Providing Media Download Services in African Taxis

Graeme Smith, Gary Marsden
ICT4D Research Centre
University of Cape Town
Rondebosch, Cape Town, South Africa
gaz@cs.uct.ac.za

ABSTRACT

In this paper we design and evaluate a system that allows users to download media over Bluetooth in a public transport situation in the developing world. Our work examines how the benefits of previous successful desktop systems can be ported to an entirely mobile platform which allows it to be deployed in a moving vehicle. We explore and test the performance of the system both in a static location (where the mobile system performs as well as the desktop system) and in a mobile setting (where results are more mixed). Finally, we make recommendations and give insights into barriers for placing media distribution systems in public transport.

Author Keywords
Bluetooth, ICT4D, M4D, Mobile data access, Situated Displays.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
Design, Experimentation.

INTRODUCTION

The ITU reports that in Africa in 2010, there were 13 million fixed line subscribers [3]. The same report also states that there were 333 million mobile subscribers in Africa in 2010. Not only are mobile phones prevalent in Africa, but many of those handsets are feature phones and owners seem adept at making full use of those features. As the mobile handset is often their only digital entertainment device, people in developing communities are, nonetheless, very eager to get media for their mobile phones [7]. The type of media ranges from music tracks of popular artists to information about a political party in an upcoming election.

Reports from researchers such as Smyth [7] show that there is a large informal information exchange community based around repair shops. Another method of media dissemination is public situated displays. These are, in general, large screens, that display information about media that 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 [2] and Snap 'n Grab, developed by Maunder et al [5]. 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.

In this paper we present a new take on Maunder et al's Snap 'n Grab system that allows it to be deployed in areas were mains power is not available. After testing the system in static locations, we investigate the potential of the system to be used in moving vehicles such as Africa's ubiquitous mini-bus taxis. The paper is structured as follows. We first lay out the design requirements for the system. We then present two experiments conducted on the new system in the laboratory to optimize performance. The system is then evaluated in a moving vehicle with mixed results. Finally, we reflect on the viability of providing these systems in public transport.

BACKGROUND

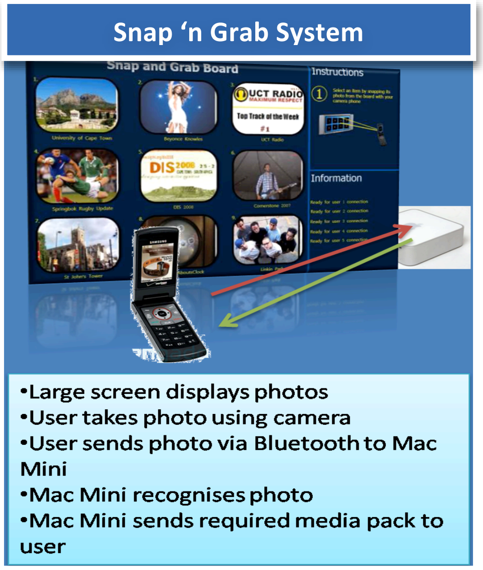
The Snap 'n Grab system [5] was the first situated display system that allowed users to select content from a screen, by means of a photograph, and choose the media they desired without the need for the installation of software on the client device. The system consists of a large screen display and a 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 local gospel choirs. 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 system a v-card, followed by the media for the package. This new package will then be displayed on the screen.

Although designed specifically for the developing world, the system has several critical shortcomings, namely *cost*, *security* and *mobility*. A large screen display of 40", the

SAICSIT '11, October 3–5, 2010, Cape Town, South Africa.

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. (Prices from 2010)

In places where there is a high theft rate in particular, (especially true in the townships of South Africa where the initial trials were conducted) 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.



Snap ‘n Grab System

- Large screen displays photos
- User takes photo using camera
- User sends photo via Bluetooth to Mac Mini
- Mac Mini recognises photo
- Mac Mini sends required media pack to user

Figure 1 - Summary of the Snap ‘n Grab interaction

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.

These issues extend to other comparable systems; both those created for the developing world [4] and for the developed [2]. The goal of our research is therefore to overcome these three problems and design a system that is cheap, easily secured and can be deployed in mobile environments, such as a moving taxi.

OVERCOMING LIMITATIONS

Below we present initial ideas on how to solve the three problems identified above.

Cost

The largest cost of any situated display is the screen. One way to reduce cost is to replace the screen with either printed posters, or individual stickers for each item. Whilst it is less convenient to create new packages (each package requires a sticker to be printed) the cost of this is easily bourn.

To overcome the cost of the computer, one can exploit the increase in computing power of cellular handsets to use the handset as a media server. By using a cellular handset as server the cost is reduced to $550, approximately 30% of the total cost of the original system.

Security

Security will be reduced to finding a suitable place to store the handset, such as a locked cupboard, or in the glove compartment of a vehicle. (In some preliminary research conducted as part of this project, we observed that many taxis already offer a handset charging service, locking passengers handsets in the glove box for the duration of the journey). The posters will not need to be secured, as they will be of relatively low cost to replace.

Mobility

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.

Design Challenges

There are two main challenges facing this solution: firstly, 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 challenges in determining the optimal print settings for the images, in terms of size and print material.

Processing power limitations

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 utilized in Snap ‘n Grab is simply too slow to be implemented on a mobile handset (it was reported as taking up to 219 seconds on a mobile architecture [1]).

A simple solution would be to replace the pictures used in Snap ‘n Grab with barcodes, like Shotcodes. 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 pictoral aspect of the Snap ‘n Grab system.

Mobile Image Recognition

A lot of work has been done in recent years on using mobile handsets 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.

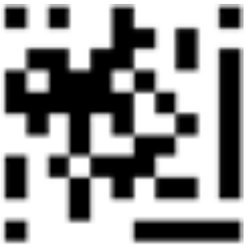


Figure 2 - An example of a VisualCode by Rohs et al. Notice the guidebars on the right and bottom of the code

Of interest to us is the Visual Code image recognition algorithm developed by Rohs et al (see Figure 2) [6]. In this code, guidebars are placed on the right hand side and bottom of the code. The guidebar has certain unique characteristics, which make it easy to locate within a photograph of the code. Once its position has been found the rest of the code can be easily read. We will adopt this mechanism by placing guidebars on each side of the pictoral representation of the media. Between these guidebars will be this picture, as well as a 1D barcode containing the unique ID of the media represented by the image. (See Figure 3).

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 (now running on a mobile handset). The server will analyse the image and return the respective media to the user.

INITIAL SOLUTION

Barcode

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 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.

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 (see Figure 3). 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.

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.

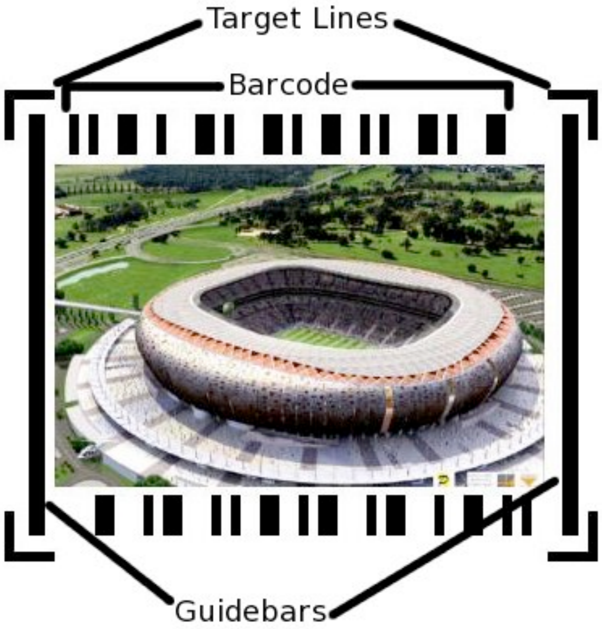


Figure 3 - A poster showing barcode, guidebars and target lines

Recognition Algorithm

In order to locate the barcode the guidebars need to be found. To do this 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 draw between them, through the barcode, which can then be read by reading off the pixels on the scanline.

The following steps are carried out:

Grayscaling

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. However, this thresholding algorithm was employed in the initial iteration as other forms of thresholding were considered too expensive.

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. We need to create a table of every region in the image, with 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 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. At the end of this process, all neighboring black pixels will have the same region values.

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 μ_{xx}, μ_{yy} and μ_{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.

The ellipse, E, is given by

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

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 $N = \sin(\theta)$

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

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

The ratio (δ) of these two axes is calculated: $\delta = \frac{s}{r}$

Guidebar Location

This is done for each region and δ 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.

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.

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.

Performance

In order to set a performance benchmark for the algorithm, we conducted a number of recognition tests with the desktop-based Snap ‘n Grab system. This system took, on average, 2.5 seconds to recognize an image. If our system is to be successful, it must meet this limit.

Testing

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 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.

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 where 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 4) 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.

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 handset to be used in Phase 2.

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.

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.

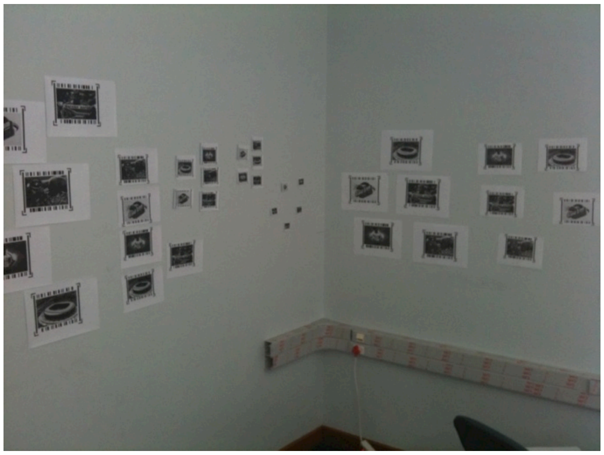


Figure 4 - The posters stuck on the wall of the experiment room

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 were all university students, aged between 20 and 26. Some 40 volunteers submitted photographs.

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.

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 where 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 5).

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.

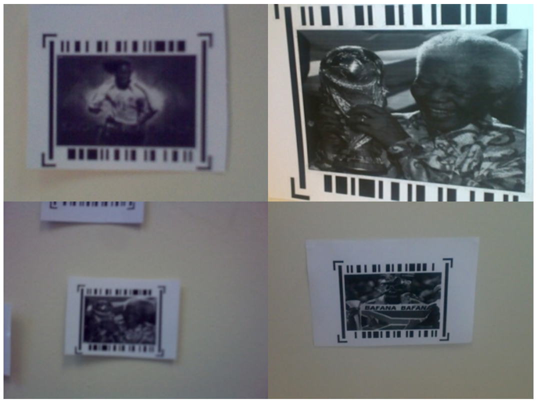


Figure 5 - Examples of the photos taken by users in the first round of experiments. Results clockwise from top-left: Successful, unsuccessful (poorly framed), unsuccessful (poorly focused), unsuccessful (poorly focused)

SECOND ITERATION

As stated, one of the major problems was the fact that users were taking photos that 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 to frame their photographs in such a way that the system is able to correctly 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, 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 6.

Image Recognition

Besides the issue or 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}).

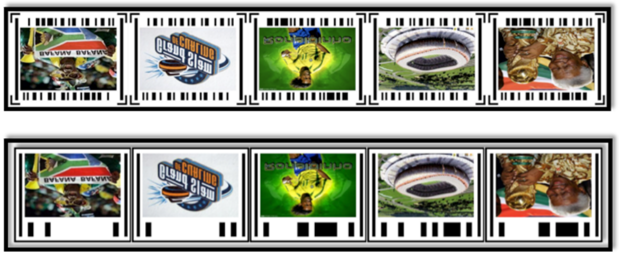


Figure 6 - The two sets of posters used in round 1 and 2 of the experiments. Note the difference in barcodes and targeting lines

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.

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 neighboring 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 7 for an example of this effect). To combat this, the thresholding code was modified to use an adaptive threshold algorithm.

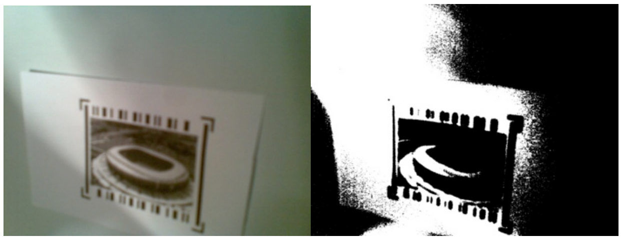


Figure 7 - Example of how naive thresholding can lead to the guidebars being enlarged or shrunk

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.

Testing

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.

Results

The same initial analysis as that for the first round was carried out, this time showing a 73% success rate. The images which were not recognized were due to poor framing on the part of the photographer. Those which were framed properly, could be recognized.

Analysing the images further allows us to make recommendations about the deployment of posters in order to maximize the success rate of the recognition algorithm. 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).

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%.

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 8). The accuracy of the new system is therefore no worse than the desktop based system in that it too could not recognize images which were badly framed. Our timing data showed that the image recognition also met the target average of 2.5 seconds to complete an image recognition.

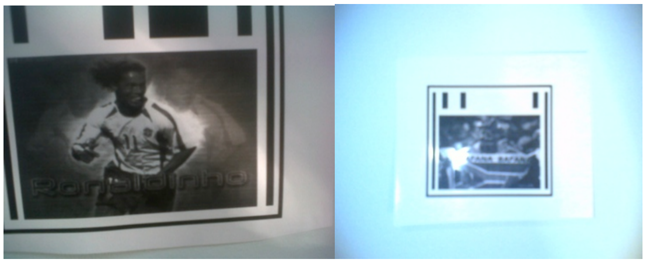


Figure 8 - Example of two poorly taken photos. The left photo is poorly framed and the right photo is not adequately focused

VEHICLE TESTING

Having addressed the issues of cost and security by showing that a system based on a mobile server is able to perform image recognition as accurately and rapidly as the desktop system, we wanted to test how well it could address the issue of mobility. That meant testing how well it could recognize images in a moving taxi, which introduces issues of:

- Variable lighting conditions
- Image blurring through vehicle motion
- Poor framing due to limited user mobility in overcrowded seats

The posters (see Figure 9 for an example) were stuck on the windows of a minibus taxi (in this case, on of the fleet owned and operated by the University of Cape Town). The carrying capacity of about 45 seated people. There were on average 15 people on the bus.

The system was deployed on the bus for 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.

Initial Experiment

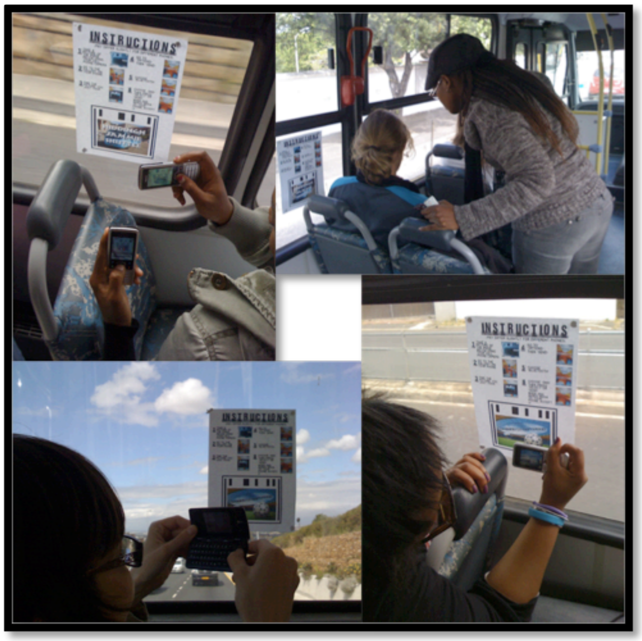


Figure 9 - Users taking part in the 4th round of experiments

During the six 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.

Second Mobile Experiment

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 9.

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 in Table 1 below:

Number of submissions	Correct Decodings	Success Percentage
54	20	37%

Table 1 – Summary of recognition algorithm success

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

Performance is summarized in Table 2:

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

Table 2 – Summary of recognition algorithm performance

Through the course of the experiment, on only two occasions was the system required to decode images simultaneously. 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 set.

The average time for a “parallel” request was 3.5 seconds. Whilst this is above the time limit set by the desktop system, these simultaneous downloads accounted for only 8% of the transactions.

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.

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%.

Number of submissions	Correct Decodings	Success Percentage
34	20	59%

Table 3 – Summary of recognition algorithm success on images where guidebar is visible

DISCUSSION OF MOBILE EXPERIMENTS

In the third experiment (the first mobile one) no quantitative data was gathered due to the fact that no users attempted to make use of the system. It is interesting to note that not one person attempted to use the system during the time it was deployed on the bus. 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 bus. These individuals would go up to people on the buses, introduce themselves, tell them about the system and invite them to use it. 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 employ 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.

Image Recognition

In the laboratory 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 the bus (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%.

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. Also, the posters are relatively large already and increasing their size further would obscure the windows almost completely, making it almost certain that the posters will be removed by third parties.

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 performance goal dictated 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. It could be that these cases of simultaneous use are rare and we need not consider them. However, until we have a system deployed and in constant use, it would be hard to measure the impact of this increased response time might have on users' reactions to the system.

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 that 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.

However, it should be noted, that these issues are intrinsic in the original Snap 'n Grab system and were not introduced by porting the system to the mobile handset. Buoyed by the fact that these limitations did not detract from the popularity of that system, we do not expect them to have a significant impact on the popularity of our mobile system.

CONCLUSION

This project set out to build a media distribution system that overcame the limitations inherent in the Snap 'n Grab system. These limitations we classified in three categories, namely: cost, security and mobility. To that end we have created a system that is demonstrably cheaper and easier to secure than the original system, whilst preserving a similar level of responsiveness. Whilst the mobility aspect was addressed, the system was shown not to perform well in public transport. 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%.

FUTURE WORK

Whilst mobile hardware improvements are not governed by Moore's Law as desktop hardware is, due to the power constraints imposed by batteries, 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.

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.

Whilst the current Snap 'n Grab Lite system is not ready for deployment in moving environments, it could still be deployed in other static environments where electricity supplies are sporadic and theft is a problem.

ACKNOWLEDGMENTS

Nicola J Bidwell for the insights around usage of handsets in taxis. Telkom/NSN Centre of Excellence for funding this research. The NRF for funding the research work. Microsoft Research for funding the original work on Big Board. Shikoh Gitau, Ilda Ladiera and Lesley Dodson for facilitating the trials in the taxi.

REFERENCES

1. Beat Fasel and Luc Van Gool. 2006. Interactive museum guide: accurate retrieval of object descriptions. In *Proceedings of the 4th international conference on Adaptive multimedia retrieval: user, context, and feedback* (AMR'06), 179-191.
2. Cheverst, K., Dix, A., Fitton, D., Kray, C., Rouncefield, M., Sas, C., Saslis-Lagoudakis, G. & Sheriadan, 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*, 47-54.
3. International Telecommunication Union. Key Global Telecom Indicators for the World Telecommunication Service Sector. 2010. http://www.itu.int/ITU-D/ict/statistics/at_glance/KeyTelecom.html
4. Jones, M., Harwood, W., Bainbridge, D., Buchanan, G., Frohlich, D., Rachovides, D., Frank, M. and Lalmas, M. 2008. *"Narrowcast yourself"*: designing for community storytelling in a rural Indian context. In *Proceedings of the 7th ACM conference on Designing interactive systems* (DIS '08). ACM, New York, NY, USA, 369-378
5. Maunder, A., Marsden, G. & Harper, R. 2007. Creating and sharing multi-media packages using large situated public displays and mobile phones. In *Proceedings of MobileHCI 07*, 9-12.
6. Rohs, M. AND Gfeller, B. 2004. Using camera-equipped mobile phones for interacting with real-world objects. *Advances in Pervasive Computing, (2004),* 265–271.
7. Smyth, T., Kumar, S., Medhi, I. and Toyama, K. 2010. Where there's a will there's a way: mobile media sharing in urban India. In *Proceedings of the 28th international conference on Human factors in computing systems* (CHI '10). ACM, New York, NY, USA, 753-762.