth-fixed earth-centered coordinate system to a camera-fixed (or projection plane fixed) coordinate system.

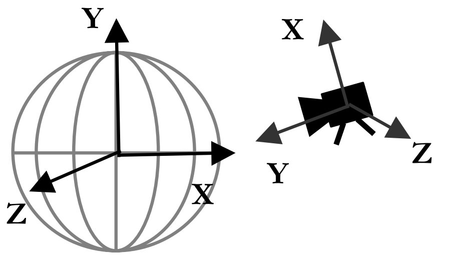


Figure 26. The earth centred earth fixed coordinate system and the camera centred camera fixed coordinate system.

To understand how this is done we need to repeat some basic linear algebra. The camera's position is like any point defined by its coordinates [x,y,z]. Its direction is in addition defined by two additional points: one indicating the direction in which it is pointed and a second to indicate what is up in the view. The earth center is found at [0,0,0] in an earth-centered coordinate system. In order to translate a point from an earth-centered coordinate system to a camera-centered, we subtract the camera position from the coordinate to be transformed. This is illustrated in a 2-dimensional example below.

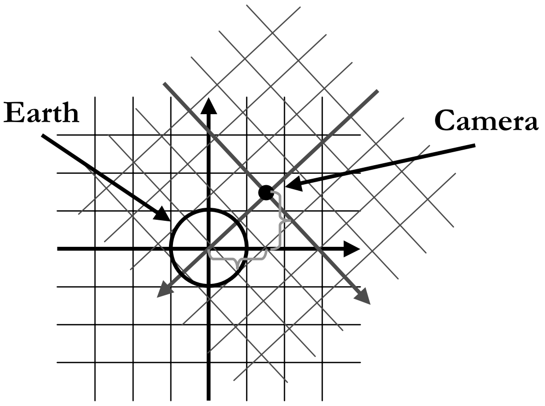


Figure 27. A 2-dimensional illustration of the two coordinate systems.

The camera coordinate system however should also be rotated according to the cameras direction. In order to translate a point from the non-rotated system to the rotated one we need to find the rotated system's unit vectors defined within the non-rotated system. A unit vector is a vector of length 1 originating from the center of the coordinate system parallel to one of the coordinate axes. Defined within itself the unit vectors of a 3-dimensional Cartesian space would be,

$$\begin{bmatrix}1\\0\\0\end{bmatrix},\begin{bmatrix}0\\1\\0\end{bmatrix},\begin{bmatrix}0\\0\\1\end{bmatrix}$$

The rotated coordinate system will have its y-axis along the camera's line of sight. This means that by normalizing the vector defining the direction at which the camera is pointed we will have the first unit vector out of three.

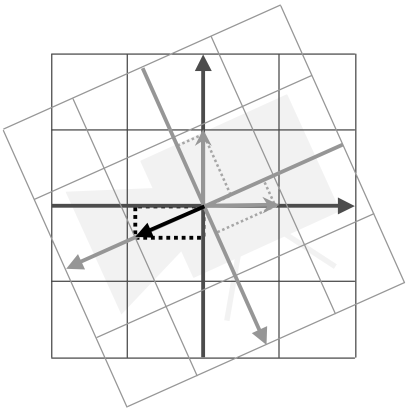


Figure 28. The two different coordinate systems. In order convert a point from one to the other, the unit vectors for one coordinate system needs to be defined within the other. To find the first unit vector the camera-to-vector is normalized.

The remaining two unit vectors should be orthogonal to the first one. The vector defining the up direction of the view might when normalized serve as a second basis vector. The problem is that this vector might not necessarily be orthogonal to the first basis vector (se figure 29). It does however lie within the plane defined by the unit vectors u1 and u2. When having two vectors defining a plane in a 3-dimensional space the cross product can be used to find a vector orthogonal to the plane. From the cross product of u1 and the Up vector we get the u3 vector. Again using the cross product on u1 and u3 we get the last unit vector u2.

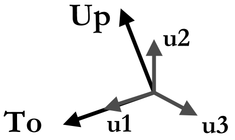


Figure 29. We want to find the unit vectors (u1,u2,u3) of the camera coordinate system by using the camera Up and To vectors. The first vector (u1) is found by normalizing the To vector. The Up vector however can not be used directly to find the u2 vector since it is not necessarily orthogonal to u1 but since we know that u1 and the Up vector defines the same plane as u1 and u2 we can find u3 by the use of the cross product.

Having the three unit vectors defining the direction of the axis in the camera coordinate system we can determine the coordinate of a point within the camera coordinate system. This is done through calculating the projection of the vector OP onto the axis.

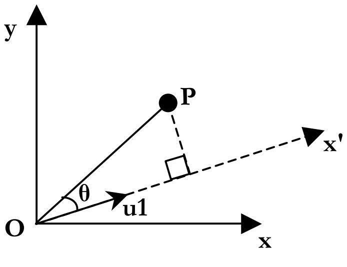


Figure 30. The point P is projected onto the unit vector u1, in order to get the x value defined within the camera coordinate system.

This projection is given by,

$$proj_{x'}\overrightarrow{OP} = \left\|\overrightarrow{OP}\right\|\cos\theta$$

but with the help of the theorem,

$$a \bullet b = \|a\|\|b\|\cos\theta$$

we can use the dot product instead of cosine. The length of the unit vector is always one (|u1| = 1) so we will then have,

$$proj_{x'}\overrightarrow{OP} = \left|\overrightarrow{u1}\right|\left\|\overrightarrow{OP}\right|\cos\theta = \overrightarrow{u1} \bullet \overrightarrow{OP}$$

The formula for translating a point P(x,y,z) from the earth-centered earth-fixed coordinate system to the camera-centered camera-fixed coordinate system, P'(x',y',z'), will be.

$$x' = (x - x_f)\, x_{BasA} + (y - y_f)\, y_{BasA} + (z- z_f)\, z_{BasA}$$
$$y' = (x - x_f)\, x_{BasB} + (y - y_f)\, y_{BasB} + (z- z_f)\, z_{BasB}$$
$$z' = (x - x_f)\, x_{BasC} + (y - y_f)\, y_{BasC} + (z- z_f)\, z_{BasC}$$

The last step is to project the 3-dimensional model onto the 2-dimensional plane. This is done by dividing the gradient of the line connecting the eye and a certain point, with the gradient of the line between the eye and the edge of the screen. This is done horizontally respectively vertical.

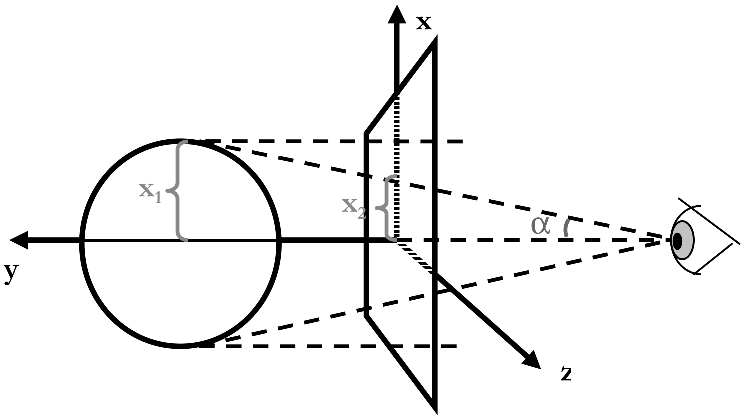


Figure 31. The image describes how to find the screen coordinate from a 3-dimensional point in a camera-fixed camera-orientated coordinate system.

3.4.6 Storing and retrieving paths

The paths, PlaceMemos and reference points are stored in databases. A critical task for the application is to, from these databases retrieve the information to be displayed in the current view. Reference points and PlaceMemos are points defined on the globe. When searching the database these points can easily be checked in order to determine if they are inside or outside our current view. Paths on the other hand are segments stretching over a large area. Theses segments are defined by several points connected by straight lines. It is not enough to simply determine if a segment point is inside or outside the view. A segment could be visible, even though none of the points that defines it are. Before explaining how this is dealt with we will first learn how the paths initially are recorded.

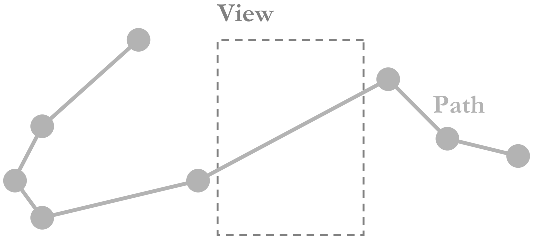


Figure 32. Even though the path is visible none of its segment points are.

While out inspecting roads the GPS will deliver one position every second during normal conditions. To store each and every position, in order to visualize the path traveled, would be resource consuming to say the least. One single inspection tour, in average 7 hours, might result in as many as 25200 positions (7*60*60). The need for a compression algorithm becomes obvious. The algorithm constructed here compresses the path in two steps. Fist of all it filters out positions not describing movement before saving the position to a buffer, it then compresses the buffer before saving it to the database.

Filtering out non-movement is a simple task. Positions are plainly ignored as long as the distance between the position and the previously stored position is smaller than a certain threshold value (user defined in the PlaceMemo configuration file). In cases where the distance is larger than the threshold value the position is stored in the buffer. The buffer is of fixed size thus limiting the length of the path that can be buffered to about 10 – 20 km (the size of the buffer * the threshold value), we will call this the buffer distance. As the buffer is filled, it will overlap and start writing over the oldest positions. The buffer will as a result always contain the last 10 – 20 km traveled or the most recent path of one buffer distance.

If a PlaceMemo is placed, the buffer is compressed and the result added the path database. This will enable us to later view the path 10 – 20 km traveled before a place memo. The buffer is after the compression cleared and a flag is set to enable logging of the path following the PlaceMemo. When later another PlaceMemo is placed, the buffer is once again completely filled or the application is shut down the buffer is compressed a second time and appended to the previously stored path in the path database. In this way the path of one buffer distance before and one buffer distance after a PlaceMemo will always be stored. As long as the distance to the next place memo is less than a buffer distances (10 – 20 km) the path will be stored continuously. A counter associated to each path will indicate the number of place memos on that specific path, by decreasing this counter each time a place memo on the path is deleted the system will know when a path no longer is needed.

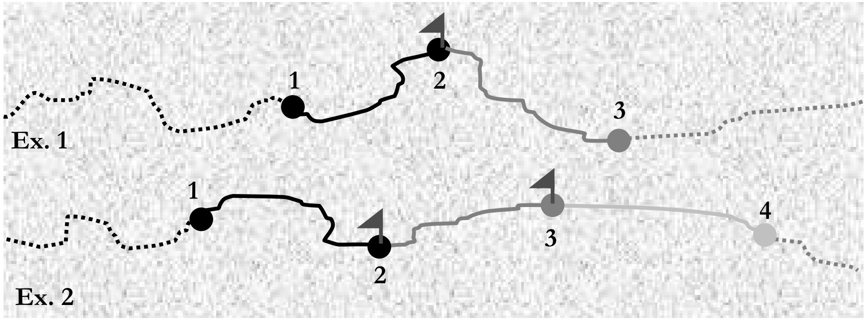


Figure 33. In both examples we travel along the marked path. Once we reach position 2 a place memo is placed. At this point the path between point 1 and 2 is in side the path buffer this path is compressed and stored.

In example 1 the buffer will be filled again once we reach position 3. This path is then compressed and appended to the previously stored path from 1 to 2. The resulting path stored once a place memo is placed will be from position 1 until 3.

In example 2 another place memo is placed before the buffer is filled causing the path between position 2 and 3 to be compressed and appended to the previously stored path (1 to 2). Once the buffer becomes filled at position 4 the path from 3 to 4 will also be appended to the two previous paths. Logging will cease at this point. The resulting path stored will have a counter value of two, indicating that there are to place memos currently connected to it.

Compression of the paths is done through a segmentation algorithm [26]. This algorithm starts out with only the first and the last points of the path. These points are connected with a straight line. It then calculates the distance from the line to all other points. The point furthest from the line is then added splitting the path into two line segments. The procedure is repeated for each segment until no point is further from the path than a certain threshold value (also defined by the user in the configuration file).

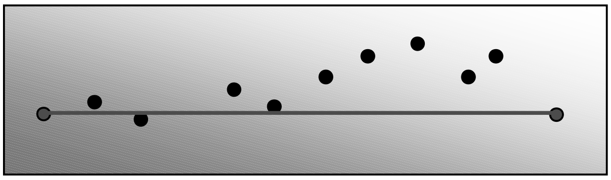


Figure 34 a. The algorithm starts by connection the first and last point in the segment.

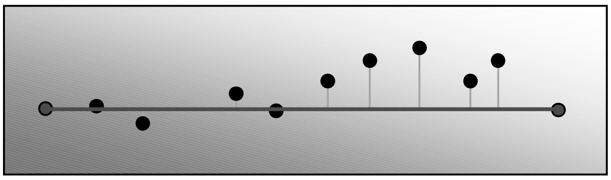


Figure 34 b. It the calculates the distance between each point and line.

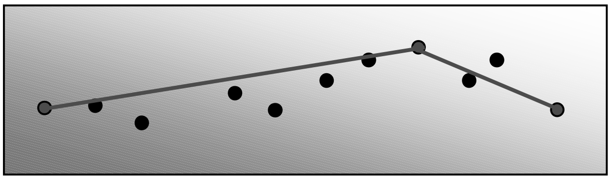


Figure 34 c. The point furthest from the line is then added to the segment, given that the distance is larger than a certain threshold value.

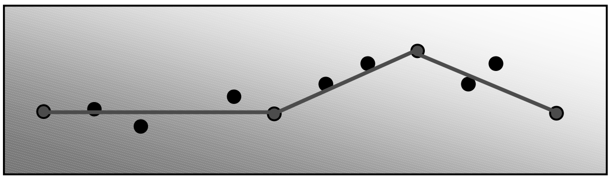


Figure 34 d. The previous operation are repeated for each line segment until there are no more points with further from the segment than the threshold value.

As anticipated this algorithm works particularly well on straight paths. But as we start making turns, larger amounts of memory will be consumed for storing the path.

A number of alternative methods for compressing the path have also been investigated, among these different kinds line-fitting methods used in regression analysis. Using this reduced major axis regression a line could for example be fitted so that, the sum of the distances from line to points is minimized. This method would in opposition to the above method remove all extreme values. It was originally thought that this kind of line fitting could be used to eliminate noise caused by GPS errors in order to compress the path, but since it is often extremes that contain the most valuable information such as where to find road crossings etc. the method was later rejected.

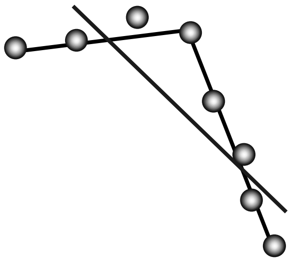


Figure 35. The points represent GPS data retrieved during a 90˚ turn. While major axis regression will remove the turn. The currently used algorithm will preserve extreme points making the turn more visible.

Once the path is stored in the database the problem becomes, as mentioned before, to find the paths currently in our view. To solve this problem we use bounding boxes.

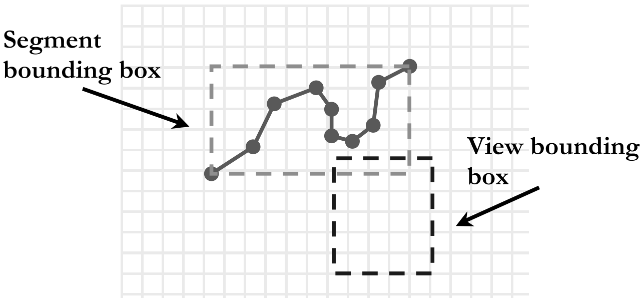


Figure 36. The method of using bounding boxes sometimes produces false hits when searching for segments inside the current view.

Each path is stored together with information about its bounding box. A bounding box is defined as the smallest box that includes the entire path. This box is described by a south, north, east and west boundary. By defining the bounding box for our current view the task becomes to determine which bounding boxes that intersect the view. Those that are, will be retrieved, projected and clipped using the Cohen-

Sutherland Line-Clipping algorithm [27]. In some instances the bounding boxes intersect even though the segment is not inside the view (se fig. 36). This method does however assure us of not missing any of the segments.

3.4.7 Geodetic bounding boxes

When defining a geodetic bounding box using geodetic coordinates (latitude and longitude) rather than Cartesian coordinates (x and y) a number of issues arises. Since distance between longitude lines decreases as we approach the poles our bounding box will in some cases be shaped less like a box as the name implies and more like a triangle or a slice of pie. The spherical form of the earth also causes problems. The same longitude line can be referred to using several different values. For example – 10°, 350° and -370° all refer to the longitude 10° west. If we define two longitude intervals in the form [westward boundary, eastward boundary] how could we easiest determine if the two intervals intersect? First of all we will construct a function determining the length of the interval. Since westward longitudes are written as negative values we can get the interval by subtracting the westward limit from the eastward. This works for all intervals except those that include the longitude leap from 180° to -180°.

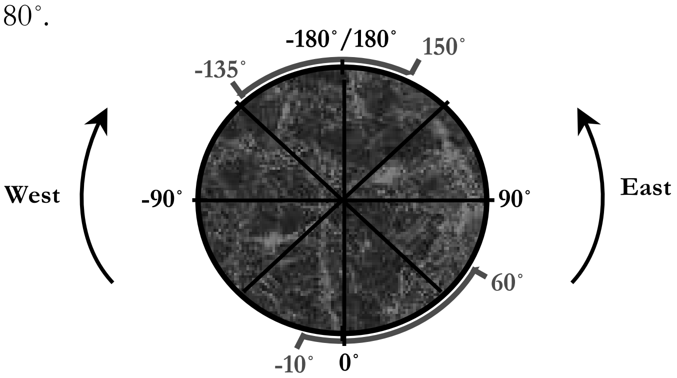


Figure 37. When calculating the length of a longitude interval we will have to consider that the interval might contain the leap from 180° to -180°.

In this case the output will become negative ((-135°) – 150° = -285°). By adding one whole lap (360°) we will get the correct interval (-285°) + 360° = 75°. So by subtracting the westward limit from the eastward and adding one revolution if the result is negative we get the length of the interval.

To determine if the intervals [b,a] and [c,d] intersect we summarize the lengths of the intervals [a,b], [b,d], [c,d] and [c,a]. This sum adds up to exactly one lap (360°) if the and only if the intervals do not intersect [28].

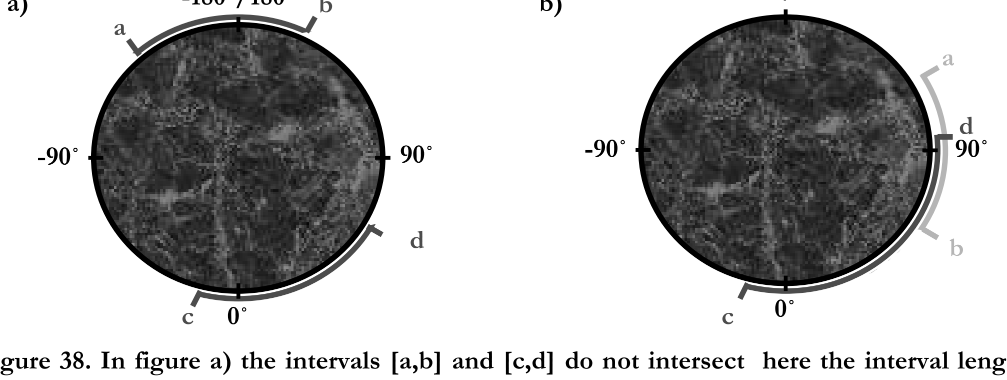


Figure 38. In figure a) the intervals [a,b] and [c,d] do not intersect here the interval lengths [a,b] [b,d] [d,c] and [c,a] add up to exactly one lap. Adding the same intervals in figure b) were the intervals [a,b] and [c,d] do intersect will not result in one lap since the interval [b,d] will be counted three times.

Intersection in the latitude direction is checked in a similar way. A problem with latitudes is that we sometimes need to define bounding boxes overlapping the poles. As we move across the pole the northward limit will start to approach south. To deal with this problem the longitude interval is expanded to encircle the pole.

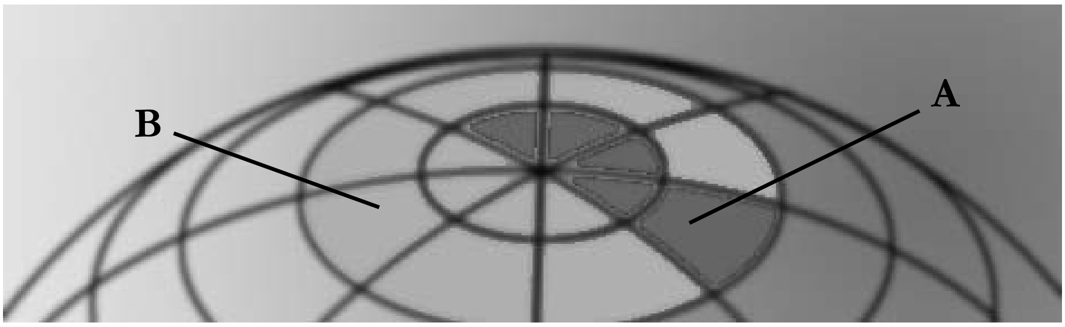


Figure 39. A path running through areas A is hard to include in a bounding box described with a north, south, east and west boundary. Often the whole area around the globe has to be included (area B). This can results in lowered performance since we will have an increasing amount of false hits during database searches.

3.4.8 Buffered zooming and panning of the information space

As the user pans the view, the information has to be updated. Since searching the databases and comparing bounding boxes takes time, pan movement will become erratic if the view is to be updated continuously. An alternative would be to load new information in intervals causing some areas in the view to momentarily become empty as the view is moved. To get around these problems a solution using a buffer was designed. The buffer enables the view information to be updated continuously even though database search operations are done in intervals. We will in this section describe how this buffer is constructed.

The visualization is done in two steps. In the first step we retrieve the information currently relevant to us from the database. This information is then loaded into a memory structure and initial calculations are made such as transforming the geodetic coordinates into earth-fixed, earth-centred Cartesian coordinates. In the second step the information available in the memory is transformed into a 2-dimensional projection and plotted onto the screen. The 2-dimensional coordinates from the projection are stored since they can be reused as long as the view is not moved.

The buffer can be thought of as enlarging the area loaded during the first step of the visualization. In doing so the view can be moved within this area without the need to perform search operations. Search operations can instead be scheduled to occur as the view approaches the boundary of the buffer. In the picture below the outer area represents the information space present in the databases. The blue-marked area is the area currently loaded into the memory buffer. The green area is the part projected and plotted onto the screen. The buffer size if defined in terms of the size of our view. Currently the buffer size set to is three times the size of the view.

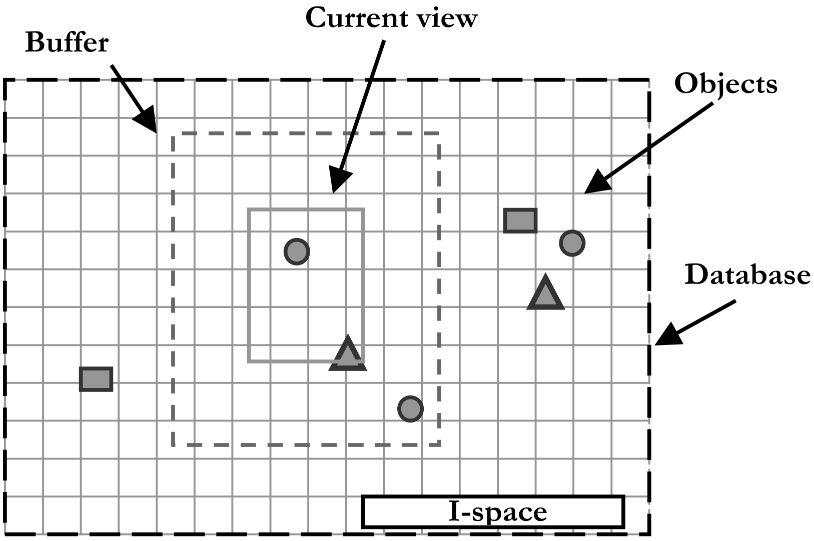


Figure 40. This figure illustrated the buffering of the view. Objects both in the view and its surrounding area are loaded into memory buffer.

To determine when it is time to renew the buffer a view displacement value is defined. The displacement is the percentage left of the buffer i.e. the distance before the view will reach the edge of the buffer compared to what this distance would be when the view is centred and the default view buffer size ratio exists. This value will obviously be 100% initially. As the view is panned or zoomed out the value will decrease.

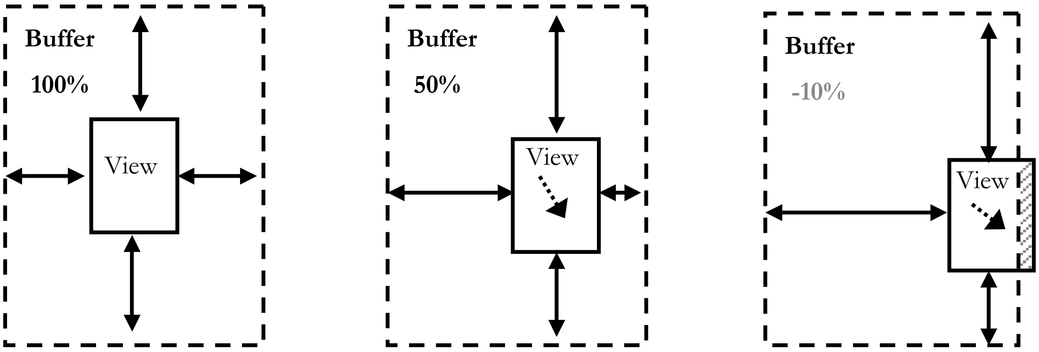


Figure 41. An illustration of three different buffer states. In the first example the view is centred in the buffer and the ratio (which changes when we zoom in and out) is default. In the second example the view is panned towards southeast with causes the buffer displacement value to decrease. In the last example the view has panned outside the view resulting in blank areas (marked with red).

Once the displacement value passes zero and becomes negative the view might includes empty areas outside the buffer not yet updated with information. Since this is not desirable updating of the buffer has to be triggered in time for the update to finish before the view reaches the buffer edge. The buffer update is done in a separate processor thread with lower priority than the tread responsible for drawing the view and thus do not inhibit view changes.

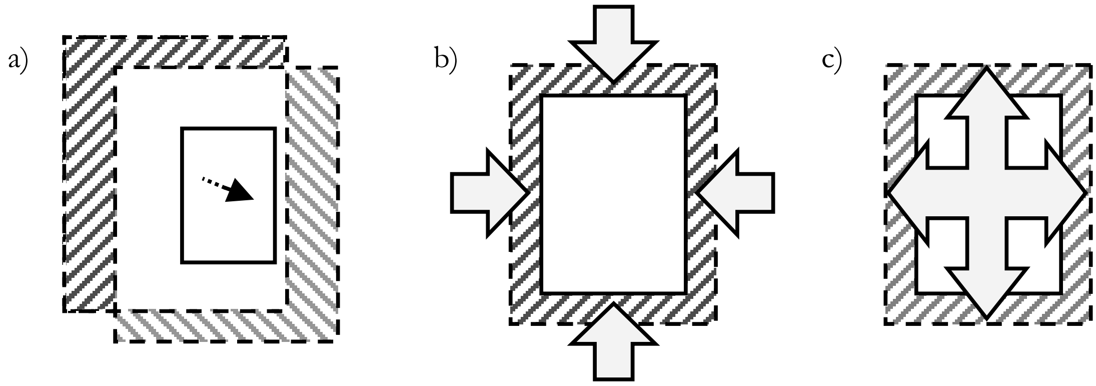


Figure 42. When the panning causes buffer displacement to become large enough, the buffer is moved. This is accomplished by loading the grey area and discarding the red area. The white area is left unchanged. As zooming occurs the buffer area is decreased to maintain buffer view ratio. As the view is zoomed out the opposite occurs.

The update is trigged once the buffer displacement passes a threshold value. Information is then added to the buffer in the gray area in the picture above. The threshold value is set to that the operation will be able finish before the area comes into view. As a buffer under run occur, often as the result of low system resources, an area with exceptionally high information density or very rapid or long pan operations, a message indicating an incomplete view is displayed. Subsequent to the buffer update old information (red area in the figure) is removed from the buffer. The view buffer is implemented as a thread secured so called linked-list enabling the thread process drawing the view to operate on it even as a buffer update is in progress.

3.5 Interacting with the I-space

So far we have we have been concerned with visualizing the collected information. The system created supports arbitrary positioning of a camera in order to create appropriate views of information within a 3-dimensional space. What now remains is to construct means for the user to navigate and interact with objects within this virtual environment.

3.5.1 Manual navigating

For moving the view in order to navigate through the information space both the touch screen and buttons could be used. But since the touch screen is better needed for selecting objects, navigation will be restricted to the use of buttons. Even though position and direction of the camera can be defined freely, the user will when navigating be restricted to view the earth surface straight from above. The view will also be rotated so that the top of the display always will point towards north.

By imposing these restrictions similarities between this visualization and the conventional maps used by the road inspectors are maintained. The task of navigating also becomes simplified as it is reduced to simply panning and zooming in opposition to within the 3-dimensional space both define our position, view direction and rotation. Navigating will consequently need a total of six buttons, four for each pan-direction and two for zooming.

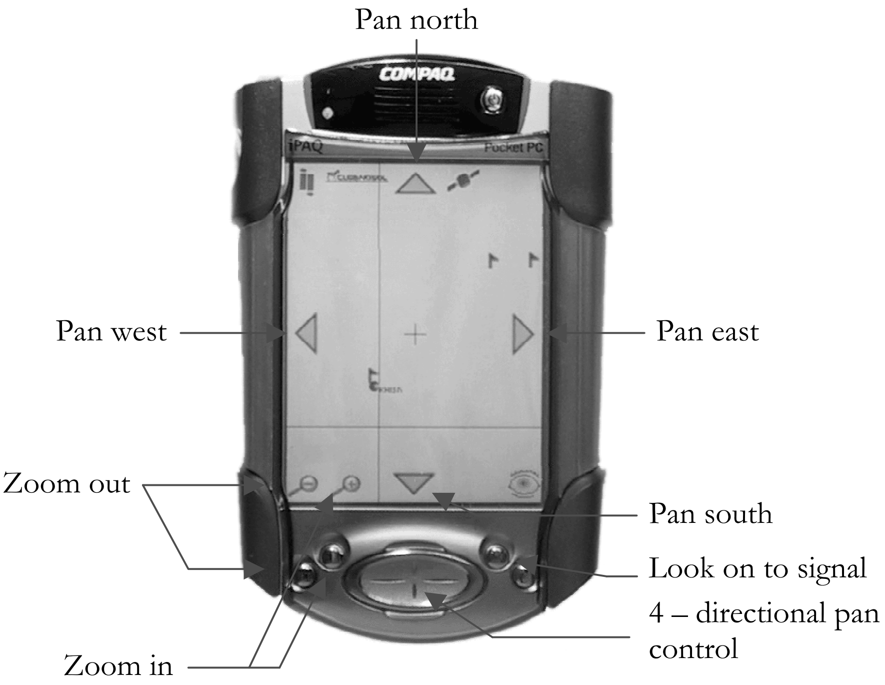


Figure 43. Description of PlaceMemo's interface for moving the view.

In order to simplify interaction while selecting objects with the stylus, navigational buttons will besides using hardware buttons also be defined as logical buttons on the touch screen. In this way both selecting and navigating can be perform with the stylus.

The restraint in the users ability to set the view will furthermore also reduce the risk of the user becoming disorientated. Since the information space is enormous compared to the small areas populated by information, losing oneself within the virtual environment is an eminent problem. Another problem also related to the dimension of the space and the distribution if the information is how to control the speed of our movement so that we quickly can move from one point to another regardless of the distance. A factor further complicating this problem is that the grid net intended as a reference for the user to track the movements of the view, will as the view is moved fast enough cause interference with the graphical frame rate. This occasionally causes the grid to appear to be moving in the wrong direction (the same optical or cognitive phenomena that you could experience by observing the spokes of a fast spinning wheel).

In the solution currently provided, the ability to control the speed in the view movement is achieved by linking it to the view altitude. The speed of the camera movement will while in close range of the ground be slow and on high altitudes fast. To the user however, a point will appear to move with the same speed over the screen regardless of view altitude since the pan speed is set proportional to the current scale. The zoom speed is in a similar way also set according to the altitude. On a smaller scale the speed of the movement is controlled by letting the view accelerate, as long as one of the pan buttons are held down. The user will with this solution be able to quickly relocate the view regardless of distance. While moving over large distances he will adjust the zoom level before panning. Even thou panning over small distances could be preformed without zooming, it has nevertheless proved rather convenient to use also use this function during these situations. As soon as the objects we want to locate are outside the view they are simply easier found by first zooming out.

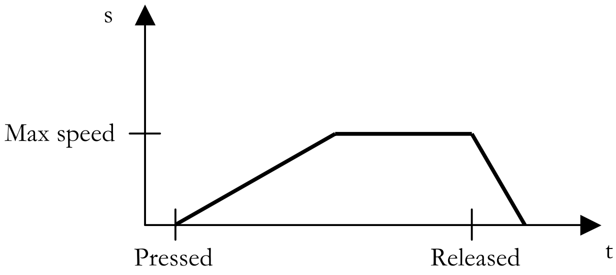


Figure 44. The graph shows pan speed as one of the pan buttons are pressed and released.

When using the pan buttons the view will start accelerate as soon as a button is pressed. The acceleration will however stop once a certain max speed is reached. When releasing the button the view will decelerate shortly before coming to rest. This enables us to control the speed by pushing the pan buttons in intervals to sort of shove the view forward. The behavior of the pan buttons also imposes a feeling of mass in the model.

3.5.2 Automatic and intelligent navigating

The methods for navigating the information space described so far have a major disadvantage. After using it for a while it feels like navigating consists of an endless amount of panning and zooming, in particular when zooming in on a particular object. While moving towards the surface an objects will often disappear from the view unless we temporarily stop and re-positions it, a procedure that often needs to be repeated several times before reaching the desirable altitude. In order to reduce the amount of actions to be preformed by the user when setting the view a number of algorithms has been added to provide a certain support. These algorithms will help the user in automation tasks such as setting the initial view, finding and locking on to the GPS signal and zooming on selected objects.

Locking on to the current GPS position is done by pressing one of the Pocket PC:s hardware keys (the key is indicated in figure 43). This function is primarily intended to be used while driving. It will as mentioned allow the user to receive and overview of approaching memos. As the user presses this button the view will pan towards the current GPS position. Panning in this case will not be done azimuthally (i.e by looking straight down on to the ground) as in regular cases. Instead the camera will first be directed towards the position before being moved. While moving, it will all the time be directed towards the new position. The method can be compared to how you could normally act while looking on a map by first redirecting your eyes and then reposition your head. Moving the view in this way gives you a much better idea of the heading and magnitude of the replacement since the view becomes less limited.

When the position is reached the view will keep reposition itself for every new incoming GPS position. The view is in this case always set to pan towards the new point in a speed set in relation to the distance to the point. As the point approached the screen center movement will decrease. This was implemented in order to keep the view from jumping around in correspondence to the incoming GPS signal. Using the pan buttons while looked on to the signal will be experienced as having a rubber band tied between the GPS points and the view. As soon as the pan button is released the view will return to its position over the signal indicator. Would the user however, insist on panning until the GPS signal disappeared entirely from the screen, the view would stop tracking. Another way to cancel GPS signal tracking would be to press the hardware button once more in order to exit from the tracking mode.

The second situation, in which automated navigation is present, is when the program is switched on. As the PlaceMemo application is started the entire world is shown in the initial view. The view will however, shortly after start to pan towards the last view setting used before ending the program. It is assumed that the user will want to continue were he last left off. If this is not the case the panning can be aborted at any time by the use of anyone of pan or zoom keys. By panning to the location instead of just setting the view on start the user will be given the opportunity to again make himself familiar with the surroundings.

To assist in zooming a special kind of function has been developed referred to as fast-zoom. This feature is mapped on the same buttons as the regular zoom. While a regular zoom is preformed by pressing and holding a button down, a fast-zoom is done by pushing and instantly releasing the same button. Using this feature, will case the view to make quick jumps between a number of predefined altitudes. The fast-zoom-in function will in addition also center on any objects within view. In case some of the objects have been selected the function will center only on those. As a result no re-positioning of the view will be necessary in order to keep an object within sight.

3.5.3 Selecting objects

Selecting objects is preformed with the stylus by simply encircling the objects of interest. The selected area will as you drag the stylus over the screen be marked in pink. Memos within this area will start to flash, to indicate the selection. To deselect the place memos simply tap on an empty area. To enable use of the entire screen area, the transparent buttons overlaying the map image, are constructed in such way that they will only be activated while tapping directly on them. As a result selecting can be preformed even under the button as long as we don't initiate the encircling directly on a button.

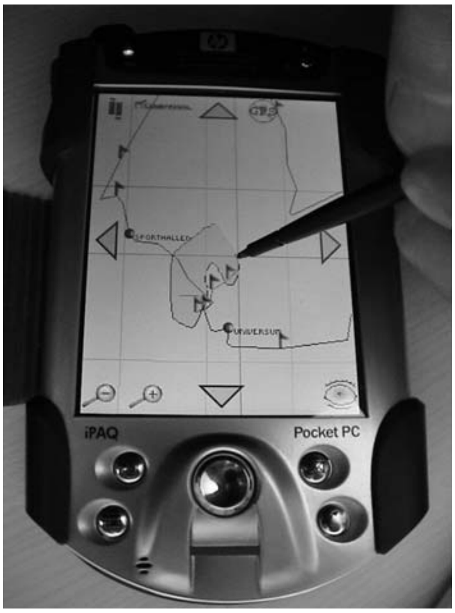


Figure 45. Selecting objects.

Selected object can later be viewed in a browser list (accessed by the browser button, se figure 43.) This browser list implemented by the user of windows tree view list functions currently provides support for playing placememos, renaming reference points and deleting both kind of objects.

4 Conclusions and future work

The intention with this master thesis has been, to create a prototype functional enough to be tested on actual users. This has been done with the purpose to enable evaluation of the ideas behind it. The resulting solution has the advantage of being both simple and general in its form. It contains no predefined maps or data of any kind and can as a result be used anywhere at any location on the globe. Its simplicity will facilitate the administration of the evaluation, as no configuration or information concerning the test user and his environment needs to be updated. It will hopefully also make the underlying concept become more apparent during the evaluation.

4.1 Evaluating the PlaceMemo concept

The evaluation of the system, preformed in its intended environment (infrastructure management) is complicated by the fact that it will become an evaluation involving tree different levels. We will at the same time be evaluating,

- The prototype
- The underlying concept
- And the ethnographic field work behind it

None of these three entities can be evaluated without the other. In order to properly relate the results from this evaluation to these levels certain consideration has to be made during the interpretation.

When evaluating the concept it is important that the prototype itself is made as transparent as possible and that the user will have enough time to familiarize himself with the device in order to stop seeing it as a toy[8]. A user without any background knowledge of the system he is set to evaluate, will otherwise be in danger of focusing too much on the prototype as such. In doing so the chance is that the user will not be able to see the prototype on a conceptual level. By a transparent prototype we mean a prototype that is so well designed that it will not be attract the users attention. It is often said that the best designs in HCI is the one that you notice the least.

Since the concept in itself, is the tool for evaluating the ethnography we will have a chain of dependences linking prototype, concept and ethnographic work. A conclusion will have to be, that even though we are to evaluate the concept of the prototype we cannot focus solely on the conceptual parts. As long as the prototype itself caused the system to fail in a certain situation we will be unable to say anything

[8] In PlaceMemo's case there is one further reason for this and that is that the system will not become interesting until the user has entered such amount of information into the system that an the occasional triggering will start occur while driving. Also PlaceMemo is a tool and in resemblance to other tools it will need a certain amount of practice before it becomes efficient.

about the success of the concept. When analyzing an event in the results we should therefore always start by analyzing it on a prototype level before applying the conceptual level. Before making any conclusions regarding the concept we need to eliminate the implementation as the source of the event. In cases where this cannot be done the prototype might have to be rebuilt.

4.2 Future work

There are a number of things to be included as future work on this prototype. Some of these things involve improvement in the current prototype, others extended functionality within a future version of the system. Had there been more time to improve the current prototype efforts would have been spent on securing system against data loss.

The loss of data is critical even in this prototype since it will effect an evaluation dramatically. A user losing several days of work will be unlikely to ever again use the system. There are several events that might cause us to lose all data saved within PlaceMemo. One is when the device runs out of battery. The Pocket PC contains no hard drive. Instead files are stored within a battery backed up RAM memory. Should the device be left long enough the danger is that all files will become lost. This problem is easily avoided. A more serious problem involves the occasional freeze up of the operating system sometimes demanding a full reset of the device. that will also clear the memory. To avoid this problem the program can be moved to the safer but smaller EEPROM memory, which also contains the Pocket PC 2002 operation system. Doing so will save us the work of downloading the executables every time the system failed. We would also be able to keep files such as the voice memos recorded. The database on the other hand seams to be lost even though it is relocated onto the safer media. The reason for this is that the database file created is just a link. The true location of the database is, it seams, determined by the operating system.

In order to improve the application against the loss of data we would either need to change the type of database used, an attractive option considering its many times suspect behavior, or store copies of the position data within the voice memo files as proposed in section 3.2.4. With the last proposal the database could be restored after each blackout.

Another thing in need of improvement in the current prototype would be the browser list. The browser list is currently implemented using Windows tree-view-list, which is somewhat unsuited for the purpose. By using the same technique as in the map view with semi transparent buttons the screen area could be utilized in a better fashion. The list could also be made easier to maneuver and to do so without using the stylus. The fingertip friendly touch screen keyboard designed for naming reference points would also be equipped with the ability to include numbers in when naming the reference points.

In a future version of PlaceMemo we can imagine expanded functionality such as the ability to append memos with new information, having place memos containing several voice memos, flagging important memos and so on. To remedy the problem of absent context while planning we could for example imagine a solution involving a digital camera providing an image of the location along with the voice memo (place picture?).

The effects of adding a road map will of course also be something to investigate further. During the development of this prototype some experiments with an overlaying map image in the form of a bitmap has been done to briefly survey the problem. The immediate impressions from those tests indicate that the increased amount of information on the screen will make it very hard to perceive the place memos in the map.

Another non-trivial problem to be solved in future version is the visualization. Since the system allows the users to provide the information to be visualized the system will become in great need of algorithms capable of displaying the information in an appropriate manner. We can as an example take the text strings denoting each reference point. In order to avoid strings to overlap causing them to become unreadable the system will need to have the ability to reposition the text in accordance to the situation. Visualizing non-predefined information is hard since we know less about its appearance.

4.3 Distribution

Most important, in order to complete the PlaceMemo concept however, is the adding of distribution. Place memos are as earlier mentioned intended to be used for delegating work. This part has been left out entirely at this stage due to time constraints. The next step will consequently be to extend the system with the ability for users to share their place memos. But before this is done it is my opinion that the current proposal for how this should be done needs to be revised.

The sharing of place memos is described as something that will occur from one user to another, meaning from one device to another on a system level. The ethnographic fieldwork suggests that this should be accomplished by the use of email, integrating PlaceMemo with an email client and sending place memos as attached files. While out on the field Internet connectivity is to be provided through the use of cellular phones. The intention with the proposal of using email for delivering place memos can be understood as an attempt to utilize existing network infrastructure. The proposal would solve the problem of finding a way for the devices to locate each other: a regular email address would be used for finding and identifying other devices.

The solution will however also mean that a server-to-client architecture would be used to implement communication that is fundamentally peer-to-peer. The point of having Internet connectivity out in the field is to be able to deliver shared place memos at once. While using email, all deliveries will be delayed until the receiving part connects himself to the mail server. This causes us to question the need for Internet while inspecting. The use of the email infrastructure would in my opinion only be sensible if the user were to lack the ability to connect during inspections and instead were to deliver place memos by connecting the device once back at the office. All place memos posted out in the field would then reside in the user's outbox until connected. Since the user in most cases will not have the opportunity to assign the place memos to be sent until after the inspection, this approach could very well prove sufficient.

If we on the other hand accept that there is a need for instantaneous delivery of place memos, we will instead need to question the use of both email and Internet. By establishing a direct connection between the devices information could be delivered at once. This connection could be established over the Internet. However since this means that we would need to pay for the transferred data twice, one time from sender to ISP and another from ISP to receiver, a better solution would be to let the cell phones connect directly without the use of internet. Except being the most inexpensive option this solution also provides us with an easy way to identify and find other devices by simply using each terminal's phone number. The system will in addition become less vulnerable for intrusion. In an interview with Cleanosol regarding the PlaceMemo it became clear that the computer security is a highly regarded issue due to the danger of industrial espionage.

4.4 Final conclusion

Although there are considerable improvements to be made, my opinion is that the goal of a prototype functional enough for testing on the actual users has been achieved. There is of course no way to know this for certain until further evaluation has been made. At the time of writing initial testing of the prototype has begun by employees at Cleanosol. The prototype will later also be brought back to the original setting at the Swedish national road administration. The only thing that can be said so far, based on our own somewhat subjective experience of the prototype is that it so far has surpassed our expectations.

In order to fully understand the PlaceMemo concept it needs to be experienced. The simple but powerful solution of appending voice memos to a position could through its usability almost be experienced as a sixth sense. It is with great expectations that we look forward to what the future will have in store for the PlaceMemo project.

5 Bug report

5.1 Critical bugs

Location: This bug is believed to reside within the path-compressor-algorithm.

Effect: Causes the system to shutdown immediately. All data within the database buffer not yet transferred to disc will be lost, typically the place memo being created at the time.

Description: Memory error, occurring within the core.dll file. The debugger is enable to provide a call stack during this error, which makes it hard to located its cause.

Location: Windows CE built in database.

Effect: The database becomes unusable. A command to connect to it will simply fail. ALL DATA WILL BE LOST!

Description: This error often occurs in relation to other system failures. We have learned that other developers also have experienced this error. To repair the system the database file has to be removed in order for the system to be able to create a new. The error is although extremely rare.

Location: Hardware error.

Effect: ALL DATA WILL BE LOST! The device will need to be sent in for service.

Description: HP has in their latest model forgotten to activate the EEPROM write protection for memory containing the operation system. This can cause any memory error within any program to overwrite parts of the operation system. The effects are diverse to say the least but ultimately the device will come to a state where it stops responding. The OS needs to be re-programmed into the device and consequently it need to be sent in for repair as HP currently does not provide the tools for doing this by yourself.

5.2 Non-critical

Location: Navigation, message handler.

Effect: The navigation will not function or no graphic will be displayed on start-up.

Description: When the device is used in combination with a Bluetooth GPS. The operating system put up a dialog window to enable the user to select the device to connect to. This happens as the program is stated. Occasionally this window will cause the application window to not attain focus properly, which causes the image to appear blank. It has also been noticed that the navigation sometimes can start to behave strangely. The problem requires a tap on the screen or restart of the application.

6 Acknowledgements

I would like to thank all the people at Interactive institute's mobility studio that has made my master thesis project such an interesting and enjoyable time. My gratitude in particular goes to Mattias Esbjörnsson and Mattias Östergren, my two mentors which have been a great support.

I would furthermore also like to tank my supervisor at the institution of computer science, Umeå University, prof. Lars-Erik Janlert for his comments and views on the master thesis report.

Last of all I would want to tank all my friends and family who has not only been a great support but also provided me with a place to stay during my time in Stockholm and further more also allowed me to find inspiration and motivation for continued work.

Tank you all!

Reference

[1] **Mattias Esbjörnsson and Oskar Juhlin.** *PlaceMemo - Supporting Mobile Articulation in a Vast Working Area Through Position Based Information.* In proceedings of ECIS'02.

http://www.tii.se/mobility/Files/placememo_020315_final.pdf (2003-05-21)

[2] **Mattias Esbjörnsson & Liselott Brunnberg.** *PlaceMemo: Using GPS and Mobile Computers to Augment the Roads.* Presented at Nordic Interactive, 2001

http://www.tii.se/mobility/Files/demoproposal_placememo_010517.pdf (2003-05-21)

[3] **Christina Wasson.** *Ethnography in the Field of Design.* In Human Organization, Vol. 59, No. 4, 2000.

[4] **John A. Hughes, David Randall and Dan Shapiro.** *Faltering from ethnography to design.* In proceedings of CSCW November 1992.

[5] **Andrew F. Monk and G. Nigel Gilbert.** *Perspectives on HCI, diverse approaches.* London : Academic Press, 1995

[6] **Yvonne Rogers and Victoria Bellotti.** *Grounding blue-sky research: how can ethnography help?* ACM Press 1997, pages 58 – 63.

[7] **Andy Crabtree, David M. Nichols, Jon O'Brien, Mark Rouncefield and Michael B. Twidale.** *Ethnometodologically informed ethnography and information system design.* Journal of the American Society for Information Science Volume 51, issue 7 (May 2000).

[8] **Anind K. Dey and Gregory D. Abowd**. *Towards a better unserstanding of context and context-awareness.*

[9] **Albrecht Achmidt, Michael Beigl, Hans W. Gellersen.** *There is more to context than location.* Computer & Graphics 1999 pages 893-901.

[10] **Gregory D. Abowd, Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper and Mike Pinkerton.** *Cyberguide: A mobile context-aware tour guide.* Wireless Networks 3 1997 pages 421–433.

[11] **Espinoza, F., Persson, P., Sandin, A., Nyström, H., Cacciatore. E. & Bylund, M.** *GeoNotes: Social and Navigational Aspects of Location-Based Information Systems.* In Abowd, Brumitt & Shafer (eds.) Ubicomp 2001: Ubiquitous Computing, International Conference Atlanta, Georgia, September 30 - October 2, Berlin: Springer, pages 2-17.

[12] **Natalia Marmasse and Chris Schmandt.** *Location-aware information delivery with comMotion.* Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing (HUC), Bristol, UK, September 25-27, 2000, pages 157 - 171.

[13**] Peter H. Dana.** *Global Positioning System Overview.* Lecture notes. Department of Geography, University of Texas at Austin, 1994.

http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html (2003-05-26)

[14] **Multiple internet recourses.**

http://www.trimble.com/gps/how.html (2003-05-26)

http://www.cmtinc.com/gpsbook/chap1.html (2003-05-26)

[15] **Nigel Thompson.** *Using CODECs to Compress Wave Audio.* Windows Multimedia Technical Articles, MSDN April 4, 1997

http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/dnarmulmed/html/msdn_codec.asp

[16] **Microsoft**. *Multimedia standards update - New Multimedia Data Types and Data Techniques.* Microsoft Corporation April 15, 1994 Revision: 3.0

[17] **Liselott Brunnbergs and Oskar Juhlin.** *Movement and spatiality in a gaming situation – boosting mobile computer games with the highway experience*

http://www.interactiveinstitute.se/mobility/Files/BSGFinal.pdf

[18] **Johan Sanneblad**. *GapiDraw.*

http://www.gapidraw.com/

[19] **Gould, J.D.** *An experimental study of writing, dictating, and speaking.* In Requin, J., editor, Attention & Pe~ormance VII, pp. 299-319. Lawrence Erlbaum,

1978.

[20] **Darren Holliday.** *Maps and Visualisation in Today's Digital World.*

http://www.gisca.adelaide.edu.au/~dhollida/new_tech/page.html (2003-05-26)

[21] **Shashi Shekhar and Yan Huang and Judy Djugash and Changqing Zhou.** *Vector map compression: a clustering approach.* Proceedings of the tenth ACM international symposium on Advances in geographic information systems 2002 pages 74-80.

[22] **Isaac Newton.** *Principia* (1687), Translated by Andrew Motte 1729.

[23] **Sven Widmalm.** *Mellan kartan och verkligheten, Geodesi och kartläggning.* Uppsala, Institutionen för idé- och lärdomshistoria, Univ. [distributör], 1990 ;

[24] **Peter H. Dana.** *Geodetic Datum Overview.* Lecture notes. Department of Geography, University of Texas at Austin, 1995.

http://www.colorado.edu/geography/gcraft/notes/datum/datum_f.html (2003-05-26)

[25] **Peter H. Dana.** *Map Projection Overview.* Lecture notes. Department of Geography, University of Texas at Austin, 1995.

http://www.colorado.edu/geography/gcraft/notes/mapproj/mapproj_f.html (2003-05-26)

[26] **Kenneth R. Castleman.** *Digital image processing*. Englewood Cliffs, N.J. : Prentice Hall, 1996

[27] A description of the Cohen Sutherland Line-Clipping algorithm can be found on: http://www.cc.gatech.edu/grads/h/Hao-wei.Hsieh/Haowei.Hsieh/mm.html (2003-05-26)

[28] **Unkown author.** *Provably correct and practical intersection testing of two segments of a circle or two georectangles.*

http://okmij.org/ftp/circle-segments-intersect-p.html (2003-05-26)

[29] **EMTEC.** *BTGPS Bluetooth GPS receiver datasheet.*

http://www.emtac.com/products/bluetooth/datasheet_btgps.html