

# **Mobile Media Distribution in Developing Contexts**

A thesis submitted in fulfillment  
of the requirements for the degree of

**MASTER OF SCIENCE**

at

**UNIVERSITY OF CAPE TOWN**

by

**Graeme Smith**

and is all my own work

Supervisor: Prof. Gary Marsden

January 2011

## **Abstract**

There are a growing number of mobile phones being used in developing contexts, such as Africa. A large percentage of these phones have the capability to take photographs and transmit them freely using Bluetooth. In order to provide people with media on their mobile phones public displays are becoming more common. Three problems with current public displays – cost, security and mobility – are discussed and system proposed that uses a mobile phone as a server. Media is displayed on specially designed paper posters, which users can photograph using their mobile phones. The resulting photographs are sent, via Bluetooth, to the server, which analyses them in order to locate a specially designed barcode, representing the media, which is then decoded and the requisite media returned to the user. Two barcoding systems are tested in laboratory conditions, and a binary system is found to perform best. The system is then deployed on a campus transportation system to test the effects of motion. The results show that the system is not yet ready for deployment on moving transport.

**Table of Contents**

**Abstract ..... i**

**Table of Figures..... vi**

**List of Tables ..... vii**

**Acknowledgements and Plagiarism Declaration .....viii**

**Chapter 1 - Introduction ..... 1**

    1.1 Context ..... 1

        1.1.1 Mobile Use in Africa..... 1

        1.1.2 The desire for – and distribution of – media to mobile phones ..... 2

        1.1.3 Problems with current situated display solutions..... 4

        1.1.4 Proposed solution..... 5

        1.1.5 Design Challenges..... 6

        1.1.6 Mobile Image Recognition..... 7

    1.2 Problem Statement..... 7

    1.3 Research Questions..... 8

        1.3.1 Question 1..... 8

        1.3.2 Question 2..... 8

        1.3.3 Question 3..... 8

        1.3.4 Question 4..... 9

        1.3.5 Question 5..... 9

    1.4 Thesis organization..... 9

**Chapter 2 - Background ..... 10**

    2.1 Public displays..... 10

        2.1.1 Hermes..... 11

## PRELIMINARIES

2.1.2	ContentCascade .....	12
2.1.3	WebWall .....	13
2.1.4	CityWall .....	14
2.1.5	Publix.....	15
2.1.6	Original Snap 'n Grab System.....	16
2.2	Bluetooth.....	18
2.2.1	Bluetooth capabilities.....	18
2.3	Mobile Image Recognition Projects.....	18
2.3.1	Interactive Museum Guide .....	19
2.3.2	CD Cover Recognition .....	19
2.3.3	Discussion.....	19
2.4	Barcodes.....	20
2.4.1	1D Barcodes.....	20
2.5	2D Barcodes.....	23
2.5.1	Examples of 2D Barcodes.....	23
2.5.2	Applications of 2D Barcodes .....	24
2.5.3	Problems with 2D barcodes related to our situation .....	25
2.5.4	Visual Code by Rohs.....	26
2.6	Summary.....	28
<b>Chapter 3 - System Design .....</b>		<b>29</b>
3.1	Introduction.....	29
3.2	Hardware and Platform .....	29
3.2.1	Windows Mobile.....	29
3.2.2	Phone Hardware .....	29
3.3	Poster Design .....	30
3.3.1	Barcode.....	30

## PRELIMINARIES

3.3.2	Target Lines.....	31
3.4	Media Packs .....	32
3.4.1	Layout.....	32
3.4.2	Generation of Media Packs .....	33
3.5	Snap 'n Grab Lite Server System.....	35
3.5.1	Bluetooth .....	36
3.5.2	Image Recognition Algorithm.....	37
3.5.3	Logging.....	45
3.6	Summary .....	46
<b>Chapter 4 - User Study 1 .....</b>		<b>47</b>
4.1	Experiment Round 1.....	47
4.1.1	Design.....	47
4.1.2	Results.....	49
4.2	Changes.....	50
4.2.1	Target Lines.....	50
4.2.2	Image Recognition.....	51
4.3	Experiment Round 2.....	52
4.3.1	Design .....	52
4.3.2	Results.....	53
4.3.3	Recommendations.....	53
4.3.4	Comparison.....	54
<b>Chapter 5 - User Study 2 .....</b>		<b>56</b>
5.1	Experiment Round 3.....	57
5.1.1	Design.....	57
5.1.2	Observations .....	58
5.2	Experiment Round 4.....	58

## PRELIMINARIES

5.2.1	Changes.....	58
5.2.2	Results.....	59
5.3	Discussion.....	60
5.3.1	Round 3 Discussion.....	60
5.3.2	Image Recognition.....	61
5.3.3	Bluetooth issues .....	63
<b>Chapter 6 - Conclusion and Future Work .....</b>		<b>65</b>
6.1	Discussion of project .....	65
6.2	Discussion of algorithm .....	66
6.3	Discussion of System .....	66
6.3.1	Platform.....	66
6.3.2	Viability.....	67
6.3.3	Future .....	67
6.4	Discussion of Literature.....	68
6.5	Discussion of project results.....	68
6.5.1	Question 1.....	68
6.5.2	Question 2.....	69
6.5.3	Question 3.....	69
6.5.4	Question 4.....	69
6.5.5	Question 5.....	69
6.6	Conclusion .....	70
6.7	Future work.....	70
<b>References.....</b>		<b>i</b>
<b>Appendices.....</b>		<b>iv</b>
Appendix A – Bresenham’s Algorithm .....		iv
Appendix B – Table of Results from User Study 1 – Round 1 .....		v

## PRELIMINARIES

Appendix C - Table of Results from User Study 1 – Round 2 .....	vi
Appendix D – Table of Results from User Study 2 .....	viii
Appendix E - Published Papers.....	x

### Table of Figures

Figure 1 - The Snap 'n Grab System.....	4
Figure 2 - A user interacting with a Snap 'n Grab display by taking a photo with is mobile phone [19].....	5
Figure 3 - An example of a VisualCode by Rohs et al [36]..	7
Figure 4 - The Hermes Photo Display [12].....	11
Figure 5 - A user interactive with a ContentCascade display [34] .....	12
Figure 6 - A WebWall display showing a video, picture, advert and poll [17].....	13
Figure 7 - The CityWall display with Flickr photos [33].....	14
Figure 8 - User interacting with a Publix display [41].....	15
Figure 9 - A Snap 'n Grab display.....	16
Figure 10–The Bluetooth protocol [7] .....	18
Figure 11 - A UPC barcode. ....	21
Figure 12 - 2D barcoding systems.....	24
Figure 13 - VisualCode, showing the guidebars and corner elements. ....	27
Figure 14 - Diagram of adapted UPC barcode used on the posters.....	31
Figure 15 - A poster showing barcode, guidebars and target lines.....	32
Figure 16 - Typical contents of a media pack folder. ....	32
Figure 17 - Xml file describing a media pack.....	33
Figure 18 - Screenshots of the workflow followed on the mobile media pack generator .....	34
Figure 19 - Screenshots of the desktop media pack generator .....	35
Figure 20 - Overview of system use .....	36
Figure 21 - Diagram showing how the starting point of the scanline is located...	43
Figure 22 - Overview of the system.....	46

## PRELIMINARIES

Figure 23 - The posters stuck on the wall of the experiment room.....	49
Figure 24 - Examples of the photos taken by users in the first round of experiments .....	50
Figure 25 - The two sets of posters used in round 1 and 2 of the experiments....	51
Figure 26 - Example of how naive thresholding can lead to the guidebars being enlarged or shrunk. ....	52
Figure 27 - Example of two poorly taken photos.....	55
Figure 28 - One of the posters, complete with instructions, used in round 3 and 4 of the experiments .....	56
Figure 29 - Users taking part in the 4th round of experiments .....	59

### List of Tables

Table 1 - Mobile Costs in South Africa [1, 11, 31, 42] .....	2
Table 2- Example system log .....	46
Table 3 - Result table for second round of experiments .....	54
Table 4 - Result table after trimming data .....	54

## Acknowledgements

To my supervisor, Gary. Thanks for the many hours of corrections, and the not so many “*No, no, no! Have some coffee then do it again*”s over the last the few months. Also, thanks for the coffee, the muffins, the braais and the dinners.

To those in the ICT4D lab: Shikoh, Ian, Sean, Ilda, Raymond, Chris, Fritz, Michael, Sena and Thomas. Thanks for the fun lab meetings, the braai’s and the constant complaining about how cold the lab is.

To my bursars, Telkom, without whose funding this MSc would not have possible.

Firstly, Digz9r 1.0 and 2.0. Thanks to Justin, Gareth, Jon, Werner, Sam, Nick and Tim. Thanks for putting up with me and for always being ready and willing to do anything that is obviously not a wise idea.

Then, to my wife, Dee. One year down, many, many more to go. Thanks for always supporting me, loving me and cleaning up after me. I love you.

Penultimately, thanks to my parents, Gavin and Alison. Everything I am today is because of you two. Thanks for being great role-models and parents.

Lastly, this would not have been possible, or enjoyable in the least, without the constant rock provided by my faith in my God, who promises that “this too shall pass” and makes it a lot easier to put an MSc into eternal perspective.

## PLAGIARISM DECLARATION

1. I know that plagiarism is wrong. Plagiarism is using another’s work and to pretend that it is ones own.
2. I have used the ACM as the convention for citation and referencing. Each significant contribution to, and quotation in, this thesis from the work, or works of other people has been attributed and has cited and referenced.
3. This thesis is my own work.

## 1. Introduction

### 1.1 Context

#### 1.1.1 Mobile Use in Africa

ITU reports that in 2010, in Africa, there were 13 million fixed line subscribers [22]. The same report also states that there were 333 million mobile subscribers in Africa in 2010. This means that any technological system designed for developing contexts, such as most of Africa, must take considerable notice of the ubiquity of mobile devices on the continent.

A common misconception is that, whilst there may be a large number of mobile phones in circulation in Africa, surely they must all be very basic, bottom-of-the-range models with low functionality. Almost counter-intuitively this turns out not to be the case. From our experience working into areas such as Khayelitsha, a township near Cape Town, South Africa, we have found that people are willing to spend a relatively large amount of money in order to purchase a good mobile phone. Gitau reports that the majority of people in her target group, low income people in Khayelitsha, had a mobile phone with a camera and Bluetooth functionality [18]. It is possible to purchase a “feature phone”, a mobile phone with a camera, Bluetooth and data capabilities for only \$50 in South Africa.

Whilst there may be a high concentration of feature phones in Africa, the added functionality offered by these phones is of no use to developers if the owners of these phones do not know how to make use of these features. What we have seen, however, is that people do know how to get the maximum utility out of their handsets. Marsden et al found that people in townships kept complete photo albums of pictures taken with the camera on their mobile phone, on their mobile phone [28, 29]. Gitau and Donner, in their study of mobile phone users in Khayelitsha, found that people were very conscious of all the costs associated with mobile phone use, and thus were well versed in the use of free technologies such as Bluetooth [15]. Bidwell, in her work in the Eastern Cape of South Africa [5, 6], found that people making use of the informal taxi system in the area used

the time to charge their phones using the car's cigarette lighter whilst swapping media between their phones using Bluetooth.

It cannot be overemphasized how little disposable income people in developing contexts have. Leibbrant reports that the average annual income of a black individual in South Africa in 2008 was 9790ZAR (about \$1200) [25]. Thus, they have to live on about \$100 per month.

The cost of mobile use in South Africa, as shown below, makes it prohibitive for an individual earning in that region to use their mobile phone to retrieve media such as images, videos or music to the phone using mediums such as mobile internet.

Provider	Cost of 1 minute phone call	Cost of SMS	Cost of 1Mb data (out of bundle)
Vodacom	R2.85	R0.80	R2.00
MTN	R1.75	R0.50	R1.00
CellC	R1.50	R0.80	R1.00
8ta	R1.50	R0.50	R1.00

Table 1 - Mobile Costs in South Africa [1, 11, 31, 42]

### 1.1.2 The desire for – and distribution of – media to mobile phones

Despite this lack of finances, people in developing communities are, nonetheless, very eager to get media for their mobile phones [15]. The type of media ranges from music tracks of popular artists to information about a political party in an upcoming election.

In addition to this desire for popular media, there are other groups, such as NGOs and governmental departments, which need to get information and media to people [10]. This can include HIV education material, information about upcoming elections or details of jobs in the area. In order to get this information to the people, the institutions need to overcome certain obstacles. The first of these, as has already been mentioned, is the cost. The user should not have to carry a cost in order to download media or there will be no incentive to download it. For this reason, information and media from these sources should

## Chapter 1 - Introduction

be freely available. Secondly, institutions need to overcome the low literacy levels inherent in developing regions. In more than 75% of African nations, over 50% of the adult population is textually illiterate [43]. For this reason audio or video media are preferable to written media.

One method of media dissemination to mobile devices is public situated displays. These are, in general, large screens, which display information about media which users can download, through some form of interaction with the system, onto their mobile devices. Examples of these are the Hermes system, developed by Cheverst et al [12] and Snap 'n Grab, developed by Maunder et al [30].

To circumvent the cost of data transfer when downloading media, public situated displays sometimes make use of Bluetooth to transfer the media to the users' mobile phones. Hermes was one of the first publicly situated displays to utilize Bluetooth technology to allow people to download media. It did, however, rely on a touch screen to allow users to select which data they wanted. This meant that only one person was able to interact with the system at any time, and that display needed to be positioned in such a way that it was able to be easily touched by users.

The Snap 'n Grab system was the first system to allow users to Bluetooth their selection, by means of a photograph, to the system in order to choose which media they desired. The system consists of a large screen display and a computer to drive it. The screen displays images representing content on the computer. This content can be pictures, videos, music files or text. It can represent job offers, AIDS information or simply (as was the case when the system was deployed in Khayelitsha, a township near Cape Town, South Africa) recordings of the local gospel choir. Users can take a photo of an image representing a subject they wish to know more about. Having done this, they can then send it, via Bluetooth, to the system. The system carries out image recognition on the image and replies by sending data, via Bluetooth, back to user (see Figure 1). All this is done at no cost to the user, as Bluetooth is free. Snap 'n Grab can also allow users to create their own "media packages": an account can be set up by sending the

## Chapter 1 - Introduction

system a v-card, followed by the media for the package. This new package will then be displayed on the screen.

Another important aspect of the Snap 'n Grab system is that it does not require any software to be downloaded onto the user's cell phone: all they are required to do is take a photo, using the cell phone's built-in camera and to send it, via Bluetooth, to the system. This is a task people have become used to performing and thus is not an unknown or intimidating endeavour.



Figure 1 - The Snap 'n Grab System

### 1.1.3 Problems with current situated display solutions

Some drawbacks to the current Snap 'n Grab system are *cost*, *security* and *mobility*. A large screen display of 40", the average size of a Snap 'n Grab screen, costs about \$1000. The computer to drive the screen would add an extra \$300. These costs are prohibitive in developing communities.

## Chapter 1 - Introduction

In places where there is a high crime rate, a high theft rate in particular, it is difficult to find a balance between system accessibility and security. Ideally the system should always be available for people to use; however, it is costly and difficult to ensure that the screens and computers are not easily stolen.

The big screens are further constrained by the fact that they require a power source. This means that they could not be easily deployed in vehicles or places where electricity is not constantly available, such as villages that rely on solar power or intermittent generators.



**Figure 2 - A user interacting with a Snap 'n Grab display by taking a photo with is mobile phone [19]**

### **1.1.4 Proposed solution**

We propose a solution to the above three problems.

Firstly, by replacing the big screen with either printed posters, or individual stickers for each item, and by replacing the computer with a cell phone, we can dramatically reduce the cost, greatly simplify the security issue and bypass the mobility problem. A high-end cell phone will cost about \$550, about 30% of the total cost of the original system. Secondly, security will be reduced to finding a suitable place to store the phone, such as a locked cupboard, or in the glove compartment of a vehicle. The posters will not need to be secured, as they will be of relatively low cost to replace. Finally, since the system is based on a cell phone, it is clearly mobile. It could be deployed in a vehicle, in the middle of a field or in a small village where there is no constant source of electricity. Obviously the cell

phone would need to be charged occasionally, but it would not need constant power.

### 1.1.5 Design Challenges

There are two main challenges facing this solution, the limited computational power of a mobile device when compared to a desktop means that a new image recognition solution needs to be developed. Additionally, moving from a large, dynamic, LCD display to small, static, paper posters will produce other challenges, discussed below.

Firstly, the processing power of a mobile phone is significantly less than that of a desktop computer. The Snap 'n Grab system is run on a Mac Mini with a 2GHz processor and 2Gb of RAM. At the start of this project the most powerful mobile phone available, the HTC Touch Pro, had a 528MHz processor and 288Mb of RAM. Additionally, almost all phones have no floating point processing capacity. Therefore, functionality such as the image recognition carried out by Snap 'n Grab need to be rethought as the SIFT algorithm [26] utilized in Snap 'n Grab is simply too slow to be implemented on a cell phone (it was reported as taking up to 219 seconds on a cell phone architecture [4]).

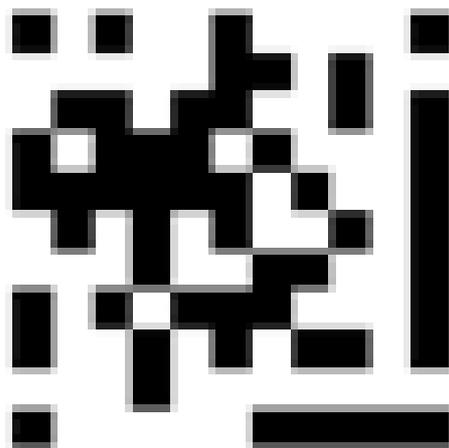
A simple solution would be to replace the pictures used in Snap 'n Grab with barcodes, like Shotcodes (see Figure 3). The problem with this, however, is twofold: Firstly, a barcode gives no indication of the media it represents. It is not human readable. Secondly, a large portion of the population in developing communities is textually illiterate. Therefore, even if the barcode was augmented with a textual description, this would not allow the users to easily ascertain what media they are being offered. Therefore an algorithm is needed that will be able to easily determine which image has been photographed, whilst maintaining the pictorial aspect of the Snap 'n Grab system.

One of the advantages of using a large LCD display is that it can be used to give instructions to users as to how to procure media. With paper posters, however, this is not as simple due to the lack of space, and the necessity to reprint the instructions for each poster.

The method of displaying the media available for download must also be investigated. Various options exist, such as posters mimicking the original displays, smaller stickers representing individual items, or hybrids of the two. Additionally it will be determined whether printing with a glossy or matte finish has any effect on the ability of the system to perform the image recognition.

### 1.1.6 Mobile Image Recognition

A lot of work has been done in recent years on using mobile phones as portals to media and information. A large subsection of this work falls under the general category of "Shotcodes". Shotcodes are graphical codes which can be photographed with a mobile phone and subsequently decoded by software running either on the phone itself or on a server elsewhere. There are many types of Shotcodes, such as QR Code, Datamatrix and Visual Code [23, 36]. These are discussed in more detail in Chapter 2.



**Figure 3 - An example of a VisualCode by Rohs et al [36]. Notice the guidebars on the right and bottom of the code.**

Usage of the system will be exactly the same as the original Snap 'n Grab. Users will photograph a poster representing media they are interested in. They will send the photo, via Bluetooth, to the server. The server will analyse the image and return the respective media to the user.

## 1.2 Problem Statement

This study aims to investigate the porting of an existing media distribution system to a mobile phone. The original system made use of a personal computer

## Chapter 1 - Introduction

and a large LCD display. The new system will use a mobile phone and paper posters.

The system must be able to:

- Receive images from users using Bluetooth. The system will listen for incoming request, and then intercept them to process the incoming files.
- Locate a barcode within the image.
- Decode the barcode in order to determine which media to return to the user.
- Return media to the user using Bluetooth.

### 1.3 Research Questions

#### 1.3.1 Question 1: Can we create an image recognition algorithm that runs efficiently on a mobile phone?

Shneiderman states that a user's perception of a system will begin to deteriorate after a certain response time threshold has past. This threshold is determined by the user's past experience of similar systems [38]. Therefore, we require that our system responds in the same time as the original Snap 'n Grab system. In testing, this response time was found to be 2.5 seconds. We therefore want the system to be able to correctly analyse submitted images and return the correct code in less than 2.25 seconds.

#### 1.3.2 Question 2: What is the best barcode form we can devise for the system?

Can we develop a barcode form that the system can accurately and efficiently locate and decode?

#### 1.3.3 Question 3: Can the system decode photographs taken in a static (laboratory) environment?

How accurately can the system recognise images taken by users from a broad demographic, using a variety of camera phones in different lighting conditions?

## Chapter 1 - Introduction

### **1.3.4 Question 4: Can the system decode photographs taking in a moving environment, such as public transport?**

Assuming Research Question 3 can be addressed successfully, will the algorithm be sufficiently robust to cope with the blurring, lighting and framing issues that might arise in capturing images in a moving vehicle?

### **1.3.5 Question 5: Can we design the posters in such a manner that users are able to determine what is required of them in terms of photographing the poster and submitting the photos to the system?**

Are there visual cues and layouts that will help users take photographs of the posters in a way that is optimal for the recognition algorithm to process?

## **1.4 Thesis organization**

This thesis is structured into several chapters.

The first chapter is an introduction to the context in which the work is set. The project is introduced, and the research questions are specified.

The second chapter is a review of the literature relevant to this study. It includes public displays, Bluetooth technology and image recognition on mobile devices.

Chapter 3 details the solution developed. The image recognition algorithm and system design are discussed.

Chapter 4 discusses the first round of user experiments undertaken to investigate the accuracy of the image recognition algorithm developed in a laboratory environment.

Chapter 5 details the second round of user experiments, where the system was deployed on public transport to determine whether users could interact with it on a moving vehicle.

Chapter 6 provides a discussion of the results of the experiments. It then concludes the thesis and discusses potential future work.

## 2. Background

In this chapter we will discuss previous work pertinent to this project. This project is an extension of the Snap 'n Grab system developed by Maunder et al [30] which falls into the category of publicly situated displays. We will look at previous projects involving public displays and discuss which aspects can be adapted for our use and which aspects we wish to improve upon.

Our system will make use of Bluetooth technology to enable free transfer of media between users and the server. A discussion of the Bluetooth protocol and its capabilities and limitations is undertaken.

Because our server will be a mobile phone, with limited processing power, we will need to approach the image recognition aspect of the system differently to how the original Snap 'n Grab system did so. For this reason we look at existing mobile image recognition projects.

Mobile image recognition is not yet at the stage where a photograph can be analysed to determine what item has been photographed. Therefore, other approaches must be investigated. Barcode reading with mobile devices is a mature problem, with many existing solutions. We shall look at the most popular of these and discuss how they could be adapted to suit our usage.

### 2.1 Public displays

Publicly Situated Displays have many uses. They can be used to display important safety information, such as traffic information on a highway; advertising; or material of interest to a specific community, such as photographs, events or news.

Interactive public displays allow users to affect the display in some way, usually by adding or changing media on the display; or be affected by it, usually by downloading media from it.

To investigate how users interact with public displays we will look at five existing public display projects.

### 2.1.1 Hermes



Figure 4 - The Hermes Photo Display [12]

The Hermes Photo Display was developed by Cheverst et al [12]. It made use of a touch screen display, which showed photographs that had been uploaded to it. Users could interact with the system by either uploading or downloading photographs to/from the system using their mobile phones.

To upload, a user would simply choose a photo on their mobile device, and then choose to send it via Bluetooth. Once the display has been discovered (the user's device has detected its presence and availability for contact via Bluetooth), the devices would be paired and the photo would subsequently be transmitted. It would then join the other photos in being displayed on the screen.

To download a photo a user would touch the photo of interest. The system would then search for nearby Bluetooth devices and display them on the screen. The user would touch their device and photo would be transmitted.

There are two items of interest with regards to the Hermes system. Firstly, we note that the system requires no client application on the user's device, an attribute desired for our system. Users are understandably distrustful of third party applications due to fears of viruses and malware.

Secondly, the system requires users to pair their devices with it before transfers can take place. This is undesirable in our context because we will be using static displays, meaning the display cannot interact with users with regard to the exchanging of passkeys, a necessity of pairing. Secondly, requiring devices to pair

## Chapter 2 - Background

opens up the user's device to exploitation via methods such as a man in the middle attack [39] as well as the fact that paired devices have unrestricted access to data on the other device. Thirdly, this, along with the fact that users must be able to touch the screen to interact with the system, limits the use of the system (with regards to downloads) to one user at a time. Lastly, the fact that users are required to have access to the touch screen introduces security concerns as it would be challenging to prevent theft, especially in the communities where the system would likely be deployed.

### 2.1.2 ContentCascade



Figure 5 - A user interactive with a ContentCascade display [34]

ContentCascade was developed by Raj et al [34] to specifically investigate the question of how to make displays interactive.

A client-side application was required on the user's mobile phone and was used to interact with public display, such as those one might find in a movie theatre, displaying trailers of upcoming movies.

A user could, using the app on their mobile device, initiate contact with the display, and subsequently download the trailer currently being displayed, as well

## Chapter 2 - Background

as other information such as show times, movie reviews and discount coupons onto his or her mobile device.

Because the system allows users to interact with the display without needing to touch it, more than one person can interact with it simultaneously. Additionally, the screen can be placed in a more secure location such as behind security glass or high up on a wall to prevent theft. The requirement of a client-side application, however, makes this unsuitable for our use, since we require the system to use only the functionality available on a typical feature phone.

### 2.1.3 WebWall



Figure 6 - A WebWall display showing a video, picture, advert and poll [17]

WebWall, by Ferscha and Vogl [17], makes use of multiple protocols to allow users to interact with a large screen display. A user with a mobile device can use email, SMS, MMS or WAP to send media to the display. Alternatively, they may log in to the backend using the web interface provided.

Users may submit notes, videos or pictures. They can also take part in auctions and opinion polls, all of which are displayed dynamically on the screen.

Since interaction from a mobile device is limited to SMS, MMS and email, no client-side application is required. Users are, however, unable to download anything from the display onto their mobile device.

As with the Hermes display, the non-requirement of a client-side application is something we wish to emulate in our system.

### 2.1.4 CityWall



Figure 7 - The CityWall display with Flickr photos [33]

CityWall, developed by Peltonen et al [33], had no capacity for the uploading or downloading of media to/from the system. It comprised a large multi-touch interface, installed on a shop window on a busy street. Photos of the city (Helsinki, Finland) were downloaded from Flickr and displayed. Users could interact with the display by dragging, rotating and resizing images on the screen. Since the screen was large (2.5m wide), and supported multi-touch, it was possible for several people to interact with the system simultaneously. The study found that this parallel use produced differing usage patterns. Some groups cooperated to organize images together, whilst, on other occasions, two separate groups, with differing goals, would come into conflict over the use of the available space.

This conflict over use of the system is something that we wish to avoid in the design of our system. Additionally, this set up is not suited to our system due to the fact that it allows no interaction via the users' mobile devices. Users can neither upload nor download content to or from the screen. They also need to be able to touch the screen, which is undesirable for reasons discussed in the section on the Hermes system.

### 2.1.5 Publix



**Figure 8 - User interacting with a Publix display [41]**

Publix was developed by Ventura et al [41]. A large screen is used to display images, news and advertisements. Users can interact with the system by means of a client-side application. The application allows them to upload and download images to/from the system but it also allows more complex interaction, such as the playing of games or downloading of promotional media.

Of particular interest is the manner in which the system entices users to interact with it. It constantly scans the nearby area for visible Bluetooth devices. On locating a device, it will send an invitation to use the system. To avoid unwanted spam, the system will note which devices it has invited, and not send subsequent invitations.

Included in the invite are instructions as to how to obtain the client-side application. This can either be downloaded over GPRS (thereby incurring data costs to the user) or by sending a blank (black) image, via Bluetooth, to the system. On receiving an image that is black (or mostly black), the system will respond by sending the client-side application, via Bluetooth, to the user. This invite system is not something we will employ on the first iteration of design, but is something that could be looked at in future iterations.

### 2.1.6 Original Snap 'n Grab System



Figure 9 - A Snap 'n Grab display. The eight images each represent a media pack [30]

A Snap n Grab system allows users to download media onto their mobile phones. Media can be images, videos, music files, calendar events or contacts. The media is organized into media packages. The media packages are represented by an iconic image, which are displayed in some manner, usually on a large screen or on posters. Users retrieve media by taking a picture of the icon representing the package they desire. This picture is then sent, using Bluetooth, to the server. Image recognition algorithms are run on the image received in order to determine which media pack the picture represents. The files in that media pack are transmitted to the user, again over Bluetooth.

This system allows users to retrieve desired media at no cost to them using their feature phones. It also requires no third party software to be running on the users' phones, as all they require is a camera and the ability to send and receive files over Bluetooth, standard features on feature phones.

A later version of the Snap 'n Grab system allows users to create their own media packages and upload them to the system. This allows communities to exchange media with each other, using the Snap 'n Grab system as an intermediary.

The original Snap 'n Grab system was developed by Maunder et al [30] and was deployed in Khayelitsha, a township in Cape Town, South Africa. It was comprised of a Mac Mini acting as a server and utilized a 42-inch LCD monitor to display the media packs.

Image recognition was done using the SIFT algorithm developed by Lowe [26].

## Chapter 2 - Background

As well as providing a means for users to retrieve media, the original Snap 'n Grab system leveraged the dynamic display offered by the LCD monitor to allow users to create their own media packs by uploading custom content to the board. Users could create their own packs by sending a v-Card from their mobile phone to the system. A new slot would appear on the screen, after which each subsequent item would be added to that slot.

The system was deployed for a period of about 3 months, during which time its usage was monitored. Some interesting outcomes were noted, such as the fact that the most prominently exchanged media were videos taken of ladies' choir practices.

Two other projects have been run using the original Snap 'n Grab system. The first was an HIV education project, which used the board to educate people about HIV and AIDS [18].

The second was called ICT4Democracy [18], where the board was used to aid voter education. Since it is often difficult for people in developing contexts to read up about various political parties, and what they stand for, party spokespersons were asked to create a short video to act as their manifesto. These were then placed on the board, and people were able to take a photo of the political party logo that interested them, after which they would be sent the relevant video via Bluetooth.

The Snap 'n Grab system is well suited to our task. It allows free transfer of media, by using Bluetooth. Users do not require any specialized client-side applications on the mobile device and, because users do not need to touch the screen at any point in the interaction, the system could be adapted to use static, paper displays instead of a large LCD.

There are, however, potential problems which must be addressed. Snap 'n Grab uses the SIFT algorithm to carry out image recognition. This algorithm will not run efficiently on a mobile device, as demonstrated by Bay et al [4]. Thus, an alternate image recognition technique will need to be developed.

## 2.2 Bluetooth

### 2.2.1 Bluetooth capabilities

Bluetooth is a wireless communication technology enabling the transmission of data over short distances. It operates in the 2.4GHz radio frequency band. Files can be exchanged by devices using Bluetooth via the Object Exchange (OBEX) protocol. Most mobile devices with Bluetooth capabilities support OBEX. (A notable exception is the Apple iPhone). Since most phones support OBEX by default, this means that users will not need to install any special software on their phone in order to make use of the system.

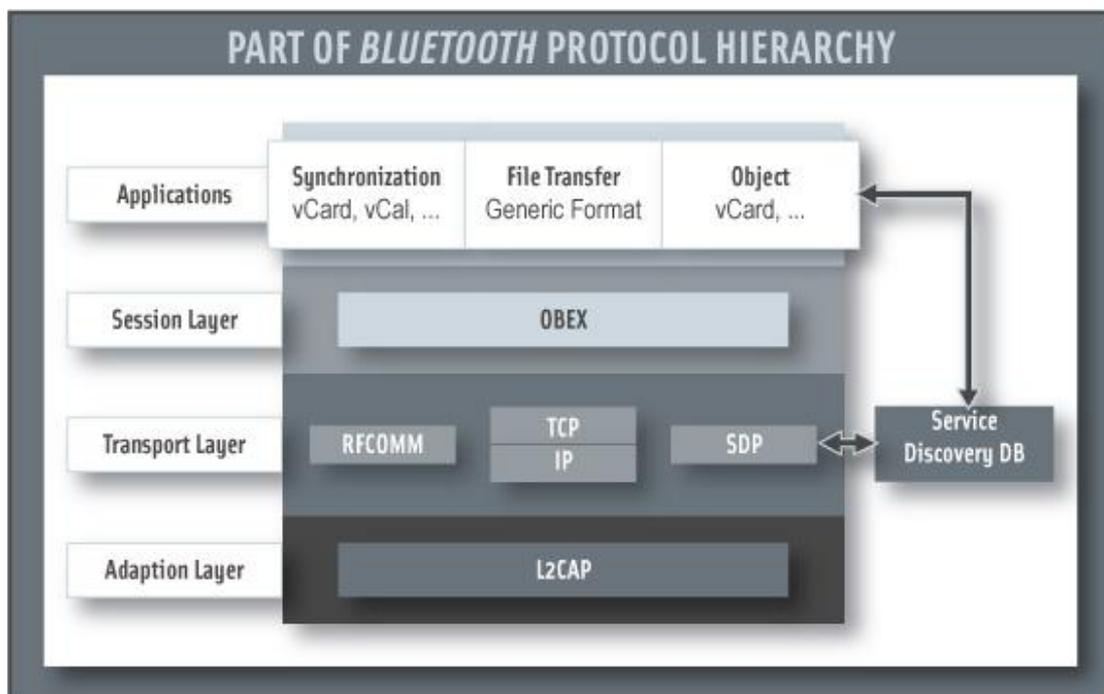


Figure 10-The Bluetooth protocol [7]

A Bluetooth device can communicate with up to 7 other Bluetooth devices at any one time [8].

## 2.3 Mobile Image Recognition Projects

The Snap 'n Grab system was able to leverage the power of the desktop computer which allowed it to use a computationally expensive image recognition algorithm. Since our system will be deployed on a mobile device, we must investigate a more computationally efficient method of determining which image

has been photographed. This section is an investigation of projects incorporating image recognition involving mobile devices.

### 2.3.1 Interactive Museum Guide

Bay et al. developed an Interactive Museum Guide [4]. The system ran on a table PC and made use of the SURF image recognition algorithm to process photographs taken of museum artifacts to determine their identity. They reported processing times of at least 42 seconds, rising to 78 seconds, depending on which version of SURF was implemented. These results were obtained on a Linux Tablet with a 1.7GHz Intel Pentium M processor. Additionally, they tested the SIFT image recognition algorithm on the system and found it to have inferior performance to SURF, with a processing time of 219 seconds, far higher than our accepted maximum time of 2.25 seconds.

### 2.3.2 CD Cover Recognition

Tsai et al developed a system that allows users to take a photo of a CD cover, using their mobile phone, and get information related to that CD [40]. The system involved a client application, installed on the user's phone, and a remote server which processed the images. Images taken were sent over a 3G network to the server, where the SURF image recognition algorithm was run to determine the CD that had been photographed. The resulting information is then relayed, again across the 3G network, back to the user's mobile device. They reported results of between 8 and 11 seconds for transferring and processing the image.

### 2.3.3 Discussion

The Interactive Museum Guide used online image recognition, image recognition done on the same device used to take the photograph. The main problem with online recognition is that it requires a client application to be installed on the user's mobile device. Additionally, the processing times reported for the system were far higher than those we are aiming at for our system.

The CD cover recognition system used offline processing: processing was carried out on a remote server. However, the system used a 3G network to transfer the image to the server for processing, thereby incurring network data costs. Once

again, the time taken for transfer and processing was higher than our accepted maximum.

### 2.4 Barcodes

As the previous section demonstrates, the current state of the art in mobile image recognition is not at the point where photographs can be efficiently analysed to determine what has been photographed. Thus we need to look at alternative methods of recognition. Barcode reading on mobile phones is a mature problem, with many accepted solutions. We shall investigate 1D and 2D barcode systems and discuss what is applicable to our system, and what is not. Further, we shall discuss what can be adapted to create a hybrid system, leading to both descriptive posters and efficient recognition.

#### 2.4.1 1D Barcodes

The concept of a barcode was developed by Bernard Silber and Norman Joseph Woodland in 1948 in response to the problems large retailers were having with keeping track of their massive inventories [2]. The first barcode reader was the size of a desk and had to be wrapped in black cloth to avoid ambient light disrupting the reading. By 1970 the technology was mature enough to be seriously considered by the retail industry and a study by McKinsey & Company predicted industry savings of \$150 million per year through the use of a barcoding system [37].

Since 1974 refinements have been made to the Universal Product Code (UPC) developed in the 1970's and new standards such as the European Article Numbering system (EAN) have been developed.

There are now two types of barcodes, one-dimensional (1D) and two-dimensional (2D). 1D barcodes encode data on their horizontal axis only. 2D barcodes encode data on both the horizontal and vertical axes. Because of this, 2D barcodes can encode a much larger amount of data than 1D barcodes.

Another method of classifying barcodes is by the amount of data they store, and whether or not they are self-sufficient. A self-sufficient barcode does not need an external network or database to be fully functional, as it contains all the required

## Chapter 2 - Background

information within itself. This is called a database barcode. The alternative is an index barcode. This contains less data, usually just an index, which is then passed, along a network, to an external database where the rest of the relative information is stored. Index barcodes tend to focus on robust, reliable reading, taking into account the environment (lighting, etc) and the limitations of the equipment with which it will be used.

UPC is a 12-digit index barcode used in many retailers in North America and around the world. It comprises 11 digits, and one check-digit.

### Theory of UPC barcodes

Each of the digits in a UPC barcode is made up of 7 bars. A bar is either “on” (black) or “off” (white).

The seven bars are arranged in such a way that a single digit will contain 2 “on” blocks and two “off” blocks.



**Figure 11 - A UPC barcode. Note the left, middle and right guard patterns and the odd and even parity used to encode the 0 digit on the left and right hand sides respectively**

A UPC barcode (see Figure 11) is made up of the following:

- A left “guard pattern”, which is 101.
- Six digits encoded with odd parity. The first of these six digits denotes the industry type. The remaining 5 are the first 5 digits of the item code.
- A center “guard pattern”, which is 01010.
- Six digits encoded with even parity. The first 5 are the remaining 5 digits of the item code. The sixth is a check digit (see below for calculation).

## Chapter 2 - Background

- A right “guard pattern”, which is 101.

<b>Digit</b>	<b>L Pattern</b>	<b>R Pattern</b>
0	0001101	1110010
1	0011001	1100110
2	0010011	1101100
3	0111101	1000010
4	0100011	1011100
5	0110001	1001110
6	0101111	1010000
7	0111011	1000100
8	0110111	1001000
9	0001011	1110100

As described above, the 12 digit barcode is divided into 2 halves of 6 digits each. Each digit on the left hand side is encoded using odd parity, and each digit on the right hand side is encoded using even parity. This has the effect that the left hand digits are the inverse of the right hand digits (e.g. 0110001 becomes 1001110).

The last digit of a UPC barcode is called the “check-digit”. This is calculated as follows:

- All digits in odd positions (1, 3, 5, 7, 9, 11) are added together and the total is multiplied by 3.
- All digits in even positions are added and their total is added to the previous result.
- Let the new result be  $x$ .
- Let  $y = x \bmod 10$  (the remainder when  $x$  is divided by 10).
- If  $y$  is not 0 then let  $z=10-y$ , otherwise let  $z=y$ ;
- $z$  is the check-digit. This is appended to the barcode.

Laser scanners are widely used in retail stores to read UPC barcodes. These use orthogonal lines to ensure that the barcode is read.

### 2.5 2D Barcodes

As mentioned, because two dimensional barcodes are able to encode information on two axes, the amount of data they can store is greatly increased. This enables them to be used as database, or self sufficient, barcodes. Database barcodes contain all data required to carry out a task within themselves. They have no need for a connection to an external database.

There is no global standard for 2D barcodes and, as a result, many different forms have been developed.

#### 2.5.1 Examples of 2D Barcodes

We will now look at five of the most prominent 2D barcodes.

##### *QRCode*

QRCode was invented by Denso Wave in 1994 [23]. It has a maximum data capacity of 7089 characters for numeric data or 2953 bytes. It has position detection patterns located in 3 of the corners of the code. This enables it to be read from any angle. Reed-Solomon error correction can be used to restore the original data in the event that the symbol is damaged.

##### *Data Matrix (Semacode)*

Data Matrix can encode up to 1556 bytes of data [23]. It uses a finder pattern consisting of solid lines along the left and bottom of the code, and broken lines along the right and top of the code.

##### *VisualCode*

VisualCode was developed by Rohs et al. [36] to enable mobile phones to interact with the physical world using barcodes.

The system uses the notion of guidebars for positioning and can be used to detect phone movement. Each code has its own coordinate system, mapped out by the guidebars and cornerstones.

Only capable of encoding 83 bits of data, VisualCode is generally used as an index barcode system.

We will discuss VisualCode further in Section 2.5.4.

### Shotcodes

Shotcodes [27] were developed by High Energy Magic (as SpotCodes). They use a circular barcode system to encode a single number. This number can be between 1 and 19683 ( $3^9$ ). The number is encoded in two concentric rings. This is also an index barcode system.

### ColorCode

ColorCode was developed by Han et al [23]. It uses colors to encode 10 digits in a 5x5 matrix. Because the color values detected by a camera change in different lighting conditions, reference cells are used to normalize the readings. Because only 40% visibility for each cell is required for decoding, it is possible to integrate ColorCode into logos and adverts (see Figure 12).



Figure 12 - 2D barcoding systems. From top left: QR Code, DataMatrix, VisualCode, ShotCode, ColorCode, ColorCode incorporated into logo

### 2.5.2 Applications of 2D Barcodes

It should be noted that the primary use case for a lot of these 2D barcodes is for use with mobile phones. As such they have been designed to take into account the lower camera quality, freedom of orientation and lack of stability that is incumbent with these devices. The uses of these barcodes have also been influenced, with most of the scenarios involving the barcode being used either to

## Chapter 2 - Background

add information to the phone, such as a phone number or a URL, or as a portal to a website giving more information on a competition, movie, book, etc.

Examples include:

### *Movie posters*

2D barcodes placed on movie posters can act as portals to more information, such as movie websites, age restrictions or show times.

### *Adverts in magazines*

In a similar manner, 2D barcodes placed in magazines also have the ability to act as portals to a wide variety of information. These include special offers, websites, social networking sites, etc. Popular culture magazines can put a 2D barcodes next to a picture of a celebrity which, when photographed, allows users to download a high quality version of that picture onto their mobile phone to use as a wallpaper.

### *Business cards*

A 2D barcodes placed on a business card can make adding a contact on a mobile phone for the person represented by the card simple. The user would simply take a photo of the card and the software on their phone would decode the 2D barcodes and retrieve the requisite information, such as name, phone numbers, addresses, etc.

### *Augmented Reality*

2D barcodes placed on everyday objects can allow users with the requisite software on their mobile device to interact virtually with that object. They could obtain more information about a DVD in a store, for example or, using an orientation aware barcode such as VisualCode, display a 3D model of a picture in a book or magazine on the screen of the mobile device. The 3D model could be rotated and zoomed by changing the orientation of the barcode, either by rotating the book or the mobile device.

### **2.5.3 Problems with 2D barcodes related to our situation**

2D barcodes provide an efficient means of encoding data which people can retrieve using their mobile phones. There is, however, one main reason why 2D

## Chapter 2 - Background

barcodes are not applicable to a mobile Snap 'n Grab system, 2D barcodes are not descriptive:

It is not possible to tell, simply by looking at a 2D barcode, what the data encoded in it is related to. Since the purpose of a Snap 'n Grab system is to allow users to choose content of interest to them, and download it, it requires that the image they are taking a picture of is descriptive of that content.

A seemingly intuitive solution would be to place a 2D barcode within the descriptive image specified above. The problem with this, however, is that the majority of 2D barcode decoding algorithms require that the barcode be, for all intents and purposes, the only item in the photo. Thus, if a user were to take a photo of the entire picture, the decoding algorithm would likely fail.

Similarly, since users who have used 2D barcodes before would have become used to photographing only the code, our solution needs to make it clear that they need to photograph the entire poster.

### 2.5.4 Visual Code by Rohs

Rohs et al have done a large amount of work into interactivity with mobile phones [35, 36]. They have developed a means of using a mobile phone camera as a sensor for interacting with real world objects using two-dimensional visual codes. An example of their system in action is pointing the camera at a certain visual code, causing the mobile phone to automatically dial a certain number. The codes could be printed on business cards or in phone directories.

The algorithms have been developed for mobile phone use and have the following criteria:

#### *Minimal floating point operations*

The mobile phones Rohs developed for were severely limited in terms of processing power. Since they have no hardware floating-point processor, all floating-point operations have to be emulated by the CPU. Operations carried out in this manner are extremely inefficient and therefore should be avoided.

### *Coarse code features*

Rohs concerned himself with images taken in VGA (640x480 pixels) or lower resolution. Therefore all features used by the code needed to be much coarser than those used by many modern shot code algorithms.

### *Colour not used as a code feature*

Rohs discovered during testing that colour should not be used as a code feature due to the large differences in colour values caused by different mobile phone cameras and lighting conditions.

### *Ability to decode rotated/tilted images*

Since mobile phones are not constrained in their movement or positioning the decoding algorithm needs to be able to read codes that have been arbitrarily rotated or tilted.

These criteria were used to design the code layout pictured below. It incorporated two guide bars, one on the right and one on the bottom of the code, and three cornerstones, positioned at the top left, bottom left and top right of the code. The guide bars were used to correctly orient the code, whilst the cornerstones were used for distortion correction.

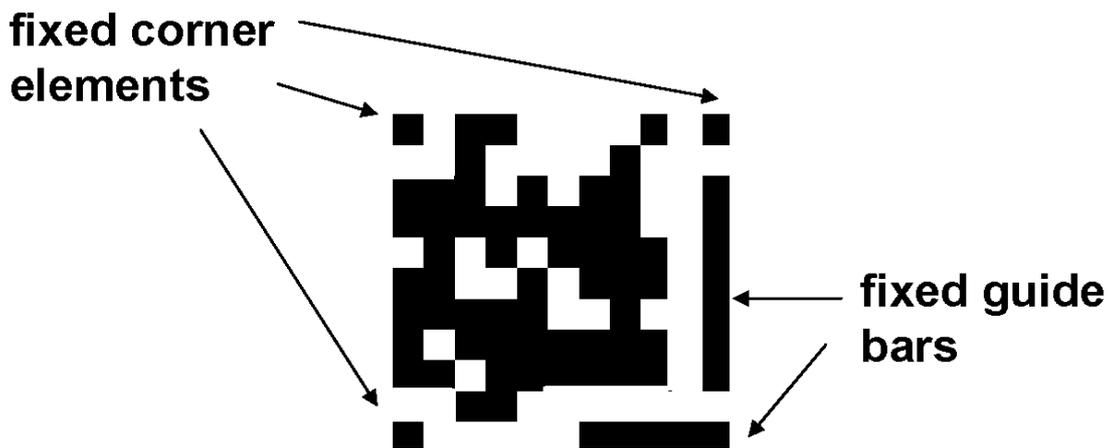


Figure 13 - VisualCode, showing the guidebars and corner elements.

By adapting the system and placing a guide bar on the left and right hand side of the image representing our media packs, we are able to efficiently locate the barcode in the photo to be decoded.

## 2.6 Summary

In this chapter we have looked at previous work which can inform the design of our media distribution system.

We investigated public displays, and found that the original Snap 'n Grab system was the most suited to our situation, since it requires no client side application and uses Bluetooth for media transfer, thereby incurring no cost to the user.

Because we require that our system responds in the same time as the original Snap 'n Grab system the process of image recognition needed to be looked at. Existing mobile image recognition solutions were found to be too slow, and thus an alternate solution – barcodes – was investigated.

Popular mobile barcode systems were discussed, after which the VisualCode solution was found to be the most suited to our task.

## 3. System Design

### 3.1 Introduction

This chapter describes the Snap 'n Grab Lite system as a whole. The system is made up of a number of components, namely posters, media packs, media pack generators and the server software. Each of these components is discussed individually in this chapter.

### 3.2 Hardware and Platform

#### 3.2.1 Windows Mobile

Windows Mobile was chosen as the platform to develop the system for. The reason for this was twofold:

1. The original Snap n Grab system was developed using the C# programming language and the .NET framework. Applications for Windows Mobile devices can be developed using the .NET Compact Framework, which is a member of the .NET family. This makes porting code from one to the other a much simpler process than it would be to port to any other existing mobile platform.
2. At the time of starting the project the HTC Touch Pro, which runs on Windows Mobile 6.5, was one of the most powerful mobile devices available.

Windows Mobile Professional 6.5 was released in 2009 for smartphones with touch screens.

#### 3.2.2 Phone Hardware

When choosing a device to develop certain criteria were considered of utmost importance.

The device should have:

- Windows Mobile as an operating system (for the reasons listed above)
- a fast processor (above 500MHz preferable) to handle complex image processing tasks.

## Chapter 3 - System Design

- enough memory to be able to process multiple images simultaneously.
- Bluetooth capabilities.
- 3G and Wi-Fi capabilities. A future plan for the system is to allow media packages to be pushed to the phones acting as servers. For this to happen they will require the ability to connect to fast broadband networks such as 3G or Wi-Fi.

The HTC Touch Pro has the following specifications:

- Windows Mobile 6.5
- 528MHz Qualcomm processor
- 288MB RAM
- Bluetooth
- 3G and Wi-Fi

For these reasons it was chosen as the device to develop the system on.

### 3.3 Poster Design

The posters need to fulfill two functions:

1. They need to portray to users an idea of what data they will receive should they photograph the poster
2. They need to contain a barcode so that the Snap 'n Grab Lite system can determine which poster the user has photographed.

The pictorial aspect is handled by the user generating the media packs in that they are allowed to choose a representative image to use as the media pack's icon. (See Section 3.4 for more info about media packs and their generation).

#### 3.3.1 Barcode

The initial barcode used was an adaption of the UPC barcode standard.

Our barcode is made up as follows:

- Left guard bars, three bars of equal width, marking the start of the barcode.
- Five digits representing the media pack code.
- One check digit.

The digits are created using the UPC standard.

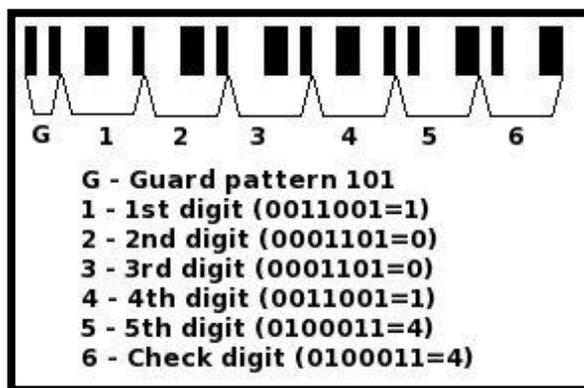


Figure 14 - Diagram of adapted UPC barcode used on the posters

The barcode is placed on the top of the image and mirrored on bottom of the image. This will allow the image to be photographed upside down, should the need arise.

Guidebars, similar to those used by Rohs, were placed on the left and right of the image. These are the most important aspects in the image recognition process. The system required that the algorithm was able to find these guidebars correctly so that the barcode could be read.

### 3.3.2 Target Lines

The target lines in the four corners of the poster were placed there to aid users in knowing what they needed to photograph. They mimic the standard framing lines used in many cameras.



Figure 15 - A poster showing barcode, guidebars and target lines

### 3.4 Media Packs

A media pack is a collection of files represented by a poster. It is the collective term for the files that will be returned by the system to a user should the photograph a particular poster.

#### 3.4.1 Layout

The media packs are located on the HTC Touch Pro. They are organized as follows:

Each pack is located in a folder named as follows: <name> + <code>.

Where <code> is the number encoded in the barcode and <name> is human readable name given to the media pack when it is created. E.g. "Worldcup + 10045".

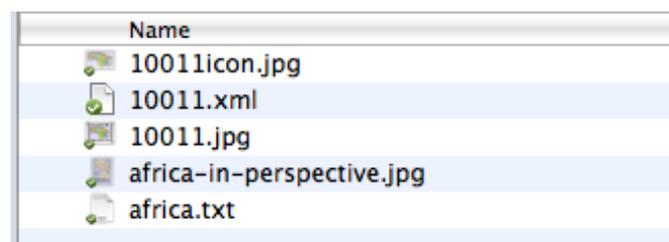


Figure 16 - Typical contents of a media pack folder.

Inside the folder is an xml file called `<code>.xml`. This is set out as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<snapandgrab>
  <object>
    <id>Media Pack Code</id>
    <title>Media Pack Title</title>
    <textbody/>
    <itemCount>Number of items in pack</itemCount>
    <media>
      <item>Item 1 filename</item>
      <item>Item 2 filename</item>
      ...
      <item>Item n filename</item>
    </media>
  </object>
</snapandgrab>
```

Figure 17 - Xml file describing a media pack

Also in the folder are the files `<code>.jpg` and `<code>icon.jpg`. `<code>.jpg` is the poster image, including the barcode, which is printed out.

`<code>icon.jpg` is the pictorial part of the poster, minus the barcode, which is used in the two media pack generator programs for display purposes.

Lastly, the remaining files in the folder are those listed in the `<code>.xml` file between the `<media>` tags. These are the files that are returned to the user.

Taking Figure 17, above, as an example, if the user had taken a photo of the poster containing `10011.jpg` and sent it to the system they would receive `Africa-in-perspective.jpg` and `Africa.txt`.

### 3.4.2 Generation of Media Packs

In order for the Snap 'n Grab system to be used, it must be possible to easily and efficiently create media packs. For this reason, two applications were developed to facilitate the creation and management of media packs, one for desktops and one for Windows Mobile.

## Chapter 3 - System Design

The functionality of the two applications is essentially the same, with only the interface and workflow differing to accommodate the different devices.

As these applications were created purely for the researcher to create sample media packages, and not for end users, we will not carry out usability evaluations on these.

The mobile version was developed using the .NET Compact Framework and was deployed on the HTC Touch Pro.

Users were able to name the media pack, give it a code or let the system generate one automatically, choose the pictorial icon for the media pack and add files to the media pack.

Below are screenshots depicting the workflow for creating a media pack using the mobile media pack generator:



Figure 18 - Screenshots of the workflow followed on the mobile media pack generator

## Chapter 3 - System Design

The desktop version is not constrained by the small screen of the mobile and hence can incorporate all the elements into a single screen, allowing for a more flexible workflow:

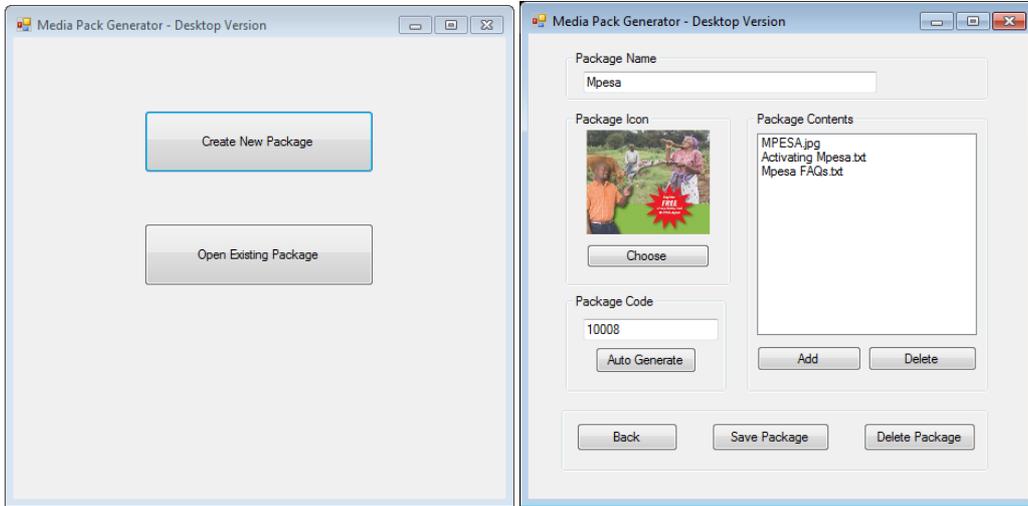


Figure 19 - Screenshots of the desktop media pack generator

### 3.5 Snap 'n Grab Lite Server System

The Snap 'n Grab Lite system was designed to use the same workflow as the original Snap 'n Grab system, namely:

- Wait for image to be received
- Save image
- Process image to get barcode
- Return files to user

Since our system was designed such that users did not have to install any extra software on their phones, image recognition could not be carried out on their devices. It was, instead, carried out on the server phone (the HTC Touch Pro).

An alternative to this would have been sending the image from the server phone to a powerful server somewhere in the cloud for processing. This option was decided against for two reasons:

## Chapter 3 - System Design

1. It would have taken longer for the image to be sent to the server in the cloud than it would to process the image on the phone.
2. Mobile internet access is not ubiquitous, especially in developing areas. Therefore this would not be a globally acceptable solution.

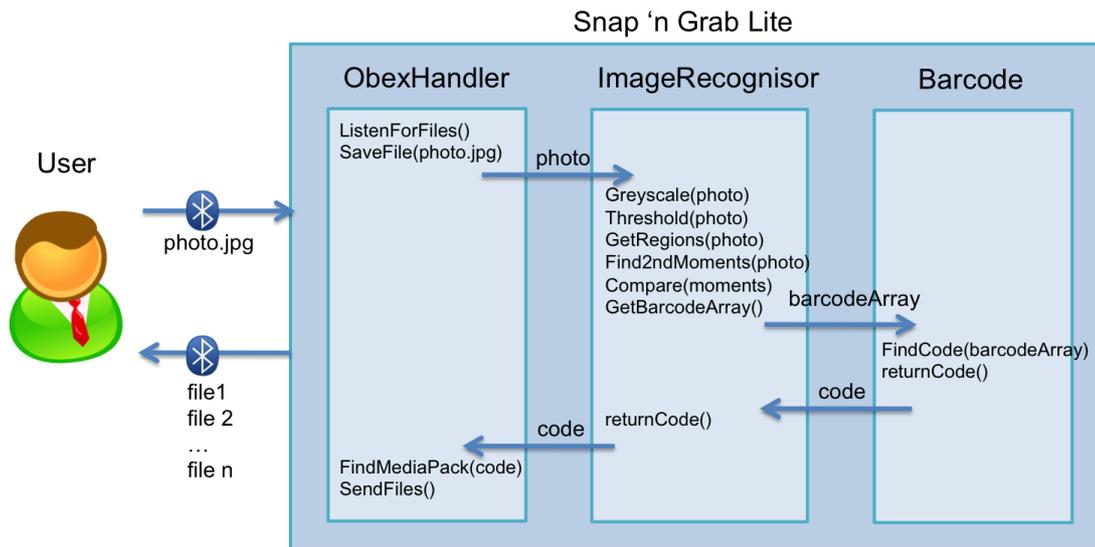


Figure 20 - Overview of system use

### 3.5.1 Bluetooth

For a fuller discussion of Bluetooth technology please see the Bluetooth section in the Literature Synthesis (Chapter 2).

#### *32feet.NET libraries*

The 32feet.NET libraries, developed by In The Hand Ltd., allow for easy development of personal networking applications using the .NET framework. It supports Bluetooth and Infrared technologies.

#### *HTC Bluetooth Bug and Solution*

A problem was encountered in using the 32feet.net libraries with the HTC Touch Pro in that there was a rogue OBEX Protocol operating on the device which prevented our application from intercepting incoming files. Instead, files either failed to send (the users received a “Connection failed” message on their mobile phone), or were handled by the operating system of the HTC TP phone in the usual manner (as if the system was not running).

The solution to this problem was to edit the registry of the phone. The value for `isEnabled` in `HKEY_LOCAL_MACHINE\Software\Microsoft\Obex` was changed from 1 to 0. This was found to turn off the troublesome OBEX protocol which allowed the system to correctly intercept incoming Bluetooth messages.

### *Receiving the image*

The system waits for incoming Bluetooth messages. Once a message is received, the Bluetooth ID of the device sending the image is taken note of. If it is the first time that device has submitted an image to the system a new folder is created with the Bluetooth ID as the folder name. The image is saved in this folder.

### *Sending the files*

Once the barcode has been decoded the files are returned to the user using the Bluetooth ID that was saved when the image was submitted. The xml descriptor file for the media pack is opened and each of the files listed is sent to the user.

If the barcode cannot be decoded, or if the code returned does not match any of the media packs present on the device, then a default “Pack not found” package is returned to the user. This contains pictorial instructions describing how to use the system and how best to photograph the poster to ensure the system is able to locate and decode the barcode.

### **3.5.2 Image Recognition Algorithm**

Once the image has been received, it needs to be processed so that the barcode can be located and decoded.

In order to locate the barcode the guidebars need to be found. The method used is an adaption of that developed by Rohs et al. To locate the guidebars the image is converted, first into grayscale, then into black and white. The black and white image is divided into individual regions of neighbouring black pixels. These regions are analysed to determine which of them are the guidebars.

Once the guidebars are found a scanline is drawn between them, through the barcode, which can then be read by reading off the pixels on the scanline.

The following steps are carried out:

### Grayscale

The RGB values for each pixel are converted to a gray value by averaging the green and red values for each pixel.

$$g_i = \frac{R_i + G_i}{2}$$

This follows the method used by Rohs, omitting the blue value since it has the lowest quality in terms of sharpness and contrast.

### Thresholding

A simple thresholding algorithm was originally employed to convert the image from grayscale to black and white.

The average pixel gray value was calculated:

$$\bar{g} = \frac{\sum_{i=1}^n g_i}{n}$$

where  $g$  is the gray value of each pixel.

Each pixel with a gray value below  $\bar{g}$  was set to black (0) and every pixel with a gray value above  $\bar{g}$  was set to white (255).

This is a very efficient method of thresholding an image, but it is also a very naive method. If there are variances in lighting across the image this will lead to the poor black and white images. A shadow across a portion of the image can lead to the black and white image being reduced to a black half and a white half due to the fact that only the global average is taken into account.

A better, yet more computationally expensive, method is an adaptive thresholding algorithm, which was implemented after the first round of user experiments. This algorithm is discussed in more detail in Chapter 4.

### Region Map Generation

The region map is a 2D array of the same size as the image. Each cell represents a pixel. The value is an index representing the region the pixel belongs to. All adjacent black pixels should have the same region value in the region map. What we are aiming to do here is to create a table of every region in the image, with

## Chapter 3 - System Design

information about each region. From this information we will be able to determine which two regions represent the guidebars on the left and right hand side of the image.

The region map is generated using a two-pass process.

In the first pass each black pixel is compared against the one above and to the left of it. If they are black, then it is considered to be part of the same region as them, otherwise it is considered to be the start of a new region.

After the first pass we have all black pixels assigned to a region but some adjacent regions may have different "region values". These occasions would have been noted in the equivalence table. The second pass of the algorithm runs through the pixels and resolves these equivalences.

This is implemented as follows:

```
int[,] regionMap = new int[height, width];
List<int>equivTable = new List<int>();
intregionCount = 0;

for (int j = 0; j < height; j++)
for (int k = 0; k < width; k++)
    {
        // If it's not a background pixel
        if (bwImage[j, k] == 0)
            {
                boolnewRegion = true;
                // If it has pixels to its left
                if (k > 0)
                    // If pixel to left is black
                    if (bwImage[j, k - 1] == 0)
                        {
                            // It's part of the same region as pixel to left
                            regionMap[j, k] = regionMap[j, k - 1];
                            newRegion = false;
                        }
                // If it has pixels above it
                if (j > 0)
                    // If above pixel is black
                    if (bwImage[j - 1, k] == 0)
                        {
                            // If it's not connected elsewhere, then same region as pixel above
                            if (newRegion)
                                {
                                    regionMap[j, k] = regionMap[j - 1, k];
                                    newRegion = false;
                                }
                        }
            }
    }
```

## Chapter 3 - System Design

```
        // If it is connected elsewhere, check if the regions are different
        else if (regionMap[j - 1, k] != regionMap[j, k - 1])
        {
            associate(equivTable, regionMap[j - 1, k], regionMap[j, k - 1]);
        }
    }
    // If not connected, start a new region
    if (newRegion)
    {
        regionCount++;
        regionMap[j, k] = regionCount;
        equivTable.Add(regionCount);
    }
}
```

### The Associate method:

```
// Makes two regions into 1 by recursively associating them
private void associate(List<int>equivTable, int a, int b)
{
    if (a > b)
        associate(equivTable, b, a);
    else
    {
        if (equivTable[b] == a)
            return;
        if (equivTable[b] == b)
            equivTable[b] = a;
        else
        {
            associate(equivTable, equivTable[b], a);
            if (equivTable[b] > a)
                equivTable[b] = a;
        }
    }
}
```

### The Second pass:

```
for (int j = 0; j < height; j++)
    for (int k = 0; k < width; k++)
        if (regionMap[j, k] != 0)
            regionMap[j, k] = resolve(regionMap[j, k], equivTable);
```

**The Resolve method:**

```
// Removes redundant regions
private int resolve(int value, List<int>equivTable)
{
    if (equivTable[value] == value)
        return value;
    else return resolve(equivTable[value], equivTable);
}
```

At the end of this process, all neighbouring black pixels will have the same region values.

*Second order moment calculation*

Each region can now be analysed, and its second order moment calculated. This is done by finding the centre of gravity for each region and then finding the eccentricity in the x, y and xy directions.

The center of gravity for a region R is given by  $(\bar{x}, \bar{y})$  where

$$\bar{x} = \frac{1}{|R|} \sum_{(x,y) \in R} x$$

and

$$\bar{y} = \frac{1}{|R|} \sum_{(x,y) \in R} y$$

The second order moments  $\mu_{xx}$ ,  $\mu_{yy}$  and  $\mu_{xy}$  are calculated as follows:

$$\mu_{xx} = \frac{1}{|R|} \sum_{(x,y) \in R} (x - \bar{x})^2$$

$$\mu_{yy} = \frac{1}{|R|} \sum_{(x,y) \in R} (y - \bar{y})^2$$

$$\mu_{xy} = \frac{1}{|R|} \sum_{(x,y) \in R} (x - \bar{x})(y - \bar{y})$$

These moments are then used to calculate the equation of an ellipse with the same major and minor axes as the region.

## Chapter 3 - System Design

The ellipse, E, is given by

$$E = \{(x, y) | dx^2 + 2exy + fy^2 \leq 1\}$$

Where

$$\begin{pmatrix} d & e \\ e & f \end{pmatrix} = \frac{1}{4\mu_{xx}\mu_{yy} - \mu_{xy}^2} \begin{pmatrix} \mu_{yy} & -\mu_{xy} \\ -\mu_{xy} & \mu_{xx} \end{pmatrix}$$

The angle of rotation of the ellipse is given by

$$\theta = \frac{1}{2} \arctan \left( \frac{2e}{d-f} \right)$$

The major and minor axes, r and s, are calculated as follows:

Let  $M = \cos(\theta)$  and  $N = \sin(\theta)$

$$\text{Now } r = \sqrt{\frac{-2e}{4MN} + \frac{d}{2} + \frac{f}{2}}$$

$$\text{And } s = \sqrt{\frac{2e}{4MN} + \frac{d}{2} + \frac{f}{2}}$$

The ratio ( $\delta$ ) of these two axes is calculated:

$$\delta = \frac{s}{r}$$

### **Guidebar Location**

This is done for each region and  $\delta$  is compared to the expected value for the guidebars. As each region is analysed a list of the two regions with ratios closest to that of the original guidebars is maintained. After all regions have been calculated the two regions left in this list are considered to be the guidebars. The region with the lower  $\bar{x}$  value is considered to be the left guidebar and the region with the higher  $\bar{x}$  is considered to be the right guidebar.

### **Barcode Reading**

A scanline is run from the top of the left guidebar to the top of the right guidebar and the pixel values are placed into an array.

## Chapter 3 - System Design

Since we have already calculated  $\bar{x}$ ,  $\bar{y}$  and the angle of rotation for each region it is possible to calculate a point towards the top of each guidebar between which a scanline can be run to read the barcode.

The point a, as shown below, on the left guidebar, is found by

$$(x, y) = (\bar{x} + \sin(\theta) * l * \frac{h_1}{h}, \bar{y} - \cos(\theta) * l * \frac{h_1}{h})$$

where l is the distance from the centre of the original guidebar to point a, h is the height of the original guidebar and  $h_1$  is the height of the region.

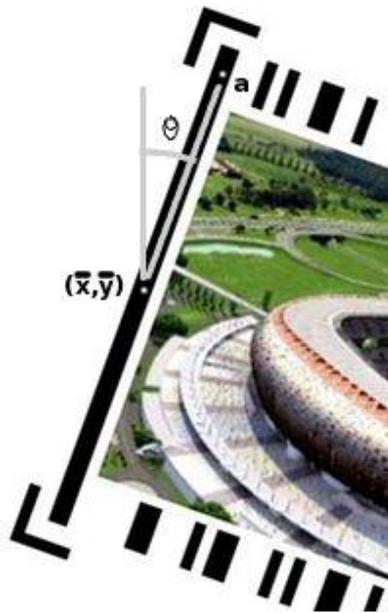


Figure 21 - Diagram showing how the starting point of the scanline is located

The respective point on the right guidebar is found similarly.

A line from a to b is traversed using Bresenham's algorithm and each pixel on the line is placed into an array. The array is then read to decode the barcode.

## Chapter 3 - System Design

The adapted UPC barcode starts with 3 bars, making up a guard pattern. Since the scanline starts inside the lefthand guidebar, there will be 5 color changes that are observed to get to the end of the guard pattern:

```
intlastPixel = pixels[0];
int pos = 1;
// Pixel at start of first black bar
intstart_pixel=0;
// Number of times the "run colour" has changed from black to white or viceversa.
intchangeCount=0;

// Locate the start of the barcode
while (changeCount<5 && pos<pixels.Length)
{
    // Has run colour changed?
    if (pixels[pos] != lastPixel)
    {
        changeCount++;
    }
    // Reset last pixel to current pixel and move onwards
    lastPixel = pixels[pos];
    pos++;
}
```

Now each digit is found. Since a UPC digit is made up of seven bars, with exactly four distinct areas, the array is read through until four changes of color are found. The middle pixel of each bar is then determined, based on widths of the areas and the total width of the digit. The seven bit string is then determined and decoded to give the digit.

This continues for the next five digits. The last digit represents the check digit, as laid out by the UPC standard (see Background chapter for full explanation).

Once the complete barcode is decoded the resulting number is passed back to the system. The respective media pack is looked up and returned, item by item, to the user.

## Chapter 3 - System Design

```
byte[] digits = new byte[42];
    //for each of the 6 digits (7 bars per digit, 4 discrete areas)
int j = 0;
while(j<6 && pos<pixels.Length)
    {
        // We're on the first pixel of the number
start_pixel=pos-1;
changeCount = 0;
pos++;
lastPixel = pixels[pos-1];
changeCount = 0;
        // go until changecount=4
while (changeCount< 4 && pos<pixels.Length)
    {
if (pixels[pos] != lastPixel)
changeCount++;
lastPixel=pixels[pos];
pos++;
    }
        // get average length of bar (total/7)
intlen = pos-start_pixel-1;
doubleaverage_bar = (double)len / 7.0;
        // get pixel in the middle of each bar
for(int k=0;k<7;k++)
digits[j * 7 + k] = (byte)(1 - pixels[(int)Math.Round(start_pixel-1 + average_bar / 2.0 + (k
* average_bar))]);
j++;
    }
```

### 3.5.3 Logging

The system creates log files for diagnostic purposes. For each image received a job ID is created so that the process can be tracked.

An example of typical job shows the entries in the log file:

```
10:03:16 Program started
10:03:18 Listener started
10:08:49 8 Creating thread
10:08:49 Request received, assigned jobID of 8
10:08:49 8 Device address: 0022FC56DCEC
10:08:50 8 Starting recognition
10:08:52 8 Code found: 901
10:08:52 8 Mediapackage found
10:08:52 8 Sending response files
```

10:08:54 8 Sending Athlone towers.mp4  
 10:09:00 8 Sending Free R10.txt  
 10:09:01 8 All items sent  
 10:10:46 Closing program

Table 2- Example system log

### 3.6 Summary

In this chapter we have described the design and implementation of the Snap 'n Grab Lite system. Below is a pictorial representation of the system as a whole. Showing the media pack generators as well as a user interacting with the system.

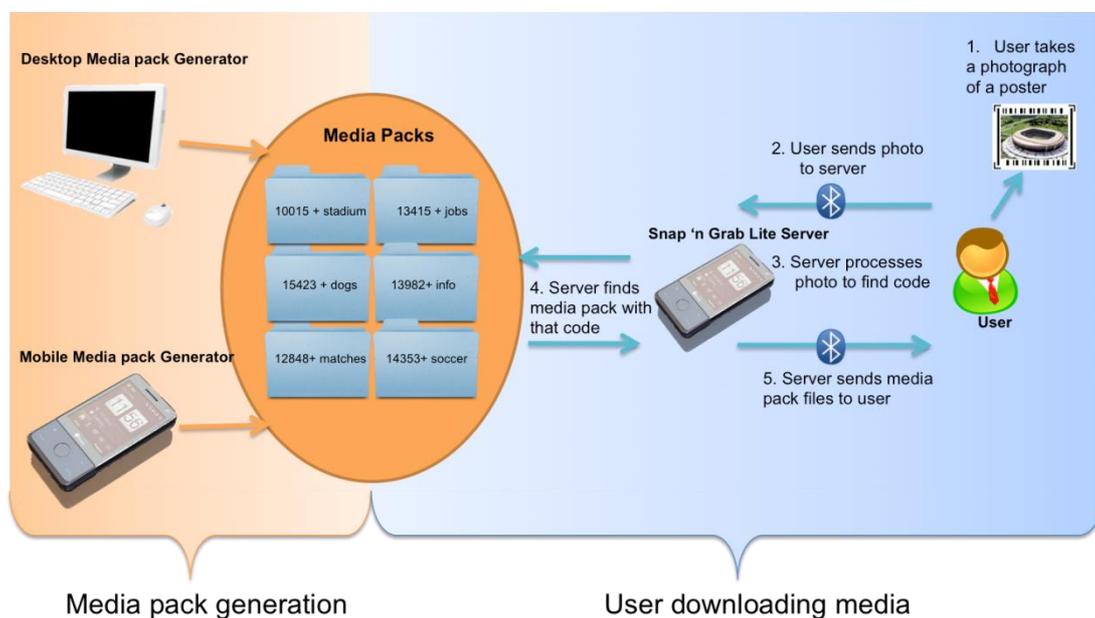


Figure 22 - Overview of the system

## 4. User Study 1

### 4.1 Experiment Round 1

The first round of experiments aimed to test whether the image recognition algorithm would work correctly when given photos taken by users as input.

As a source of sample images, four different posters were created. To determine the optimal image size, each was printed in five different sizes, ranging from 5cm to 20cm in width, and on both glossy and normal paper. These were then stuck on a wall in a laboratory (see Figure 23) where lighting conditions could be controlled to be one of ambient, direct and dappled lighting.

Whilst it would have been possible to take photographs ourselves and submit these to the system, it was decided that a more representative and unbiased sample would be gained by enlisting volunteers to photograph the posters.

#### 4.1.1 Design

##### *Hypothesis*

The image recognition algorithm is able to analyze images taken by users and match the target image, according to the barcode for the image.

##### *Task*

The experiment had two distinct phases: one in which images were captured by the users and another in which the researcher submitted those images to the algorithm.

Phase 1: The users were required to take ten photographs. The posters were grouped according to size and print-medium. There were five different sizes and two different print mediums, giving a total of ten groups. Users were asked to pick a poster at random from each of the groups to photograph. After each photograph the lighting conditions were changed. At the end of the experiment the photos were retrieved from the user's mobile phone to be used in Phase 2.

## Chapter 4 - User Study 1

Phase 2: The pictures taken by the users were input, one by one, into the image recognition algorithm to determine whether it could analyze them and match them correctly.

### *Variables*

Dependent variables:

Accuracy: Whether the algorithm outputted the correct code as embedded in the barcode..

Independent variables:

Poster medium: Glossy and matte posters will be tested.

Poster size: The size of the physical poster (which therefore affects the size of the barcodes)

Lighting: Three types of lighting were tested: ambient, directed and dappled.

### *Equipment*

The image recognition algorithm ran on an HTC Touch Pro.

The users made use of their own mobile phones to take the pictures. This is important since it allows us to gather a wide sample of both photography styles and pictures taken with different mobile phone cameras. Different mobile phone cameras use different shutter speeds, aperture settings and focal lengths, leading to differing pictures being taken. Therefore the power of the experiment would be improved by testing the system with a variety of cameras.

### *Venue*

The experiment was conducted in a closed room. There were blinds over the windows. This allowed us to control the lighting for each picture. The posters were placed on the wall of the room.

### *Participants*

Participants were all university students, aged between 20 and 26. Between 20 and 40 volunteers are expected. Students were chosen because they have a wide spectrum of phones, as discussed above.

### *Procedure*

On entering the room, users were asked to choose a poster from a group and to take a photo of it using their mobile phone. The process was repeated until one poster from each of the ten groups had been photographed. They were not given any instruction as to how they should take the photos of the posters. The ordering of both the groups and the lighting was randomized between users to reduce bias.



**Figure 23 - The posters stuck on the wall of the experiment room**

### **4.1.2 Results**

A simple initial analysis was carried out on the data. This involved simply finding the success ratio (number of successful decodings compared to the total number of images). The results showed that the algorithm only worked on 44% of the images. This was not an acceptable level of success and, hence, a second iteration of development would be required.

Two key reasons for this poor recognition performance were identified. The first was that users were taking photos that were not adequately focused on the target poster. For example, some users took photos where the poster in question occupied approximately 20% of the total image area. This makes it difficult for the algorithm to correctly locate the guidebars (see Figure 24).

The second problem area is somewhat related to the first in that, even if the bars were successfully located, the barcodes were of such a high density (containing

forty nine “bits” representing the seven digit number that was encoded) that they were often misread and returned incorrect values.



Figure 24 - Examples of the photos taken by users in the first round of experiments. Results clockwise from top-left: Successful, unsuccessful (poorly framed), unsuccessful (poorly focused), unsuccessful (poorly focused).

## 4.2 Changes

### 4.2.1 Target Lines

As stated, one of the major problems was the fact that users were taking photos which were not adequately focused on the poster. Photos were either too “zoomed out” - containing too much of the area around the poster - or too “zoomed in” - taken so close to the poster that either the guidebars or the barcode were not in the frame. Since it would be impractical, given the limitations of the hardware, to improve the algorithm to the point where it could cope with all possible image angles generated by users we need to add affordances to the poster which encourage the photographer frame their photographs in such a way that the system is able to correct analyse them. To do this we replaced the “target lines” from the first iteration with a full frame. This change was made after critical incident interviews with users following the first round of experiments. Using principles of participatory design [32] users were asked how they could be best informed what to focus on when taking the

photograph and one of the major suggestions was the replacing of the target lines with a full frame, as shown in Figure 25, below.



Figure 25 - The two sets of posters used in round 1 and 2 of the experiments . Note the difference in barcodes and targeting lines.

#### 4.2.2 Image Recognition

Besides the issue of photographic framing, we sought to improve the second reported problem of barcode interpretation by exploring an alternative barcode encoding system. The new system utilized binary encoding rather than the UPC system. A thirteen bit binary code was used, thereby reducing the number of bars from twenty eight to thirteen. The new coding system was much less dense than the UPC system, with only 13 bars instead of 49. The barcode along the bottom of the image was removed and the height of the top barcode was increased by 40% to further increase the chance of a successful decoding. Under the original UPC system the total number of media packs representable by the barcode was 99999 (Five digits). Under the revised system the highest number representable is 8191 ( $1111111111111_2=8191_{10}$ ).

Whilst this is a significant drop in capacity, it seems unlikely, given the locality inherent in the system (after all, a Bluetooth device has a maximum range of 32 feet), that there will ever be a need for a single system to manage that many media packs. If that ever did become necessary then the posters could be enlarged, the barcode reading algorithm altered slightly, and the barcodes increased in size by a single bit, thereby doubling the media pack capacity of the system.

## Chapter 4 - User Study 1

Further, the combination of lighting variations and the low quality of the mobile phones' cameras was found to produce a high variance of brightness levels across a single image. For this reason the simple thresholding algorithm implemented was found to be inadequate as, in many cases, the guidebars would be expanded to join neighbouring regions, or shrunk or split. Similarly the barcode elements could be expanded to join together or shrunk to the point where they disappeared (see Figure 26 for an example of this effect). To combat this, the thresholding algorithm was modified to use an adaptive threshold algorithm as described by Gu [21] and Wellner [44].



**Figure 26 - Example of how naive thresholding can lead to the guidebars being enlarged or shrunk.**

Adaptive thresholding algorithms do not consider the average pixel values of the whole image, but rather a smaller subsection of pixels, when deciding whether a pixel should be given a white or black value.

### 4.3 Experiment Round 2

#### 4.3.1 Design

The new posters were then created, using the new barcode system and framing lines, but using the same pictorial images as those from the first round. The experiments were then repeated. The second round was run in the same manner as the first, with the same images being used for the posters, the participants being sourced from the same demographic and the same instructions being given.

### 4.3.2 Results

The same initial analysis as that for the first round was carried out, this time showing a 73% success rate.

The framing of images was better, with 0.02% of images in the second round being incorrectly frames, against 7.9% in the first round.

### 4.3.3 Recommendations

Given the reasons discussed above, we felt the algorithm could not be improved; we make recommendations about the deployment of posters in order to maximize the success rate of the recognition algorithm. To do this we must determine which poster design elements had a significant effect on the outcome of a submitted image.

Since we have multiple independent explanatory variables and a binary response (the system either outputs a correct value or it does not), the chosen modeling approach was logistic regression within the framework of the generalized linear model (i.e. GLM with a logistic link function) [13].

This enables us to determine to what extent each variable influences the chance of getting the correct output from the algorithm.

The results showed that most of variables were not found to be significant factors on the final result. The size of the posters, however, was a significant predictor of success, with images 10cm and larger performing significantly better than those of 5cm and 7cm (with p-values of 0.0076, 0.00674 and 0.00827 for 10cm, 15cm and 20cm respectively).

Thus, when the system was deployed at later stages in the project, it was ensured that posters were larger than 10cm in width.

In order to determine the accuracy of the algorithm under “best case” conditions the dataset was trimmed to exclude all images under 10cm in size. The resulting success rate was 87%.

	Estimate	Std. Error	z value	Pr(> z )	
(Intercept)	-0.004007	0.879252	-0.005	0.99636	
Glossy	0.060945	0.549659	0.111	0.91171	
as.factor(Size)7	0.802364	0.713907	1.124	0.26105	
as.factor(Size)10	2.606685	0.976485	2.669	0.0076	**
as.factor(Size)15	2.445808	0.90269	2.709	0.00674	**
as.factor(Size)20	2.540905	0.962112	2.641	0.00827	**
as.factor(Image)111	-1.10419	1.072332	-1.03	0.30315	
as.factor(Image)222	-1.305182	0.942912	-1.384	0.1663	
as.factor(Image)1024	-0.296681	0.928562	-0.32	0.74934	
as.factor(Image)1111	0.326001	0.953193	0.342	0.73234	

Table 3 - Result table for second round of experiments

The final step was to combine the trimmed datasets from the first and second rounds of experiments and run the Generalised Linear Model again to determine if the second round results were significantly better than the first round.

The resulting table showed the second round to be significantly more successful than the first round (p-value=0.00178).

	Estimate	Std. Error	z value	Pr(> z )	
(Intercept)	-0.2903	0.54603	-0.532	0.59496	
as.factor(Size)15	0.0487	0.59491	0.082	0.93476	
as.factor(Size)20	1.08997	0.66551	1.638	0.10146	
as.factor(Image)111	0.49429	0.80982	0.61	0.54162	
as.factor(Image)222	0.69052	0.77231	0.894	0.37127	
as.factor(Image)1024	-0.02106	0.74154	-0.028	0.97734	
as.factor(Image)1111	0.22506	0.74406	0.302	0.76229	
Round	1.60528	0.51387	3.124	0.00178	**

Table 4 - Result table after trimming data

#### 4.3.4 Comparison

Thus we can conclude that the second version of the algorithm, in conjunction with the new poster design, performed significantly better than the original algorithm and poster design. A success ratio of 87% was considered acceptable since the remaining 13% of images that did not work were, in general, too poorly

## Chapter 4 - User Study 1

photographed for the current algorithm ever to work on (see Figure 27 for an example). This was either because the photo was too poorly focused or, in the case of two of the images, still not adequately framed.



**Figure 27 - Example of two poorly taken photos. The left photo is poorly framed and the right photo is not adequately focused.**

## 5. User Study 2

In order to gauge the effectiveness of a system it is imperative that it is tested in an environment similar to that in which it would eventually be deployed. Duh et al [16] showed there is a significant difference in results obtained in a laboratory when compare against those obtained “in the field”. For this reason, we deployed our system on public transport to observe its use and effectiveness. We chose the Jammie Shuttles, the student public transport system at the University of Cape Town, as our deployment environment due to its easy accessibility and due to the fact that deploying on an actual minibus taxi could involve numerous logistical and safety risks.

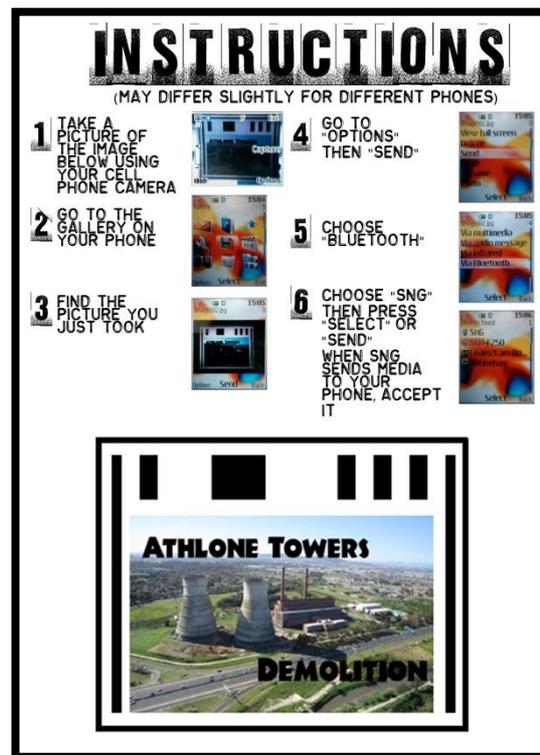


Figure 28 - One of the posters, complete with instructions, used in round 3 and 4 of the experiments

## 5.1 Experiment Round 3

### 5.1.1 Design

#### *Hypothesis*

Users are able to, without external instruction or prompting, correctly photograph the posters and subsequently send the photograph, via Bluetooth, to the Snap 'n Grab server which is, in turn, able to correctly analyse the photograph and return the relevant media to the user within 2.5 seconds of receiving the photograph.

Secondary hypothesis: The system is able to handle multiple requests simultaneously.

#### *Task*

The posters (see Figure 28 for an example) were stuck on the windows of a Jammie shuttle bus, the student transport service at the University of Cape Town.

Users were to take photos of the posters and submit them, via Bluetooth to the HTC Touch Pro, which would analyse the photos and return the requisite media to the users.

#### *Variables*

Success rate: Percentage of submissions for which the system is able to analyse the image and correctly decode the code embedded in it.

Time taken: How long (in seconds) the system takes to return the media packs.

#### *Equipment*

The system ran on an HTC Touch Pro.

The users were to use their own mobile phones to take the pictures.

#### *Venue*

The system was tested on a Jammie Shuttle, a bus with a carrying capacity of about 45 seated people. There were on average 15 people on the bus.

## Chapter 5 – User Study 2

The posters were stuck onto the windows of the bus. This was done to allow the most access and the most visibility of the poster.

The experiments were carried out during the day, and the weather was overcast on all days, meaning that there was little glare and not many shadows.

### *Participants*

Participants were students and staff of UCT. Between 20 and 40 volunteers were expected. This would again give us a good spectrum of photographic styles and camera types.

#### **5.1.2 Observations**

The system was deployed on a Jammie shuttle on two consecutive days. During this time an observer sat on the bus and watched how people interacted with the posters. A total time of about 6 hours was spent on the bus.

During the 6 hours, not one person made use of the system. The only interaction observed was one person who took a poster off the window of the bus and tore it in half as a joke.

It was decided that, in order to test the system in a situated environment, people on the bus would have to be prompted to make use of the system.

## **5.2 Experiment Round 4**

### **5.2.1 Changes**

The fourth round of experiments was set up in the same way as the third. The only change was that two extra facilitators joined the observer on the bus and engaged with people to encourage them to use the system.

These two people would move around the bus, introduce themselves to people, ask them if they had a camera phone and, if so, to attempt to use the system. If they chose to do so they were rewarded with a soft drink and R10 (about \$1.50) as a small incentive. Pictures of the facilitators engaging with people and those people subsequently using the system are showing in Figure 29.



Figure 29 - Users taking part in the 4th round of experiments

### 5.2.2 Results

A total of 6 hours was spent on the bus, during which time 54 people made use of the system. The results are detailed below:

Number of submissions	Correct Decodings	Success Percentage
54	20	37%

The algorithm managed to correctly decode the submitted images 37% percent of the time.

Regarding the time taken for the decodings:

Type	Number of submissions	Average Time per submission
Single decodings	50	2.1 seconds
Parallel decodings	4	3.5 seconds

## Chapter 5 – User Study 2

Through the course of the experiment, on only two occasions, was there more than one image to be decoded at one point in time. For the remainder of the cases the system only handled single requests.

The average time for a single request was 2.1 seconds. This is below the time limit of 2.25 seconds.

The average time for a “parallel” request was 3.5 seconds. Whilst this is above the time limit set, the result does not have much power due to the fact that the sample size is only four submissions.

What was noted was that a large number of the submissions had no chance of being decoded by the algorithm. These are characterised by the fact that either the guidebars or the barcode are not part of the picture taken. A further analysis of these images is undertaken in Section 5.3.2 below.

Factoring out the improperly captured images, where the frame or barcode were not captured in the image, we can investigate how the algorithm coped with lighting and image shake alone. By this measure, 20 images were trimmed from the input set and the success percentage rose to 59%.

<b>Number of submissions</b>	<b>Correct Decodings</b>	<b>Success Percentage</b>
34	20	59%

### 5.3 Discussion

#### 5.3.1 Round 3 Discussion

In this section we will discuss the observations from the third round of experiments. Since no quantitative data was gathered during this round due to the fact that no users attempted to make use of the system, all data is in the form of observations of how people interacted with the posters and the system on the bus.

It was interesting to note that not one person attempted to use the system during the time it was deployed on the Jammie Shuttle. Possible reasons for this are

1. They had no conceptual idea of how the system worked, and thus had no desire to use it.
2. They were unable to determine from the posters what they were expected to do
3. The posters were insufficient to motivate them to make the effort to use the system. The posters did not make it clear that they would receive something of potential value for using the system.
4. They were afraid of using the system for various reasons such as the fear of failure or the fear of receiving malicious content that could damage their phones.

We were able to circumvent these initial problems by utilizing “technology evangelists” on the Jammie Shuttles. These individuals would go up to people on the buses, introduce themselves, tell passengers about the system and invite them to use it. This is similar to the helper technique employed by Brignull et al [9] to encourage people to cross the threshold from awareness to participation. This approach was successful in that it got people to interact with the system. The problem with this approach is that it is not sustainable (we cannot deploy a technology evangelist with every deployment of the system). Thus, the four potential reasons for the perceived apathy towards the system would need to be addressed before the system could be deployed.

It should be noted that, in the fourth round of deployment, there were cases of people who were not approached by the facilitators, but instead observed other people using the system, noticed a poster next to them on the bus, and made use of the system themselves, without external prompting. This would suggest that, once there are a number of people using the system and giving it credibility, others are more inclined to “take the risk” and attempt to use it on their own – viral behaviour similar to that noted by Donner [14] and the “honey pot effect” noted by Brignull et al [9].

### 5.3.2 Image Recognition

In this section we will discuss the performance of the image recognition algorithm. We will look at its accuracy in laboratory and field conditions, discuss

the reasons for the differences and put forward suggestions for improvement. We will also look at its efficiency and discuss how the system performed when being used by multiple users simultaneously.

### *Accuracy*

In the laboratory (the third round of experiments) the image recognition algorithm produced the correct output code 87% of the time. This was deemed an acceptable level of success.

When deployed on campus transport (in the fourth round of experiments) the algorithm returned the correct output code only 37% of the time. This is a drop of 50%. Even after discarding images that were incorrectly framed the algorithm worked only 59% of the time, a drop of 28%.

This must lead us to conclude that There are two possible reasons for this severe drop in results:

1. The movement inherent in public transport makes it more difficult for users to take photographs that are adequately focused on the posters. That is, the photos taken are too blurred to be correctly analysed.
2. Users, in general, are not able to gauge, from the posters and the accompanying instructions, how to correctly photograph the posters.

Considering that there was no difference in the instructions given in round two and round four, it is unlikely that the second reason would be responsible for the drop in results. Therefore, we must conclude that the blur and shake introduced by the movement of the bus lead to the drop in results.

A possible solution to this problem would be to print bigger posters, thereby making it easier for users to focus their photographs on the poster. A problem with this solution, however, would be that, as the size of the poster increases, so does the possibility that the user will incorrectly frame the photograph. That is, they are more likely to not include the guidebars in the photograph if it is a larger poster.

It is, however, also possible, given the current state of the art in mobile image recognition, that it is not feasible to develop a shot-code-like system for use on public transport where there is a lot of bouncing or shaking, such as a bus or taxi. We could find no example in the literature of a shot-code system that has successfully been deployed on a bus or taxi.

### *Efficiency*

The second question, that of the efficiency of the algorithm, required that the system was able to decode the image in less than 2.5 seconds. The results show that, on average, it took 2.1 seconds to decode an image when the system was concerned with only one image. This means that the speed of the algorithm is acceptable when the system is not being used by more than one user.

Since only two cases of simultaneous use were encountered it is difficult to draw any significant conclusions relating to how the system performs under higher loads. Further tests would need to be carried out in order to determine how the system fairs under those circumstances.

The secondary hypothesis posed the question as to whether the system would be able to handle multiple requests simultaneously. This hypothesis is confirmed by the fact that two cases were noted in which more than one user was able to make use of the system at the same time. As mentioned, however, further testing would need to be carried out in order to determine the performance impact of this simultaneous usage.

### **5.3.3 Bluetooth issues**

Aside from the image recognition problems encountered several issues with the use of Bluetooth on mobiles were noted during this round of experiments.

1. Blackberry and iPhone mobile devices were unable to make use of the system. This is because iPhones do not allow files to be exchanged using the OBEX protocol via Bluetooth and Blackberry devices do not allow files be exchange via Bluetooth without first pairing the devices.
2. This leads to the second problem noted: A number of people, instead of simply sending the file via Bluetooth to the system, attempted to first pair

the devices. Since pairing devices requires the exchange of passkeys, and therefore interaction on the server side, this is not possible. This is an education problem which would have to be overcome.

3. The Bluetooth security settings on some users' mobile phones prevented the server from returning files to the user. This meant that they could send files to the server, but when the server attempted to send the requisite files back to the user the sending would fail. This, once again, is an education problem.

## 6. Conclusion and Future Work

This chapter discusses the Snap ‘n Grab Lite project in terms of how it answered the research questions laid out in Chapter 1. The results gathered from the two user experiments are analysed to determine the extent to which the system is ready for deployment in the like of the environments specified in Chapter 1. Finally, possible future work to extend and improve the project is discussed.

### 6.1 Discussion of project

This study aimed to investigate extending the capabilities of the Snap ‘n Grab media distribution system by replacing the desktop-based server with a mobile phone and by replacing the large LCD display screen with paper posters. These changes introduced unique challenges, largely due to the lower processing power afforded by the mobile phone. As such, the image recognition algorithm, the most computationally expensive aspect of the system, needed to be redesigned.

In Chapter 1 we introduced the Snap ‘n Grab system and motivated for its adaption as a mobile system. Work done in developing contexts was discussed, focusing on Bidwell’s work in the Eastern Cape of South Africa [5, 6], detailing how people in the back of taxis exchange information using their mobile phones.

Chapter 2 investigated prior work in mobile displays, usability and mobile image recognition.

The design of the Snap ‘n Grab Lite system was discussed in Chapter 3, with particular emphasis on the image recognition algorithm. Rohs’ Visual Code algorithm was investigated and adapted for our use.

The system was then tested in laboratory conditions (Chapter 4) and in the field, on a Jammie Shuttle (Chapter 5) in order to determine how it performed in different circumstances and if users were able to gauge how it should be used based on the affordance offered by the paper posters. The results showed that the system functioned well (87% success rate) in laboratory conditions but that

the success rate dropped substantially (to 37%) in the field. Various reasons for this decrease were proposed and discussed.

### 6.2 Discussion of algorithm

The algorithm used in this project to locate the barcode was adapted from the system developed by Rohs [36]. Whereas Rohs' algorithm use one guidebar on the right and one on the bottom of the barcode, our adaption placed a guidebar on each side of the image. This was done so that the barcode running between the guidebars could be more easily located. In testing it was found that utilizing the method employed by Rohs could cause difficulty in locating the barcode due to perspective deformations. Placing the guidebars on each side reduced the effect these deformations had.

The speed of the algorithm could be improved by removing, as far as possible, all floating point operations. This is because most mobile devices have no dedicated hardware floating point processor, and thus must carry out all floating point operations in software, which leads to a significant decrease in efficiency.

We have sought to limit the use of floating point operations in our implementation of the system, but found there was a tradeoff between accuracy and efficiency.

### 6.3 Discussion of System

#### 6.3.1 Platform

The Windows Mobile platform made porting the original Snap 'n Grab system to a mobile device a trivial exercise. Additionally, the development of a system on the platform is made easier by the tools available, namely the .NETCF and Visual Studio IDE. The Visual Studio IDE gives the developer the ability to deploy the code to the mobile device, and debug it on the PC whilst it runs, though a USB connection. This makes development and debugging a much simpler task.

Since the start of the project a new mobile operating system, Android, has achieved popularity. With this popularity comes new, powerful devices built to run it. At the time of writing the most powerful Android device available was the

## Chapter 6 – Conclusion and Future Work

Samsung Galaxy S, with a 1GHz processor and 512MB of RAM [20], almost twice as powerful as the HTC Touch Pro used in this project.

Also since the start of this project, Windows Mobile released a new version of their operating system: Windows Mobile 7, which should also bring with it new and more powerful devices.

The results of these more powerful devices are discussed in the section “Future” below.

### 6.3.2 Viability

Whilst the current Snap ‘n Grab Lite system is not ready for deployment in moving environments, as shown by the results of Chapter 5, it could still be deployed in some of the other environments mooted in Chapter 1. The system still has advantages over the original Snap ‘n Grab system in that it does not require constant power, can be secured far more easily, and is far more mobile. As such it could be deployed in a villages where there is not a constant supply of electricity (a reality in many of the villages in the Eastern Cape), or in public spaces where the theft of a large LCD display is too high a risk.

It would be beneficial to test the system in environments such as these as it would give greater insight into how the usage of the system changes in differing environments.

### 6.3.3 Future

Whilst mobile hardware improvements are not governed by Moore’s Law as desktop hardware is, due to the power constraints imposed by batteries [24], mobile devices are still improving at a rapid pace. As such, it is not inconceivable that it will be only a short period of time before mobile hardware has developed to the stage where it is feasible to deploy a previously “desktop designed” image recognition algorithm such as SIFT or SURF on a mobile device. Once hardware reaches this level the original Snap ‘n Grab system could be ported, with much greater ease, to a mobile server.

## Chapter 6 – Conclusion and Future Work

With this porting would come different challenges, however, as the paper posters would remain, this time without barcodes. As such, they would need to be redesigned to ensure that they offer affordance to users so that they are not just seen as pictures but as portals to more media.

As mobile phones improve, so will the cameras included with them. This will mean that the images submitted to the system will, in the future, probably be of a higher quality. The iPhone 4 includes technology to sharpen images by interpolating multiple blurred images into one sharper one [3]. Pertinent to our system, this should improve the quality of images obtained in bumpy environments, such as moving transport.

### 6.4 Discussion of Literature

As mentioned, we could find no mention in the literature of any previous projects involving image recognition of photographs taken on moving transport.

Additionally, there are no cases of public displays deployed on moving transport.

With the rise in popularity of touch screen tablet computers, we see an increase in the number of projects focusing on low-cost, smaller screened, public displays. Projects such as Hermes [12] or CityWall [33] could be adapted to run on a tablet computer.

These could then easily be deployed on transportation such as a bus, taxi or subway.

### 6.5 Discussion of project results

Five questions were raised at the outset of this project. In this section we will assess each of these in the light of the results of the two sets of user experiments.

#### 6.5.1 Question 1: Can we create an image recognition algorithm that runs efficiently on a mobile phone?

As the results of both sets of experiments show, it is possible to design an image recognition algorithm that runs efficiently on a mobile phone. It took the system, on average, 2.1 seconds to analyse a single image, which is 0.4 seconds below the threshold of 2.25 seconds laid out in the problem statement.

**6.5.2 Question 2: What is the best barcode form we can devise for the system?**

The UPC-based barcoding system was tested in the first round of experiments was found to be too dense for the image recognition algorithm to successfully decode. It was then replaced with a binary system which, whilst lowering the capacity of the system from 99999 to 8191, was found to deliver significantly better results.

**6.5.3 Question 3: Can the system decode photographs taken in a static (laboratory) environment?**

In the laboratory studies the image recognition algorithm, in conjunction with the binary barcoding system, was found, under conditions which were later implemented in the field, to return the correct barcode value 87% of the time.

**6.5.4 Question 4: Can the system decode photographs taking in a moving environment, such as public transport?**

When deployed on public transport, the Jammie Shuttle, the system was only able to return the correct barcode value for submitted images 37% of the time. Even when trimming the data set to images which were correctly framed (those in which the guidebars are visible, hence giving the algorithm a realistic chance of success) the success rate rose only to 59%. As discussed at the end of Chapter 5, this would lead us to believe that the system cannot be deployed on moving transport, due to the inherent movement and shake from the vehicle.

**6.5.5 Question 5: Can we design the posters in such a manner that users are able to determine what is required of them in terms of photographing the poster and submitting the photos to the system?**

The field deployment initially required that users interact with the system of their own volition, without external prompting. Through the course of this part of the deployment not one person attempted to make use of the system. Only once facilitators were introduced did people begin to interact with the system. Once people were interacting with the system, however, we did observe cases of people using the system themselves after simply seeing other people making use of it. This would lead us to conclude that the posters are designed in a way that people are able to understand how to use the system, but, for the reasons

## Chapter 6 – Conclusion and Future Work

discussed at the end of Chapter 5, they are, in many cases, apathetic or reluctant to do so until they have seen other people using it.

### 6.6 Conclusion

This project has aimed to answer five questions around mobile image recognition and the usability of a mobile media distribution system, Snap 'n Grab Lite.

We can conclude that a mobile image recognition algorithm was successfully designed, along with a binary barcoding system. The system was able to analyse and return the correct barcode value embedded in photos taken in a laboratory environment 87% of the time. When deployed in a moving environment, public transport, the success rate dropped to 37%.

Users were able to determine how to use the system, based on the posters designed. They were, however, apathetic towards the system, when it deployed on public transport, until they saw other people making use of it, at which point they became interested enough to attempt to interact with the system.

### 6.7 Future work

This project could be extended in the future allowing users to upload media, similar to the original Snap 'n Grab. This brings with it many challenges, mostly due to the static nature of the paper posters. One possibility would be to make use of a small Bluetooth printer which the server could send the newly created poster to.

Another extension would be to allow remote updating of the media packs, through a web interface. This would allow the creation of generic media packs, such as “Today’s news” where the content could be updated daily from a remote location. This would also mean that all instances of the system could be updated from one central location rather than going out and adding media one by one.

In order for the system to be financially viable, a business case would need to be made for it. One possibility is the selling of “Advertising Packs”, where someone like a retailer can devote an entire media pack to advertising their goods or services, or “Advertising Inserts” where a single advert image is inserted into

## Chapter 6 – Conclusion and Future Work

another media pack. This would have to be carefully considered, however, to prevent users resenting the extra download and thinking of it as spam.

## References

## References

- [1] 8TA. 2010. <http://www.8ta.com>. 2010.
- [2] ADAMS, R. 2009. UPC and EAN Bar Code Page. 2010.
- [3] APPLE INC. 2010. Apple - iPhone 4 - Camera. 2010.
- [4] BAY, H., FASEL, B. AND VAN GOOL, L. 2006. Interactive museum guide: Fast and robust recognition of museum objects. In *Proceedings of the first international workshop on mobile vision*.
- [5] BIDWELL, N. 2009. Anchoring design in rural customs of doing and saying. *Human-Computer Interaction-INTERACT 2009* 686-699.
- [6] BIDWELL, N. AND HARDY, D. 2009. Dilemmas in situating participation in rural ways of saying. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7*, ACM, 145-152.
- [7] BLUETOOTH SIG. 2010. Object Exchange (OBEX). 2010.
- [8] BLUETOOTH SIG. 2010. Bluetooth Basics. 2010.
- [9] BRIGNULL, H. AND ROGERS, Y. 2003. Enticing people to interact with large public displays in public spaces. In *Human-computer interaction: INTERACT'03; IFIP TC13 International Conference on Human-Computer Interaction, 1st-5th September 2003, Zurich, Switzerland*, Ios Pr Inc, 17.
- [10] CELL LIFE. 2010. Cell Life. 2010.
- [11] CELLC. 2010. <http://www.cellc.co.za>. 2010.
- [12] CHEVERST, K., DIX, A., FITTON, D., KRAY, C., ROUNCEFIELD, M., SAS, C., SASLIS-LAGOUDAKIS, G. AND SHERIDAN, J.G. 2005. Exploring bluetooth based mobile phone interaction with the hermes photo display. In *Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*, ACM New York, NY, USA, 47-54.
- [13] COOK, D., DIXON, P., DUCKWORTH, W.M., KAISER, M.S., KOEHLER, K., MEEKER, W.Q. AND STEPHENSON, W.R. 2001. Binary response and logistic regression analysis. *Part of the Iowa State University NSF/ILI project Beyond Traditional Statistical Methods*.
- [14] DONNER, J. 2008. The rules of beeping: Exchanging messages via intentional "missed calls" on mobile phones. *Journal of Computer-Mediated Communication* 13, 1-22.
- [15] DONNER, J. AND GITAU, S. 2009. New paths: exploring mobile-centric internet use in South Africa. In *Pre-Conference on Mobile Communication at the Annual Meeting of the International Communication Association*.
- [16] DUH, H.B.L., TAN, G.C.B. AND CHEN, V.H. 2006. Usability evaluation for mobile device: a comparison of laboratory and field tests. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*, ACM, 181-186.
- [17] FERSCHA, A. AND VOGL, S. 2002. Pervasive web access via public communication walls. *Pervasive Computing*, 84-97.
- [18] GITAU, S. 2010. ICT Aided Citizenry Participation.
- [19] GROSLAND, L. 2007. Innovation Day - Big Board.

## References

- [20] GSM ARENA. 2010. Samsung I9000 Galaxy S.
- [21] GU, J. 2009. Development of computer vision algorithms using J2ME for mobile phone applications.
- [22] ITU. 2010. Key Global Telecom Indicators for the World Telecommunication Service Sector.
- [23] KATO, H. AND TAN, K.T. 2007. Pervasive 2D barcodes for camera phone applications. *Pervasive Computing, IEEE* 6, 76-85.
- [24] KOUSHANFAR, F., PRABHU, V., POTKONJAK, M. AND RABAEY, J.M. 2002. Processors for mobile applications. In *Computer Design, 2000. Proceedings. 2000 International Conference on*, IEEE, 603-608.
- [25] LEIBBRANDT, M.V., WOOLARD, I., FINN, A. AND ARGENT, J. 2010. Trends in South African income distribution and poverty since the fall of apartheid. OECD.
- [26] LOWE, D.G. 1999. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*.
- [27] MADHAVAPEDDY, A., SCOTT, D., SHARP, R. AND UPTON, E. 2004. Using camera-phones to enhance human-computer interaction. In *Proceedings of Ubiquitous Computing (Adjunct Proceedings: Demos)*.
- [28] MARSDEN, G. 2008. Toward empowered design. *Computer* 41, 42-46.
- [29] MARSDEN, G. 2010. "What can we learn from the Developing World?" Impact of Mobile Applications from developing Markets on Mature Economies. *ICT for the Next Five Billion People* 47-54.
- [30] MAUNDER, A., MARSDEN, G. AND HARPER, R. Creating and sharing multi-media packages using large situated public displays and mobile phones. In *Proceedings of the 9 th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 07)*, 9-12.
- [31] MTN. <http://www.mtn.co.za>. 2010.
- [32] MULLER, M.J. AND KUHN, S. 1993. Participatory design. *Communications of the ACM* 36, 24-28.
- [33] PELTONEN, P., KURVINEN, E., SALOVAARA, A., JACUCCI, G., ILMONEN, T., EVANS, J., OULASVIRTA, A. AND SAARIKKO, P. 2008. It's Mine, Don't Touch!: interactions at a large multi-touch display in a city centre.
- [34] RAJ, H., GOSSWEILER, R. AND MILOJICIC, D. 2004. ContentCascade incremental content exchange between public displays and personal devices. In *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, IEEE, 374-381.
- [35] ROHS, M. AND GFELLER, B. 2004. Using camera-equipped mobile phones for interacting with real-world objects. *Advances in Pervasive Computing*, 265-271.
- [36] ROHS, M. 2005. Real-world interaction with camera phones. *Lecture notes in computer science*, 3598, 74.
- [37] SEIDEMAN, T. 1993. Bar codes sweep the world. *American Heritage of Invention and Technology*, 8, 56-63.

## References

- [38] SHNEIDERMAN, B. 1984. Response time and display rate in human performance with computers. *ACM Computing Surveys (CSUR)*, 16, 265-285.
- [39] SUOMALAINEN, J., VALKONEN, J. AND ASOKAN, N. 2007. Security associations in personal networks: A comparative analysis. *Security and Privacy in Ad-hoc and Sensor Networks*, 43-57.
- [40] TSAI, S.S., CHEN, D., SINGH, J.P. AND GIROD, B. 2008. Rate-efficient, real-time cd cover recognition on a camera-phone. In *Proceeding of the 16th ACM international conference on Multimedia*, ACM, 1023-1024.
- [41] VENTURA, P., SOUSA, H. AND JORGE, J. 2008. Mobile Phone Interaction with Outdoor Advertisements. In *Workshop on Designing and evaluating mobile phone-based interaction with public displays. CHI2008*.
- [42] VODACOM. <http://www.vodacom.co.za>. 2010.
- [43] WEISS, B.D., HART, G. AND PUST, R.E. 2010. The relationship between literacy and health. *Journal of health care for the poor and underserved*, 1, 351-363.
- [44] WELLNER, P.D. 1993. Adaptive thresholding for the DigitalDesk. *Xerox, EPC1993-110*.

## Appendices

### Appendix A – Bresenham's Algorithm

```
public byte[] getBarcodePixels(byte[,] image, int x0, int y0, int x1, int y1)
{
    // Use Bresenham's alg to draw a line through the barcode
    intdeltax = x1 - x0;
    intdeltay = y1 - y0;
    double error = 0;
    doubledeltaerr = (double)deltay / (double)deltax; // Assume deltax != 0 (line is not
    vertical),
    // note that this division needs to be done in a way that preserves the fractional
    part
    int y = y0;
    int length = x1 - x0 + 1;
    byte[] pixels = new byte[length];
    for (int x = x0; x <= x1; x++)
    {
        if (image[y, x] >= 125)
            pixels[x - x0] = 1;
        else
            pixels[x - x0] = 0;
        error = error + deltaerr;
        if (Math.Abs(error) >= 0.5)
        {
            if (error > 0)
            {
                error -= 1;
                y++;
            }
            else
            {
                error += 1;
                y--;
            }
        }
    }
    return pixels;
}
```

## Appendices

### Appendix B – Table of Results from User Study 1 – Round 1

Name	Device	Flash	Poster Size	Glossy	Lighting	Code	Correct Output
Image0093	5130	n	15	n	a	1111	n
Image0095	5130	n	7	n	dap	1	n
Image0096	5130	n	5	n	dap	1024	n
Image0094	5130	n	10	n	d	1	y
image040	6234	n	7	y	d	1111	n
image038	6234	n	20	y	a	111	y
image039	6234	n	10	y	d	222	y
8042010(003)	6870	y	7	y	dap	1111	n
8042010(004)	6870	y	5	y	a	222	n
8042010(009)	6870	y	5	n	dap	222	n
8042010	6870	y	20	y	dap	111	y
8042010(001)	6870	y	15	y	d	1	y
8042010(002)	6870	y	10	y	d	1024	y
8042010(005)	6870	y	20	n	a	111	y
8042010(006)	6870	y	15	n	a	1024	y
8042010(007)	6870	y	10	n	d	222	y
8042010(008)	6870	y	7	n	d	1	y
img0006	curve	n	10	n	d	1024	n
img0007	curve	n	5	n	d	111	n
img0005	curve	n	15	n	a	1111	y
img0008	curve	n	20	n	dap	222	y
img00082	curve 8900	n	20	y	d	1	n
img00085	curve 8900	n	7	y	dap	222	n
img00086	curve 8900	n	5	y	d	1111	n
img00089	curve 8900	n	10	n	d	111	n
img00091	curve 8900	n	5	n	dap	111	n
img00083	curve 8900	n	15	y	a	111	y
img00084	curve 8900	n	10	y	dap	1024	y
img00087	curve 8900	n	20	n	a	1	y
img00088	curve 8900	n	15	n	dap	222	y
img00090	curve 8900	n	7	n	d	222	y
05032010(003)	e63	n	5	n	a	111	n
8042010(001)#1	e63	n	15	y	dap	1111	n
8042010(002)#1	e63	n	10	y	d	1	n
8042010(005)#1	e63	y	20	n	a	1	n
8042010(006)#1	e63	n	15	n	dap	1	n
8042010(007)#1	e63	n	10	n	dap	1111	n
8042010(008)#1	e63	n	7	n	d	222	n
8042010(009)#1	e63	n	5	n	dap	1111	n
8042010(010)	e63	y	5	y	a	1	n
8042010#1	e63	n	20	y	dap	1024	n
5032010	e63	n	15	n	d	1111	y
05032010(001)	e63	n	10	n	dap	1	y
05032010(002)	e63	n	7	n	d	1111	y
8042010(003)#1	e63	n	7	y	d	111	y
5032010003	E66	y	15	y	d	111	n

## Appendices

5032010005	E66	y	15	n	d	1024	n
5032010002	E66	y	20	n	a	1111	y
5032010004	E66	y	10	y	dap	1	y
436	E71	n	10	y	d	222	n
437	E71	n	5	y	a	111	n
434	E71	n	20	y	a	1	y
435	E71	n	15	y	dap	222	y
image0027	expressmusic	n	10	y	a	1	n
image0029	expressmusic	n	5	y	dap	111	n
image0031	expressmusic	n	15	n	d	1111	n
image0032	expressmusic	n	10	n	a	1	n
image0034	expressmusic	n	5	n	dap	1111	n
image0025	expressmusic	n	20	y	d	1	y
image0026	expressmusic	n	15	y	a	111	y
image0028	expressmusic	n	7	y	dap	1024	y
image0030	expressmusic	n	20	n	d	1024	y
image0033	expressmusic	n	7	n	a	1024	y
68	f480	y	5	y	a	1111	n
69	f480	n	10	y	a	1111	n
70	f480	n	15	y	d	1024	n
71	f480	n	20	y	dap	111	y
photo004	jet	n	15	n	a	1111	n
photo005	jet	n	10	n	a	1024	n
photo006	jet	n	7	n	d	111	n
photo007	jet	n	5	n	d	1111	n
5032010238	n95	y	5	y	d	1	n
5032010239	n95	y	10	y	dap	222	n
5032010240	n95	y	15	y	dap	111	y
5032010241	n95	y	20	y	dap	1024	y

## Appendix C - Table of Results from User Study 1 – Round 2

Name	Device	Flash	Poster Size	Glossy	Lighting	Code	Correct Output
22042010278	Nokia 5800	N	20	n	d	1111	y
22042010279	Nokia 5800	N	15	n	dap	111	y
22042010280	Nokia 5800	N	10	n	a	1	y
22042010281	Nokia 5800	N	7	n	d	222	y
22042010282	Nokia 5800	N	5	n	dap	1111	y
22042010283	Nokia 5800	N	20	y	a	1024	y
22042010284	Nokia 5800	N	15	y	d	1111	y
22042010285	Nokia 5800	N	10	y	dap	1111	y
22042010286	Nokia 5800	N	7	y	a	1024	y
22042010287	Nokia 5800	N	5	y	d	1111	y
22042010063	Noka N70	N	20	n	dap	111	y
22042010064	Noka N70	N	15	n	a	222	y
22042010065	Noka N70	N	10	n	d	1	y
22042010066	Noka N70	N	7	n	dap	1111	n
22042010067	Noka N70	N	5	n	a	1024	n

## Appendices

22042010068	Noka N70	N	20	y	d	1111	y
22042010069	Noka N70	N	15	y	dap	111	y
22042010070	Noka N70	N	10	y	a	1111	y
22042010071	Noka N70	N	7	y	d	222	n
22042010072	Noka N70	N	5	y	dap	1024	n
Image016	Nokia 2600	N	20	n	a	1111	y
Image019	Nokia 2600	N	15	n	d	222	y
Image020	Nokia 2600	N	10	n	dap	222	y
Image021	Nokia 2600	N	7	n	a	1024	y
Image025	Nokia 2600	N	5	n	d	222	n
Image026	Nokia 2600	N	20	y	dap	1111	y
Image028	Nokia 2600	N	15	y	a	1111	y
Image030	Nokia 2600	N	10	y	d	222	y
Image031	Nokia 2600	N	7	y	dap	222	n
Image022	Nokia 2600	N	5	y	a	1111	y
DSC00066	SE Z610i	n	20	n	d	111	y
DSC00067	SE Z610i	n	15	n	dap	1024	y
DSC00068	SE Z610i	n	10	n	a	1	y
DSC00069	SE Z610i	n	7	n	d	1024	y
DSC00070	SE Z610i	n	5	n	dap	111	y
DSC00071	SE Z610i	n	20	y	a	1024	y
DSC00072	SE Z610i	n	15	y	d	1111	y
DSC00073	SE Z610i	n	10	y	dap	222	y
DSC00074	SE Z610i	n	7	y	a	1	y
DSC00075	SE Z610i	n	5	y	d	1024	y
422201001	Nokia N96	n	20	n	dap	111	n
422201002	Nokia N96	n	15	n	a	222	y
422201003	Nokia N96	n	10	n	d	1	y
422201004	Nokia N96	n	7	n	dap	10224	y
422201005	Nokia N96	n	5	n	a	1	y
422201006	Nokia N96	n	20	y	d	1024	n
422201007	Nokia N96	n	15	y	dap	111	n
422201008	Nokia N96	n	10	y	a	1111	y
422201009	Nokia N96	n	7	y	d	1	y
422201010	Nokia N96	n	5	y	dap	1024	n
Image096	Nokia 6680	n	20	n	a	1024	y
Image097	Nokia 6680	n	15	n	d	222	y
Image098	Nokia 6680	n	10	n	dap	222	y
Image099	Nokia 6680	n	7	n	a	1111	y
Image100	Nokia 6680	n	5	n	d	1111	y
Image101	Nokia 6680	n	20	y	dap	1024	y
Image102	Nokia 6680	n	15	y	a	1111	y
Image103	Nokia 6680	n	10	y	d	111	y
Image104	Nokia 6680	n	7	y	dap	1	y
Image105	Nokia 6680	n	5	y	a	222	n
Image00100	Blackberry 8520	n	20	n	d	1111	y
Image00101	Blackberry 8520	n	15	n	dap	222	n
Image00102	Blackberry 8520	n	10	n	a	222	y
Image00103	Blackberry	n	7	n	d	1024	y

## Appendices

	8520						
Image00104	Blackberry 8520	n	5	n	dap	1	n
Image00105	Blackberry 8520	n	20	y	a	111	y
Image00106	Blackberry 8520	n	15	y	d	111	y
Image00107	Blackberry 8520	n	10	y	dap	1	y
Image00108	Blackberry 8520	n	7	y	a	1111	y
Image00109	Blackberry 8520	n	5	y	d	1	y
26042010(001)	Nokia 5800	y	20	n	dap	1111	y
26042010(002)	Nokia 5800	y	15	n	a	222	n
26042010(003)	Nokia 5800	y	10	n	d	1	n
26042010(004)	Nokia 5800	y	7	n	dap	111	n
26042010(005)	Nokia 5800	y	5	n	a	111	n
26042010(006)	Nokia 5800	y	20	y	d	1111	y
26042010(007)	Nokia 5800	y	15	y	dap	1	y
26042010(008)	Nokia 5800	y	10	y	a	1024	y
26042010(009)	Nokia 5800	y	7	y	d	1111	n
26042010(010)	Nokia 5800	y	5	y	dap	1024	n
Image024	Nokia	n	20	n	dap	222	y
Image025#1	Nokia	n	15	n	a	1024	y
Image026#1	Nokia	n	10	n	d	1	n
Image027	Nokia	n	7	n	dap	222	n
Image028#1	Nokia	n	5	n	a	1111	n
Image029	Nokia	n	20	y	d	111	y
Image000	Nokia	n	15	y	dap	222	y
Image001	Nokia	n	10	y	a	1	y
Image002	Nokia	n	7	y	d	1111	n
Image003#1	Nokia	n	5	y	dap	1024	n

## Appendix D – Table of Results from User Study 2

Device ID	Media Pack	Correct Code	Received by User	Session	Time
0025CF987D44	1	1	0	1	2
0021FE1B2224	1	1	0	1	2
0021AAC76391	2	0	0	1	3
0023F1DC6188	1	0	0	1	2
6CD68A00E3DE	3	0	0	1	2
0022FC56DCEC	4	1	1	1	2
001FDFC3B942	4	1	1	1	2
001DBE0F63D0	4	1	1	1	2
0021FC5BA88A	1	1	1	1	2
0022A93B4B38	2	0	0	1	2
444E1AF32EBF	1	0	1	1	2
001FE4E45901	2	0	1	1	2
34C3AC31278D	1	0	0	2	2
0CDDEF9234B7	2	1	1	2	2

## Appendices

2021A5045AB4	4	0	1	2	2
001C9A71AA8E	2	1	1	2	2
001C9A2939F6	3	1	1	2	2
001CD635C369	2	0	1	2	3
44F459E7FD5B	4	1	1	2	3
001CD611AE67	1	0	1	2	2
0021FC0378B0	2	0	1	2	2
D87533040CC5	4	0	1	2	2
00233965EF39	2	0	1	2	2
A007985C536B	2	1	1	2	1
00256638918A	2	0	0	2	2
C038F9AFD5E0	2	0	0	2	2
00265FFEFEA6	2	1	1	2	2
44F459E59A5F	3	0	1	2	3
00256696F857	4	0	0	2	2
001EA321D803	3	0	0	3	2
00233900F169	3	0	1	3	1
00BD3AF36348	3	0	0	3	3
68EBAEB5A622	3	0	1	3	2
C87E75C9354F	1	1	0	3	3
0026E278BCAE	4	1	0	3	2
0022FCA8BCD4	3	0	0	3	2
0025674B74AE	4	0	1	3	2
001FE4463996	3	1	1	3	3
0024916E4697	1	0	1	3	2
0022987F0EB5	4	0	0	3	2
347E39FB169C	4	0	0	3	2
001C62E49F16	3	0	1	3	2
001979CED080	4	1	1	3	2
C87E75C03E2F	4	0	0	3	5
0025CFBB45A8	2	0	1	3	4
6C0E0DBECFBF	4	1	1	3	2
6C0E0DBE9996	4	1	1	3	2
001C62E49F16	3	1	1	3	3
303855B9968F	3	0	1	3	2
0021FECAA846	4	0	1	3	2
0024EF0816E0	3	0	0	3	3
0021FBE08BA7	4	0	0	3	2
0023F1DE3136	1	1	1	3	2
002345D4049E	4	0	1	3	3

### Image Recognition Techniques for use in a Mobile Public Interactive Display

Graeme Smith<sup>1</sup>, Professor Gary Marsden<sup>2</sup>  
Department of Computer Science  
University of Cape Town, Private Bag X3, Rondebosch, 7701  
Tel: +27 21 650 2712, Fax: +27 21 650 4511  
email: [graeme0811@gmail.com](mailto:graeme0811@gmail.com)<sup>1</sup>; [gaz@cs.uct.ac.za](mailto:gaz@cs.uct.ac.za)<sup>2</sup>

**Abstract-** This paper describes a system which runs on a mobile phone that allows for the distribution of media packages to users with Bluetooth enabled camera phones. Users take photographs of specially designed posters and send them, using Bluetooth, to the system. An algorithm that enables the system to recognize the user images is developed, evaluated and modified using two rounds of user experiments. The final algorithm is found to have a success rate of 87%.

**Index Terms**—Bluetooth, image recognition, ICT4D, usability, mobile

#### I. INTRODUCTION

A World Bank study in 2006 found that in “upper middle income” nations (a class which includes South Africa) 20.2% of the population has a fixed line telephone subscription, whereas 65.8% has a mobile phone. In countries classed as “lower middle income” the ratio was 30.4% mobile users against 10.9% fixed line and in “low income” nations the ratio was 7.5% against 0.9% [1]. Clearly, then, when designing technology solutions for the developing world, cellular handsets should feature prominently in our thought processes.

A project which utilizes the high prevalence of cellular handsets is the Snap ‘n Grab project (also known as Big Board) [2]. A Snap ‘n Grab system consists of a large screen display and a computer to drive it. The screen displays images representing content on the computer. This content can be pictures, videos, music files or text and is divided into packages known as “media packs”. It can represent job offers, AIDS information or simply (as was the case in a recent trial) recordings of the local gospel choir. Users can take a photo of an image representing a subject they wish to know more about (there are eight such options shown in Figure 1). Having done this, they can then send it, via Bluetooth, to the system. The system carries out image recognition on the image and replies by sending data, via Bluetooth, back to the user. All this is done at no cost to the user, as Bluetooth is free. Snap ‘n Grab can also allow users to create their own “media packs”. A user can set up an account by sending the system a v-card, followed by the media for the package. This new package will then be displayed on the screen.



Figure 1 - The Snap ‘n Grab system being displayed on a large LCD display.

Some drawbacks to the current Snap ‘n Grab system are *cost*, *security* and *mobility*. A large screen display of 40” (the average size of a Snap ‘n Grab screen) costs about \$1000. The computer to drive the screen would add an extra \$300. These costs are prohibitive in developing communities.

In places where there is a high crime rate, a high theft rate in particular, it is difficult to find a balance between system accessibility and security. Ideally the system should always be available for people to use; however, it is costly and difficult to ensure that the screens and computers are not easily stolen. The big screens are further constrained by the fact that they require a power source. This means that they could not be easily deployed in vehicles or places where electricity is not constantly available. We have developed a solution to the above three problems. Firstly, we replace the big screen with either printed posters, or individual stickers for each item. We can remove the need for a PC by porting the software to a cellular handset thereby dramatically cutting the cost. Using a cellular handset and paper posters greatly simplifies the security issue and bypasses the mobility problem. A high-end cellular handset costs about \$550, approximately 30% of the total cost of the original system. Secondly, security will be reduced to finding a suitable place to store the phone, such as a locked cupboard, or in the glove compartment of a vehicle. The posters will not need to be secured, as they will be of relatively low cost to replace. Finally, since the system is based on a cellular handset, it is clearly mobile. It could be deployed in a vehicle, in the middle of a field or in a small village where there is no constant source of electricity. The cellular handset would need to be charged occasionally, but it would not need constant power.

## II. RELATED WORK

### A. Image Recognition

In recreating the Snap 'n grab system on a cellular handset we encounter the problem of conducting image recognition, an inherently complex process, on reduced hardware. The Snap 'n Grab system utilized Scale Invariant Feature Transform (SIFT) [3] to carry out the image recognition step. It was, however, too slow to run on a cellular handset's limited processor, as it took about 5 seconds to run on a normal desktop computer.

Cellular handset image recognition research has focused primarily on barcode reading – both one and two dimensional – with many phones, such as the Nokia N95, being released with a built in barcode scanner. Cellular handsets have been shown to be capable of reading and decoding a barcode using the built-in camera [4]. Our system, however, required that the cellular handset be able to locate the barcode in a larger image. Rohs [5] has developed a method of orienting two-dimensional barcodes using a system of guide bars. The system works by locating a “guide” bar on the side, and then using that location to find the two blocks at top and bottom left of the image. The image can then be transformed to its original dimensions, and the two-dimensional barcode read accurately (Figure 2) (this is discussed in greater detail in the Algorithm Design section of this paper). Our system makes use of this concept of guide bars in order to locate the one-dimensional barcode along the top of the image. The barcode can then be decoded, and the image matched to an equivalent media pack.



Figure 2 - The 2D barcode used by Rohs, with the guide bars on the right and bottom of the image

### B. Public Displays

The Hermes system, developed by Cheverst et al. [6] was one of the first publicly situated displays to utilize Bluetooth technology to allow people to download media. It did, however, rely on a touch screen to allow users to select which data they wanted. The Snap 'n Grab system [2] was the first system to allow users to Bluetooth their selection, by means of a photograph, to the system in order to choose which media they desired. Another important aspect of the Snap 'n Grab system is that it did not require any software to be downloaded onto the user's cellular handset: all they are required to do is take a photo, using the cellular handset's built-in camera and to send it, via Bluetooth, to the system. This is a task people have become used to performing and thus is not unknown or intimidating.

In terms of static posters, shot codes have been placed on posters such as movie posters to allow cellular handset users to easily access websites related to the poster. Software to read these the codes has been included on cellular handsets such as the Nokia N95 [7,8]. All of these systems require

3<sup>rd</sup>-party software to be downloaded onto the user's cellular handset in order to perform the decoding task.

Our system uses the Bluetooth technology in the same way as the Snap 'n Grab system, and couples it with a form of static poster-based shot codes, in the form of barcodes and guide bars. This system does not require any special software to be installed on the users' handsets.

### C. Usability

Interactive displays are still fairly novel. To improve their usability Somervell et al. have suggested a list of heuristics associated with large, publicly situated displays [9]. Due to the fact that a static display was used, not all the heuristics apply to this project. Those that do are:

- Appropriate colour schemes to support information understanding;
- The layout should reflect the information according to its intended use;
- The display should show the presence of information, but not the details.

We will use these heuristics in creating our system. Finally, the Hermes system was evaluated by a user study where the user was asked to perform tasks such as taking a picture of items on the screen and sending these pictures to the display. Our evaluation will follow a similar form, wherein typical users will be asked to take photographs with their own handsets to create a realistic corpus of test data.



Figure 3 - An example of a poster used in the first iteration of Snap 'n Grab Lite

## III. SNAP 'N GRAB LITE

### A. System Overview

The new system, termed Snap 'n Grab Lite, was developed in two, iterative phases with user experiments taking place at the end of each phase.

The system was developed for Windows Mobile devices, and was deployed on an HTC Touch Pro running Windows Mobile 6.1. This platform was chosen due to the high processing power of the device and the fact that the original Snap 'n Grab system was written in C#, which simplified the transition from desktop computer to mobile device significantly.

The Snap 'n Lite program is a server program that runs on the Touch Pro and listens for incoming Bluetooth messages. When a message is detected, it is received and, if it is an image, it is analyzed and the requisite data is then sent back to the sender.

## Appendices

### B. Algorithm Design

#### 1) Guide Bars

As mentioned previously, image recognition is a complex process. Because speed of response is a significant factor in our application, it is not, at present, possible to utilize existing image recognition algorithms such as SIFT or SURF [3,10]. This is because the average response time for these algorithms, when implemented on a mobile phone, is greater than 45 seconds [11], which is far slower than the response time required for our application. Thus additional mechanisms must be utilized in order to simplify the recognition process. Barcode scanning is essentially a solved problem, with many applications now being shipped with standard mobile phones. A simple solution, from a technical viewpoint, would be to replace the images used in the original Snap 'n Grab system with barcodes. This, however, would not be of great use, considering the application of the system, since people need to know what they are taking a picture of, as that will allow them to make a decision as to what media they wish to receive. Therefore a dual system must be implemented, utilizing both pictures, for the human users, and a barcode system, for ease of recognition. The next technical challenge is that, in most barcode reading applications implemented on mobile phones, the barcode is assumed to take up the majority of the image, and is thus simple to find and read. In our application there is no guarantee that this will be the case, and indeed it is extremely likely that it will not be. Thus we need a way to efficiently locate the barcode within the image, so that it can be read and the requisite media sent to the user.

Rohs et al developed an algorithm for locating and orienting a two-dimensional barcode within an image [12]. They introduced the concept of guide bars, which sit alongside the barcode and are designed in such a way as to make them easy to identify within an image. Their algorithm for finding the guide bar works in the following manner:

- The image is thresholded (converted to black and white).
- The resulting image is then divided into contiguous regions using a two-pass approach.
- The regions are analyzed and their second-order moments are determined. These signify their "eccentricity" in the x and y direction.
- The ratios of the moments for each region are calculated and compared to the expected ratio for the guide bar. The best matching region is thus accepted as the guide bar.

In initial tests it was found that it was simpler to adapt Rohs algorithm and use two guide bars, one on each side of the image. Once found, a scanline can be run from the top of the left guide bar to the top of the right guide bar.

#### 2) Barcode System

In the first iteration of the Snap 'n Grab Lite system we decided to use an adaption of the UPC barcode system [13]. The UPC system uses a combination of seven bars of equal thickness to represent a single digit. Further, the bars are arranged in such a manner that, since there is no gap between them, there will always appear to be only four bars

per digit (since adjacent bars of the same colour will appear to form one, thicker bar).

We originally used a seven digit code with the UPC barcode system and positioned it along the top and bottom of the image, so that the picture can be taken upside down if necessary.

## IV. EXPERIMENTAL DESIGN

### A. Aim of experiment

The aim of the experiment was two-fold:

Firstly, to ascertain ideal values for a set of controlled independent variables (namely image size, poster medium and lighting conditions) under which the algorithm performs best. This will assist us in future deployments of the system to ensure optimal results.

Secondly, we wish to test how well the algorithm performs under the best-case set controllable variables, as well as some uncontrollable variables such as differing camera types, resolutions and "user idiosyncrasies" in taking photographs. We wish to test whether users are able, without instruction, to take photos that the system can analyze and from which it can decode the correct barcode number.

### B. Task

The experiment took place in two parts, one involving users and one conducted by the designer.

For the user-participation part: Five different posters were created. Each poster was printed in five different sizes (5cm, 7cm, 10cm, 15cm and 20cm in width) on glossy and matte paper. They were then grouped according to size and material, thereby forming ten groups of images (a group will contain all the posters of a certain size and a certain paper material). The users were required to take 10 photographs using their cellular handset cameras. They took one photo of a random poster in each group. The lighting conditions were changed after each photograph, alternating between directed lighting (simulated by a bright lamp), dappled light (simulated by lamp light through a diffusing material) and ambient light.

For the second part: The pictures taken by the users were retrieved and sent to the HTC Touch Pro. The recognition software was run on each image to determine whether or not it could locate and decode the barcode.



Figure 4 - The five posters used in the first round of experiments

### C. Variables

Poster medium:	Glossy and matte posters were tested.
Poster size:	The size of the physical poster (which therefore affects the size of the barcodes)
Lighting:	Three types of lighting were tested: ambient, directed and dappled. Directed light was created by aiming a lamp directly at the poster. Dappled light was

# Appendices

created by placing a diffusing material in front of the lamp.  
Success ratio: The ratio of correct media to incorrect.

## D. Materials

### 1) Equipment and venue

The system was run on an HTC Touch Pro. The users used their own mobile phones to take the pictures. The experiment was conducted in a closed room in the Computer Science Department at UCT.

### 2) Participants

Participants were sourced from the University of Cape Town. Ages ranged from 21 to 26. All were comfortable with taking photographs using a mobile phone.

## E. Procedure

The users were asked to choose a random poster from each group and to take a photo of it using their mobile phone. Once they had taken all ten photographs the images were retrieved. They were not told how to take photos of the posters.

## V. RESULTS OF EXPERIMENT

### A. Results

A simple initial analysis was carried out on the data. This involved finding the success ratio (number of successful decodings to the total number of images). The results showed that the algorithm worked on 44% of the images. This was not deemed an acceptable level of success and, hence, a second iteration of development was undertaken.

### B. Discussion

Two key problem areas were identified. The first was that users were taking photos that were not adequately focused on the poster they were photographing. For example, some users took photos where the poster in question occupied approximately 20% of the total image area. This makes it difficult for the algorithm to correctly locate the guidebars. The second problem area is somewhat related to the first in that, even if the bars were successfully located, the barcodes were of such a fine grain (containing forty nine "bits" representing the seven digit number that was encoded) that they were often misread and returned incorrect values.

## VI. REVISION OF SNAP 'N GRAB LITE SYSTEM

### A. Changes to algorithm

To enable easier reading of barcodes, the barcode system was altered. The new system utilized binary encoding rather than the UPC system. An eleven bit binary code was used (allowing up to 2047 different mediapacks to be accessible simultaneously), thereby reducing the number of bars from twenty eight to eleven. The barcode along the bottom of the image was removed and the height of the top barcode was increased by 40% to further increase the chance of a successful decoding. Some technical changes were also made to the algorithm. The thresholding algorithm was modified to use an adaptive

threshold algorithm as described by Gu [14]. This enabled the algorithm to perform better in conditions where the light intensity differed across the image. Additionally, in the guide bar location phase, the algorithm was altered to keep track of the top four matches, instead of only the top two. These four matches could then be analyzed and the best two matches from them would be used as the guide bars. This helped eliminate cases where poster images had vertical stripes that had similar eccentricity ratios to those of the guide bars, leading to false matches.

### B. Changes to sticker design

There was a definite need to get users to take photos that were composed, as far as possible, of only the poster in question. To do this we replaced the "target lines" from the first iteration with a full frame (as can be seen in Figure 5).



Figure 5 – An example of a poster used in the second iteration of Snap 'n Grab Lite

### C. Second round of experiments

New posters were then created, using the new barcode system and framing lines. The experiments were then repeated. The second round was run as similarly as possible to the first round, with the same images (albeit with the new barcodes and frames) being used for the posters, the participants being sourced from the same demographic and the same instructions being given.



Figure 6 - The five posters used in the second round of experiments

## VII. RESULTS OF SECOND EXPERIMENT

### A. Results

The same initial analysis as that for the first round was carried out, this time showing a 73% success rate. The next step was to determine the best set of controllable parameters under which to deploy the system. Since we have multiple independent explanatory variables and a binary response (the system either outputs a correct value or it does not), the chosen modeling approach was logistic regression within the framework of the generalized linear model (GLM) (i.e. GLM with a logistic link function) [15].

This enables us to determine to what extent each variable influences the chance of getting the correct output from the algorithm.

The results showed that the majority of the variables were not found to be significant factors on the final result. The size of the posters, however, was a significant predictor of success, with images 10cm and larger performing significantly better than those of 5cm and 7cm (with p-values of 0.0045, 0.00595 and 0.03390 for 10cm, 15cm and 20cm respectively).

In order to determine the accuracy of the algorithm under "best case" conditions the dataset was trimmed to exclude all images under 10cm in size. The resulting success rate was 87%.

The final step was to combine the trimmed datasets from the first and second rounds of experiments and run the GLM again to determine if the second round results were significantly better than the first round.

The resulting table showed the second round to be significantly more successful than the first round (p-value=0.031).

### B. Discussion

Thus we can conclude that the second version of the algorithm, in conjunction with the new poster design, performed significantly better than the original algorithm and poster design. A success ratio of 87% was considered acceptable since the remaining 13% of images that did not work were, in general, too poorly photographed for the current algorithm ever to work on (see Figure 7 for an example).



**Figure 7 - An example of a poorly taken image. Notice how the left guide bar is not in the frame, thereby causing the algorithm to fail.**

## VIII. CONCLUSION AND FUTURE WORK

### A. Conclusion

This paper has showed how the original Snap 'n Grab media distribution system has been adapted to run on a mobile phone, thereby opening it up to many potential further uses. The challenges faced were presented, as well as how the system was adapted to overcome these challenges. Experiments were carried out to evaluate how the algorithm performed. It was found to perform poorly in the first round of experiments, thus it was revised and a second round of experiments followed. The results from the second round of experiments were significantly better than those of the first round, with a success rate of 87% for posters of 10cm and larger.

### B. Future Work

Further study will be conducted by deploying the system in environments where it is likely to be used in the future. We will use the information gathered from the experiments as to what poster sizes to use to ensure that the algorithm performs optimally. These further studies will enable us to better understand how the algorithm performs with photographs taken in an environment such as a moving vehicles, as well as how users go about acquiring media and what media they are most interested in.

## IX. REFERENCES

- [1] Donner J. Research approaches to mobile use in the developing world: A review of the literature. *The Information Society* 2008;24(3):140-159.
- [2] Maunder A, Marsden G, Harper R. Creating and sharing multi-media packages using large situated public displays and mobile phones. *Proceedings of the 9th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 07)*.
- [3] Lowe D.G. Object recognition from local scale-invariant features. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*; 1999.
- [4] Ohbuchi E, Hanaizumi H, Hock L.A. Barcode readers using the camera device in mobile phones. *Cyberworlds, 2004 International Conference on*; 2004.
- [5] Rohs M. Real-world interaction with camera phones. *Lecture notes in computer science* 2005;3598:74.
- [6] Cheverst K, Dix A, Fitton D, Kray C, Rouncefield M, Sas C, et al. Exploring bluetooth based mobile phone interaction with the hermes photo display. *Proceedings of the 7th international conference on Human computer interaction with mobile devices & services: ACM New York, NY, USA; 2005*.
- [7] LeBrun J, Chuah C.N. Bluetooth content distribution stations on public transit. *Proceedings of the 1st international workshop on Decentralized resource sharing in mobile computing and networking: ACM New York, NY, USA; 2006*.
- [8] Rukzio E, Schmidt A, Hussmann H. Physical posters as gateways to context-aware services for mobile devices. *Sixth IEEE Workshop on Mobile Computing Systems and Applications, 2004. WMCSA 2004; 2004*.
- [9] Somervell J, Wahid S, McCrickard D.S. Usability heuristics for large screen information exhibits. *Proceedings of the Ninth IFIP TC13 International Conference on Human Computer Interaction; 2003*.
- [10] Bay H, Tuytelaars T, Van Gool L. Surf: Speeded up robust features. *Computer Vision-ECCV 2006* 2006:404-417.
- [11] Bay, H; Fasel, B; Van Gool, L. Interactive museum guide: Fast and robust recognition of museum objects. *Proceedings of the first international workshop on mobile vision: Citeseer; 2006*.
- [12] Rohs M, Gfeller B. Using camera-equipped mobile phones for interacting with real-world objects. *Advances in Pervasive Computing* 2004:265-271.
- [13] Adams R. UPC and EAN Bar Code Page. 2009; Available at: <http://www.adams1.com/upccode.html>. Accessed 30/4/2010, 2010.

# Appendices

- [14] Gu J. Development of computer vision algorithms using J2ME for mobile phone applications. University of Canterbury. Computer Science and Software Engineering; 2009.
- [15] Cook D, Dixon P, Duckworth W.M, Kaiser M.S, Koehler K, Meeker W.Q, et al. Binary response and logistic regression analysis. Part of the Iowa State University NSF/ILI project Beyond Traditional Statistical Methods 2001.

**Graeme Smith** received his undergraduate degree in 2007 from the University of Cape Town and is presently studying towards his Master of Science degree at the same institution. His research interests include usability, mobile computing and ICT4D.