Interacting with our Environment through Sentient Mobile Phones

Diego López de Ipiña[1], Iñaki Vázquez[2] and David Sainz[3]

Faculty of Engineering, University of Deusto, Bilbao, Spain

Abstract. The latest mobile phones are offering more multimedia features, better communication capabilities (Bluetooth, GPRS, 3G) and are far more easily programmable (extensible) than ever before. So far, the "killer apps" to exploit these new capabilities have been presented in the form of MMS (Multimedia Messaging), video conferencing and multimedia-on-demand services. We deem that a new promising application domain for the latest Smart Phones is their use as intermediaries between us and our surrounding environment. Thus, our mobiles will behave as personal butlers who assist us in our daily tasks, taking advantage of the computational services provided at our working or living environments. For this to happen, a key element is to add senses to our mobiles: capability to see (camera), hear (michrophone), notice (Bluetooth) the objects and devices offering computational services. In this paper, we present a solution to this issue, the MobileSense system. We illustrate its use in two scenarios: (1) making mobiles more accessible to people with disabilities and (2) enabling the mobiles as guiding devices within a museum.

1 Introduction

Ambient Intelligence (AmI) [1] involves the convergence of several computing areas: Ubiquitous Computing and Communication, Context-Awareness, and Intelligent User Interfaces. Ubiquitous Computing [2] means integration of microprocessors and computer services into everyday objects like furniture, clothing, toys and so forth. Ubiquitous Communication enables these objects to communicate with each other and the user by means of ad-hoc and wireless networking. Context-Awareness implies adding sentient capabilities both to the environment and the user mobile devices so that they can understand the current context of the user and offer them suitable services. An Intelligent User Interface enables the inhabitants of the AmI environment to control and interact with it in a natural (voice, gestures) and personalised way (based on preferences and context).

So far, a widespread adoption of Ubiquitous Computing, and as a consequence of Ambient Intelligence, has not been possible. The main reason for this has been that to make these ubiquitous environments reality is necessary to populate them with proprietary hardware and network infrastructure only available in research and industry labs. This hardware and network infrastructure provides access to services and also permits capturing the context of the user, so that the environment can react

López de Ipiña D., Vázquez I. and Sainz D. (2005).
Interacting with our Environment through Sentient Mobile Phones.
In *Proceedings of the 2nd International Workshop on Ubiquitous Computing*, pages 19-27
DOI: 10.5220/0002564000190027

sensibly to the current situation of the user. By sensing the current location, identity, or activity of the user, the surrounding environment can give an impression of having certain degree of intelligence, i.e. of being sentient. The inhabitants of those intelligent spaces are often aided by electronic devices that report their identity (presence) and enable the interaction with the environment.

A key factor to extend the adoption (deployment and use) of Ambient Intelligence is to utilize off-the-shelf hardware. Nowadays, by far the most commonly owned consumer device is the mobile phone. Last generation Smart Phones, converging mobile phone and PDA capabilities all in one, are more capable than ever before. They have the unique feature of incorporating short range (local) wireless connectivity (e.g., Bluetooth and Infrared) and Internet (global) connectivity (e.g., GPRS or UMTS) in the same personal device. Most of them are Java enabled programmable devices, and are therefore easily extensible with new applications. They also feature significant processing power, memory and added multimedia capabilities (camera, MP3 player). Most importantly, they are the ideal candidates to intermediate between us and the environment, since they are with us *anywhere* and at *anytime*.

The Bluetooth[3] sensing and communication capabilities of Smart Phones can be complemented with the utilisation of the mobiles built-in camera and microphone to sense the objects, devices and services available in the surroundings. So, it is possible to turn a mobile phone into a sentient device which sees (camera) and listens to (Bluetooth) surrounding services. The core of this paper illustrates how the Smart Phone sentient features can be applied to sense and interact with the objects in a Smart Environment [4], and also to enable user/mobile interaction in a more natural/accessible way. Thus, a Smart Phone can resemble a universal personal assistant that makes our daily life much simpler.

The structure of this paper is as follows. Section 2 describes TRIP, a 2-D barcode-based identification and location system, aimed for PCs. In Section 3 we adapt TRIP for mobile devices into the MobileEye system. Section 4 describes MobileSense, an extension to MobileEye, which adds further sensing capabilities to mobile phones. Section 5 explains how the infrastructure developed for MobileSense and MobileEye is applied to a real case, the creation of mobile phone-based museum guiding system. Section 6 discusses some related work and Section 7 finishes with some conclusions.

2 Sensing Context by Reading Barcodes

A sentient entity (environment or device) senses the user current context so that it can adapt its behaviour and offer, without her explicit intervention, the most suitable actions. A key part of every sentient system is to gather the context information of the user such as her location, identity, current time or current activity. Several sensing technologies [5] have been proposed in the last decade, focusing especially in the location-sensing area, since location is considered by far the most useful attribute of context. Previous work of one of the authors in one of those technologies, namely TRIP, motivated our aim of making mobiles more sentient.

TRIP (Target Recognition using Image Processing) [6] is a vision-based sensor system that uses a combination of 2-D circular barcode tags or *ringcodes* (see Fig. 1), and inexpensive CCD cameras (*e.g.* web-cams, CCTV cameras or even camera phones) to identify and locate tagged objects in the cameras' field of view. Compared with untagged vision-based location systems, the processing demands of TRIP are low. Optimised image processing and computer vision algorithms are applied to obtain, in real-time[1], the identifier (*TRIPcode*) and *pose* (location and orientation) of a target with respect to the viewing camera. Fig. 2 depicts the video filtering process undertaken to a frame with a TRIPtag. For more details on the algorithms refer to [6].

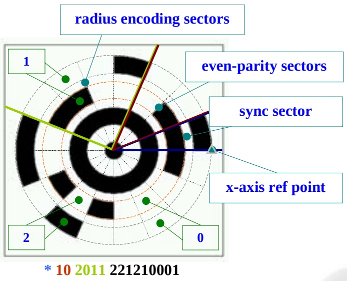
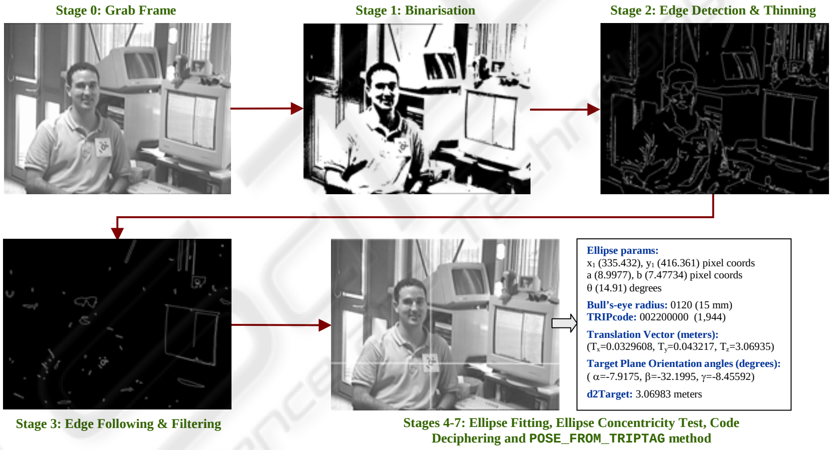


Fig. 1. TRIPcode of radius 58mm and ID 18,795. TRIP's 2-D printable and resizable ringcodes (*TRIPtags*) encode a ternary number in the range 1-19,683(3^9 - 1) in the two concentric rings surrounding the bull's-eye of a target. These two rings are divided into 16 sectors.


Stage 0: Grab Frame
Stage 1: Binarisation
Stage 2: Edge Detection & Thinning
Stage 3: Edge Following & Filtering
Stages 4-7: Ellipse Fitting, Ellipse Concentricity Test, Code Deciphering and POSE_FROM_TRIPTAG method

Ellipse params:
x_1 (335.432), y_1 (416.361) pixel coords
a (8.9977), b (7.47734) pixel coords
θ (14.91) degrees

Bull's-eye radius: 0120 (15 mm)
TRIPcode: 002200000 (1,944)

Translation Vector (meters):
(T_x=0.0329608, T_y=0.043217, T_z=3.06935)

Target Plane Orientation angles (degrees):
(α=-7.9175, β=-32.1995, γ=-8.45592)

d2Target: 3.06983 meters

Fig. 2. Identifying and Locating the TRIPtag attached to a user's shirt in a 768x576 pixel image. Location extraction is only possible when the camera used is calibrated.

TRIP is a cost-efficient and versatile sensor technology. Only low-cost, low-resolution cameras are required. The spare computing resources within a LAN can be allocated to run the TRIP image parsing software. Commonly available monochromatic printers can be used to print out the black and white TRIPtags. Its off-

[1] Over 20 640x480 pixel images can be processed in a Pentium IV running the C++ implementation of TRIP.

the-shelf hardware requirements and software nature (downloadable [6]) enable its installation in any environment equipped with some PCs.

3 MobileEye: Applying TRIP to Mobiles

Most of the newest Smart Phones feature a camera and programming facilities (Java [7] or Symbian APIs [8]) to capture images and deliver them to processing servers via Bluetooth (local) or GPRS (global). These technological factors motivated our adaptation of the TRIP technology to camera phones. We want to make mobile phones reactive not only to our explicit interactions (key presses) and network events (incoming call), but also to the objects and services at the location where the user is. In essence, we aim to add sensing capabilities to mobile phones. As a first result of this effort we delivered MobileEye.

The MobileEye application installed in a Smart Phone displays augmented, enriched views of tagged objects in the user surroundings (e.g. a painting in a museum), or alternatively renders simple forms to input parameters and issue operations over the sighted objects (e.g. the front door of our office). In MobileEye, every augmented object is associated to a web service. When we want to obtain extra information about an object or issue a command to it, we are really interacting with its web service counterpart. In fact, we are simply invoking methods on the web services (e.g. `MuseumWebService` or `DoorWebService`). MobileEye is downloadable from [6].

3.1 MobileEye Implementation

The MobileEye system presents a client/server architecture. Despite the increasing computational capabilities of mobile phones, they are still generally[2] unsuitable for undertaking heavy processing tasks such as image processing. This explains why in our system, images captured by mobiles are uploaded into processing servers. Even if the mobiles were able of processing images, MobileEye would still need to contact a server to obtain additional information or recover the operational interfaces.

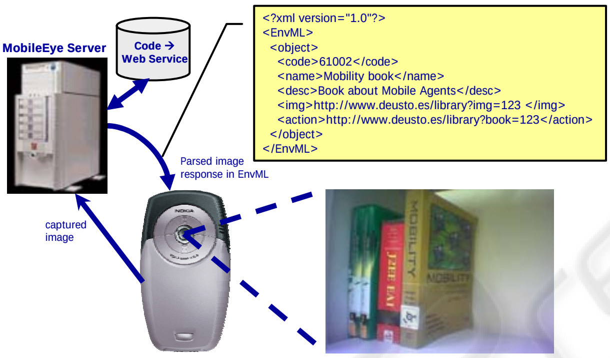
GPRS and UMTS give global access to information and allow the offload of images to processing servers located far away from the tagged objects. More importantly, they enable the mobiles to access the web service counterparts of the objects, anywhere in the Internet. On the other hand, due to performance and cost factors, it is convenient to use within controlled environments (e.g. universities or offices) local severs accessible through PAN networks (e.g. Wi-Fi or Bluetooth). In consequence, the MobileEye server can be accessed by co-located users via Bluetooth and by remote users through GPRS.

Fig. 3 shows the architecture of the MobileEye system. Every item in the environment augmented with computational functionality is tagged with a ringcode. Jpeg images captured from tagged objects are delivered to a server. This server

[2] Some latest models such as the Nokia 6600 and 6630 or the Ericsson P900 have been applied to undertake such heavy processing tasks.

delivers responses in EnvML format, which are interpreted by the player in the MobileEye client. EnvML is a simple XML-based language we have devised.

For the case where the mobile and the server are co-located (within Bluetooth range of each other), the MobileEye client discovers the Bluetooth-enabled server via Bluetooth SDP (Service Discovery Protocol) and delivers the image content via Bluetooth RFCOMM (wireless serial port). The responses are delivered from the server using the same mechanism.



```
<?xml version="1.0"?>
<EnvML>
  <object>
    <code>61002</code>
    <name>Mobility book</name>
    <desc>Book about Mobile Agents</desc>
    <img>http://www.deusto.es/library?img=123 </img>
    <action>http://www.deusto.es/library?book=123</action>
  </object>
</EnvML>
```

Fig. 3. MobileEye application architecture.

There is only one core J2EE-based implementation of the MobileEye server with two communication channels: an HTTP and a Bluetooth one. Once an image has been posted via HTTP or Bluetooth, the server uses the TRIP software to extract the identifier of a ringcode. A database is used to map an identifier into a MobileEye object, which contains a name, a description, and a URL pointing to its WSDL interface. An XSL style sheet is applied to transform the WSDL interface into EnvML, which is delivered to the mobile. Operations issued from the MobileEye rendered graphical interface (obtained from the EnvML) are received via HTTP or Bluetooth, and delegated to action JavaBeans that undertake the user commands.
The MobileEye client is MIDP 2.0 application which makes use of the MMAPI and Bluetooth API extensions of J2ME. The EnvML player uses TinyXML to transform the XML content received into a J2ME graphical interface.

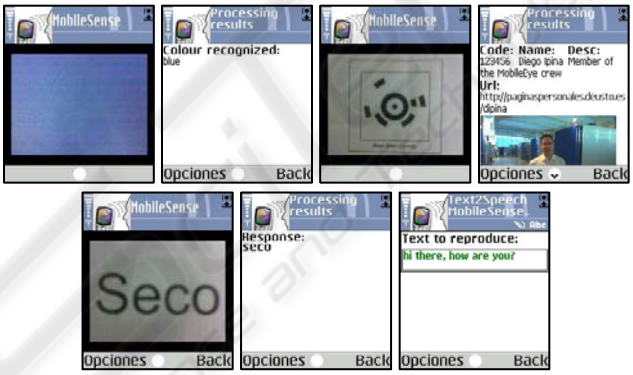
3.2 MobileEye Performance

The MobileEye system has been tested on a Nokia 6630 device which communicates via HTTP or Bluetooth with a 3.2 GHz Bluetooth-enabled Pentium IV with 1 Gbyte RAM, running a Tomcat 5 application server and Blue Cove[9]. The server-side Java implementation of TRIP was able of processing 30 160x120 pixel JPEG frames per second. On the other hand, a Nokia 6630 was capable of sending 11 of those frames per second to the server-side. The size of the frames delivered was on average 2208 bytes. Therefore, the effective frame transfer rate through the Bluetooth channel, taking into account the server side acknowledgments, was around 200 Kbps. The biggest drawback we experienced while using Bluetooth was its slow service discovery mechanism. Our J2ME-based Bluetooth client invested around 12 seconds on average to discover the MobileEye Server.

4 MobileSense: Adding Senses to Mobiles

Mobile phones are turning into essential tools which assist us in our daily tasks. Among other things, mobiles entertain us, help us to communicate with other people and increasable serve as PDAs. Unfortunately, mobile phones are hard to use for people with disabilities (blind, deaf) or advanced age. Thus, it is necessary to devise solutions to offer those facilities to the people that precisely need more help.

Some companies have offered partial solutions, both software [10] and hardware [11], to this problem. All these solutions are good but present two main problems: (1) they are addressed to only a specific collective (blind or old people), and (2) they are not freely available. MobileSense is a more generic solution which leverages from open source technologies to improve the accessibility of mobile phones. It improves MobileEye sensing capability by adding mechanisms to synthesize voice, apply OCR, recognize colours, or even understand basic voice commands. MobileSense pursues the following design goals: (1) be available to the widest range of currently sold mobile phones, (2) be accessible by anybody, without focusing in a specific disability group and (3) operate both in managed (e.g. home/office) and unmanaged environments (street/country side).



Colour recognized:
blue

Code: Name: Desc:
123456 Diego Ipina Member of the MobileEye crew
Url:
http://paginaspersonales.deusto.es/dipina

Response:
seco

Text to reproduce:
hi there, how are you?

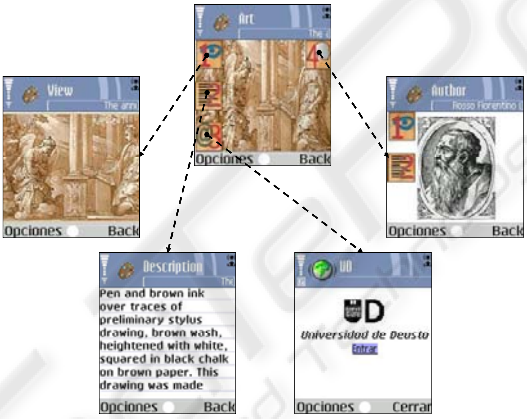
Fig. 4. MobileSense in action.

The MobileSense implementation is available at [6]. Fig. 4 shows some snapshots of the MobileSense client. All the multimedia processing tasks (image processing, speech recognition, speech synthesis) are delegated to a server accessible both locally (through Bluetooth) or remotely (GPRS and UMTS). The server leverages from a plethora of open source utilities: (1) transforms text into voice with FreeTTS [12] generating a WAV file as response, (2) recognises TRIP ringcodes and displays an interface or extra information about an object, by means of EnvML, (3) undertakes text recognition using the freely available GOCR [13], (4) undertakes colour recognition using a custom-built colour recognition algorithm and (5) processes basic voice commands through Sphinx [14].

It is interesting to mention that the OCR process incorporated within MobileSense demands far higher image resolution than TRIP to achieve reliable results. MobileSense, by default, operates with 160x120 pixel images, an image size which guarantees a good overall performance of the system, but, unfortunately, yields frames good enough to only reliably recognize banners with big fonts (see Fig. 4).

5 MoMu: the Mobile Phone as a Guiding Device

When we visit a museum we often rent a mobile device which guides us offering additional information about the pieces of art we observe. We usually type in a number in the device which responds with an explanation in our own language about the artwork referred by that code. This information complements the insufficient details usually placed in the form of a card beside the piece of art.



Pen and brown ink over traces of preliminary stylus drawing, brown wash, heightened with white, squared in black chalk on brown paper. This drawing was made

Fig. 5. MoMu portraying information about “The Annunciation” by Fiorentino

The MoMu (Mobile Museum) application transforms a mobile phone into a guiding device. The communication (Bluetooth/GPRS) and multimedia (audio/video playback) capabilities of mobiles are used to obtain enriching information about the artworks in a museum. In each room within a museum, a local Bluetooth server offers multimedia content about the art pieces in that location. The MIDP 2.0 based MoMu application on the mobile uses Bluetooth discovery to detect the locally available media server. Then, the user either selects an item from a list of pieces of art returned by the server or points her mobile to a TRIP ringcode placed beside each piece of art. The server identifies the piece of art selected and sends extra information about that piece of art. Our current implementation, see Fig. 5, offers: information about the author, detailed description of a work, audio or video content, or even links to external websites providing more info or allowing the purchase of some related merchandising. MoMu makes use of the MobileSense server-side infrastructure.

6 Related Work

Other researchers have also considered the use of Smart Phones or custom-built wireless devices to interact with Sentient Environments. The SMILES (SMartphones for Interacting with Local Embedded Systems) [15] project proposes the use of Smart Phones as universal remote controllers. They define a service discovery protocol built on top of Bluetooth SDP, an interaction mechanism to operate over the services discovered, and a payment protocol to pay for their use. They limit to the use of Bluetooth for service discovery and GPRS for the download of Java applications implementing the logic that permits the interaction with the objects sensed. Our approach can use Bluetooth, vision and even audio for sensing other devices and services, and can operate over the services discovered via Bluetooth or GPRS.

Personal Server [16] is a small-size mobile device that stores user's data on a removable Compact Flash and wirelessly utilizes any I/O interface available in its proximity (e.g., display, keyboard). Its main goal is to provide the user with a virtual personal computer wherever the user goes. Unlike Personal Server which cannot connect directly to the Internet, Smart Phones do not have to carry every possible data or code that the user may need; they can obtain on demand data and interfaces from the Internet.

CoolTown [17] proposes *web presence* as a basis for bridging the physical world with the World Wide Web. For example, entities in the physical world are embedded with URL-emitting devices (beacons) which advertise the URL for the corresponding entities. Our model proposes *web service presence* and makes use of only off-the-shelf hardware.

Rohs et al.[18] have also used the built-in cameras of consumer mobile phones as sensors for 2-dimensional visual codes. Their matrix-shaped codes have been used to extract the phone number from business cards on which their barcodes are printed. They suggest to use these codes in public displays in airports or train stations so that they can direct customers to on-line contact just by retrieving the url associated to each code. Our solution has the same barcode recognition functionality but it also adds other sensing capabilities and is not only a client-side solution.

7 Conclusions

This work has illustrated how the interactions with our environment can be facilitated significantly by means of our mobile phones. Smart Phones can sense what other devices or services are around and present us with additional information about them or interfaces that enable us to operate over those objects. Our adaptation of a mobile phone into a sentient device has enabled us to interact in a more natural way with objects which otherwise we would have not considered augmenting with computational services, such as a door or a painting. Moreover, our work permits people with disabilities (blind, deaf) [19] to also make use of a mobile phone. The MoMu system applies the MobileSense concept to a museum, enabling a user to interact through her mobile with the artworks and so obtain extra information about them. As future work we are planning to apply the MobileSense infrastructure in the development of sentient services for outdoor environments.

Acknowledgements

This work has been partly financed by the Catedra de Telefónica Móviles at University of Deusto, Bilbao, Spain. The work on the TRIP sensor was carried out at the Laboratory for Communications Engineering, University of Cambridge, UK.

References

1. Shadbolt N. Ambient Intelligence. IEEE Intelligent Systems, July/August 2003 (2003) 2-3
2. Weiser M. The Computer for the 21st Century. Scientific American, (1992) vol. 265, no. 3, pp. 94-104.
3. Bluetooth. http://www.bluetooth.org/.
4. Hopper, A. The Clifford Paterson Lecture, 1999 Sentient Computing. Philosophical Transactions of the Royal Society London, vol. 358, no. 1773, pp. 2349-2358, August 2000.
5. Hightower J., Borriello G. Location Systems for Ubiquitous Computing. IEEE Computer 2001; 34(8): 57-66
6. López de Ipiña D., Mendonça P. and Hopper A. TRIP: a Low-Cost Vision-Based Location System for Ubiquitous Computing. Personal and Ubiquitous Computing, Springer, vol. 6, no. 3, pp. 206-219, May 2002. Software downloadable at: http://www.ctme.deusto.es/trip
7. Java 2 Platform, Micro Edition (J2ME). http://java.sun.com/j2me/
8. Symbian OS: The Mobile Operating System. http://www.symbian.com/
9. Blue Cove: An open source implementation of the JSR-82 Bluetooth API for Java, http://sourceforge.net/projects/bluecove/
10. TALKS for Series and Nokia Communicator, http://www.talx.de/index_e.shtml, 2004
11. OWASYS 22C y 112C, http://www.owasys.com/index_en.php, 2004
12. FreeTTS 1.2beta2, http://freetts.sourceforge.net/docs/index.php, 2004
13. Schulenburg J., GOCR, http://jocr.sourceforge.net, 2004
14. Walker w. et al., Sphinx-4: A Flexible Open Source Framework for Speech Recognition, http://cmusphinx.sourceforge. net/sphinx4/doc/Sphinx4Whitepaper.pdf, 2004
15. Iftode L, Borcea C., Ravi N., Kang N., and Zhou P. Smart Phone: An Embedded System for Universal Interactions, Proceedings of 10th International Workshop on Future Trends in Distributed Computing Systems FTDCS (2004).
16. Want R., et al. The Personal Server: Changing the Way We Think about Ubiquitous ComputingIn *Proceedings of Ubicomp2002*, pp. 194–209. Springer LNCS, September 2003.
17. Kindberg T., Baron J., et al. People, places, things: Web presence for the real world. 3rd IEEE Workshop on Mobile Computing Systems and Applications, December 2000.
18. Rohs M., Gfeller B., Using Camera-Equipped Mobile Phones for Interacting with Real-World Objects. Advances in Pervasive Computing, Austrian Computer Society (OCG), ISBN 3-85403-176-9, pp. 265-271, Vienna, Austria, April 2004
19. López de Ipiña et al. Accesibilidad para Discapacitados a través de Teléfonos y Servicios Móviles Adaptables. JANT 2004, Bilbao, SPAIN, Diciembre 2004.