ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 3, pp: (208-218), Month: July - September 2014, Available at: www.researchpublish.com

EYEPHONE

[1]Ms. A. Sivasankari, [2]Mrs. G. Sangeetha Lakshmi, [3]M. Saranya

[1] M.Sc., M.Phil., HOD, Department Of Computer Science, D.K.M College For Women, India
[2] M.S(I.T)., M.Phil. Guide, Department Of Computer Science, D.K.M College For Women, India.
[3] M.C.A., Research Scholar, Department Of Computer Science, D.K.M College For Women, India.

Abstract: **As Smartphones evolve researchers are studying new techniques to ease the human-mobile interaction. We propose EyePhone, a novel "hand-free" interfacing system capable of driving mobile applications/functions using only the user's eyes movement and actions (e.g., wink). EyePhone tracks the user's eye movement across the phone's display using the camera mounted on the front of the phone; more specifically, machine learning algorithms are used to Track the eye and infer its position on the mobile phone display as a user views a particular application; and Detect eye blinks that emulate mouse clicks to activate the target application under view.**

We present a prototype implementation of EyePhone on a Nokia N810, which is capable of tracking the position of the eye on the display, mapping this positions to an application that is activated by a wink. At no time does the user have to physically touch the phone display.

Keywords**: Human-Phone Interaction, Mobile Sensing Systems, Machine Learning, Mobile Phones.**

1. INTRODUCTION

Human-Computer Interaction (HCI) researchers and phone vendors are continuously searching for new approaches to reduce the effort users exert when accessing applications on limited form factor devices such as mobile phones. The most significant innovation of the past few years is the adoption of touch screen technology introduced with the Apple iPhone and recently followed by all the other major vendors, such as Nokia and HTC. The touch screen has changed the way people interact with their mobile phones because it provides an intuitive way to perform actions using the movement of one or more fingers on the display (e.g., pinching photo to zoom in and out, or panning to move a map). Several recent research projects demonstrate new people-to-mobile phone interactions technologies. For example, to infer and detect gestures made by the user, phones use the on-board accelerometer , camera , specialized headsets , dedicated sensors or radio features . We take a different approach than that found in the literature and propose the EyePhone system which exploits the eye movement of the user captured using the phone's front-facing camera to trigger actions on the phone.HCI research has made remarkable advances over the last decade facilitating the interaction of people with machines.

We believe that human-phone interaction (HPI) extends the challenges not typically found in HCI research, more specially related to the phone and how we use it. We term HPI as developing techniques aimed at advancing and facilitating the interaction of people with mobile phones. HPI presents challenges that differ some what from traditional HCI challenges. Most HCI technology addresses the interaction between people and computers in ideal" environments, i.e., where people sit in front of a desktop machine with specialized sensors and cameras centered on them. In contrast, mobile phones are mobile computers with which people interact on the move under varying conditions and context. Any phone's sensors, e.g., accelerometer, gyroscope, or camera, used in a HPI technology must take into account the constraints that mobility brings into play. For example, a person walking produces a certain signature in the accelerometer readings that must be filtered out before being able to use the accelerometer for gesture recognition (e.g., double tapping the phone to stop an incoming phone call). Similarly, if the phone's camera is adopted in a HPI application the different light conditions and blurred video frames due to mobility make the use of the camera to infer events very challenging.

For these reasons HCI technologies need to be extended to be applicable to HPI environments. In order to address these goals HPI technology should be less intrusive; that is,

- It should not rely on any external devices other than the mobile phone itself;
- It should be readily usable with minimum user dependency as possible;
- It should be fast in the inference phase;
- It should be lightweight in terms of computation; and
- It should preserve the phone user experience, e.g., it should not deplete the phone battery over normal operations.

We believe that HPI research advances will produce a leap forward in the way people use their mobile phones by improving people safety, e.g., HPI techniques should aim to reduce the distraction and consequently the risk of accidents if driving for example, or facilitating the use of mobile phones for impaired people (e.g., people with disabilities).

We propose EyePhone, the first system capable of tracking a user's eye and mapping its current position on the display to a function/application on the phone using the phone's front-facing camera. EyePhone allows the user to activate an application by simply "blinking at the app", emulating a mouse click. The front camera is the only requirement in EyePhone. Most of the smartphones today are equipped with a front camera and we expect that many more will be introduced in the future (e.g., Apple iPhone 4G [1]) in support of video conferencing on the phone.The EyePhone system uses machine learning techniques that after detecting the eye create a template of the open eye and use template matching for eye tracking. Correlation matching is exploited for eye wink detection .

We implement EyePhone on the Nokia N810 tablet and present experimental results in different settings. These initial results demonstrate that EyePhone is capable of driving the mobile phone. An EyePhone demo can be found .The paper is organized as follows. We discuss the challenges encountered in the development of HPI technology. In presents the design of the EyePhone system followed by its evaluation. The future research direction are reported in related work and finishes with some concluding remarks.

2. HUMAN-PHONE INTERACTION

Human-Phone Interaction represents an extension of the field of HCI since HPI presents new challenges that need to be addressed specifically driven by issues of mobility, the form factor of the phone, and its resource limitations (e.g. ,energy and computation). More specifically, the distinguishing factors of the mobile phone environment are mobility and the lack of sophisticated hardware support, i.e., specialized headsets, overhead cameras, and dedicated sensors, that are often required to realize HCI applications.

2.1 Mobility Challenges.

One of the immediate products of mobility is that a mobile phone is moved around through unpredicted context, i.e., situations and scenarios that are hard to see or predict during the design phase of a HPI application. A mobile phone is subject to uncontrolled movement ,i.e., people interact with their mobile phones while stationary, on the move, etc. It is almost impossible to predict how and where people are going to use their mobile phones.

A HPI application should be able to operate reliably in any encountered condition. Consider the following examples: two HPI applications, one using the accelerometer, the other relying on the phone's camera. Imagine exploiting the accelerometer to infer some simple gestures a person can perform with the phone in their hands, e.g., shake the phone to initiate a phone call, or tap the phone to reject a phone call. What is challenging is being able to distinguish between the gesture itself and any other action the person might be performing.

For example, if a person is running or if a user tosses their phone down on a sofa, a sudden shake of the phone could produce signatures that could be easily confused with a gesture. There are many examples where a classifier could be easily confused. In response, erroneous actions could be triggered on the phone. Similarly, if the phone's camera is used to infer a user action, it becomes important to make the inference algorithm operating on the video captured by the camera robust against lighting conditions, which can vary from place to place.

In addition, video frames blur due to the phone movement. Because HPI application developers cannot assume any optimal operating conditions (i.e., users operating in some idealized manner) before detecting gestures in this example, (e.g., requiring a user to stop walking or running before initiating a phone call by a shaking movement), then the effects of mobility must be taken into account in order for the HPI application to be reliable and scalable.

2.2 Hardware Challenges

As opposed to HCI applications, any HPI implementation should not rely on any external hardware. Asking people to carry or wear additional hardware in order to use their phone might reduce the penetration of the technology. Moreover, state-of-the art HCI hardware, such as glass mounted cameras, or dedicated helmets are not yet small enough to be conformably worn for long periods of time by people. Any HPI application should rely as much as possible on just the phone's on-board sensors. Although modern smart phones are becoming more computationally capable , they are still limited when running complex machine learning algorithms . HPI solutions should adopt lightweight machine learning techniques to run properly and energy efficiently on mobile phones.

3. EYE PHONE DESIGN

One question we address in this paper is how useful is a cheap, ubiquitous sensor, such as the camera, in building HPI applications. We develop eye tracking and blink detection mechanisms based algorithms originally designed for desktop machines using USB cameras. We show the limitations of an off-the-shelf HCI technique when used to realize a HPI application on a resource limited mobile device such as the Nokia N810. The EyePhone algorithmic design breaks down into the following pipeline phases:

- An Eye Detection Phase;
- An Open Eye Template Creation Phase;
- An Eye Tracking Phase;
- A Blink Detection Phase.

3.1 Eye Detection

By applying a motion analysis technique which operates on consecutive frames, this phase consists on finding the contour of the eyes. The eye pair is identified by the left and right eye contours. While the original algorithm identifies the eye pair with almost no error when running on a desktop computer with a fixed camera (see the left image in Figure).

we obtain errors when the algorithm is implemented on the phone due to the quality of the N810 camera compared to the one on the desktop and the unavoidable movement of the phone while in a person's hand (refer to the right image in Figure).

Figure : Left figure: example of eye contour pair returned by the original algorithm running on a desktop with a USB camera. The two white clusters identify the eye pair. Right figure: example of number of contours returned by EyePhone on the Nokia N810. The smaller dots are erroneously interpreted as eye contours.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 3, pp: (208-218), Month: July - September 2014, Available at: www.researchpublish.com

Based on these experimental observations, we modify the original algorithm by:

- reducing the image resolution, which according to the authors in reduces the eye detection error rate,
- adding two more criteria to the original heuristics that filter out the false eye contours.

In particular, we filter out all the contours for which their width and height in pixels are such that $Widthmin \le width \le widthmax$ ***and*** $heightmin \le height \le heightmax$. The widthmin, widthmax, heightmin, and heightmax thresholds, which identify the possible sizes for a true eye contour, are determined under various experimental conditions (e.g., bright, dark, moving, not moving) and with different people. This design approach boosts the eye tracking accuracy considerably.

3.2 Open Eye Template Creation

While the authors in adopt an online open eye template creation by extracting the template every time the eye pair is lost (this could happen because of lighting condition changes or movement in the case of a mobile device), EyePhone does not rely on the same strategy. The reduced computation speed compared to a desktop machine and the restricted battery requirements imposed by the N810 dictate a different approach. EyePhone creates a template of a user's open eye once at the beginning when a person uses the system for the first time using the eye detection algorithm described above.

The template is saved in the persistent memory of the device and fetched when EyePhone is invoked. By taking this simple approach, we drastically reduce the runtime inference delay of EyePhone, the application memory footprint, and the battery drain. The downside of this off-line template creation approach is that a template created in certain lighting conditions might not be perfectly suitable for other environments.

We intend to address this problem as part of our future work. In the current implementation the system is trained individually, i.e., the eye template is created by each user when the application is used for the first time. In the future, we will investigate eye template training by relying on pre-collected data from multiple individuals. With this supervised learning approach users can readily use EyePhone without going through the initial eye template creation phase.

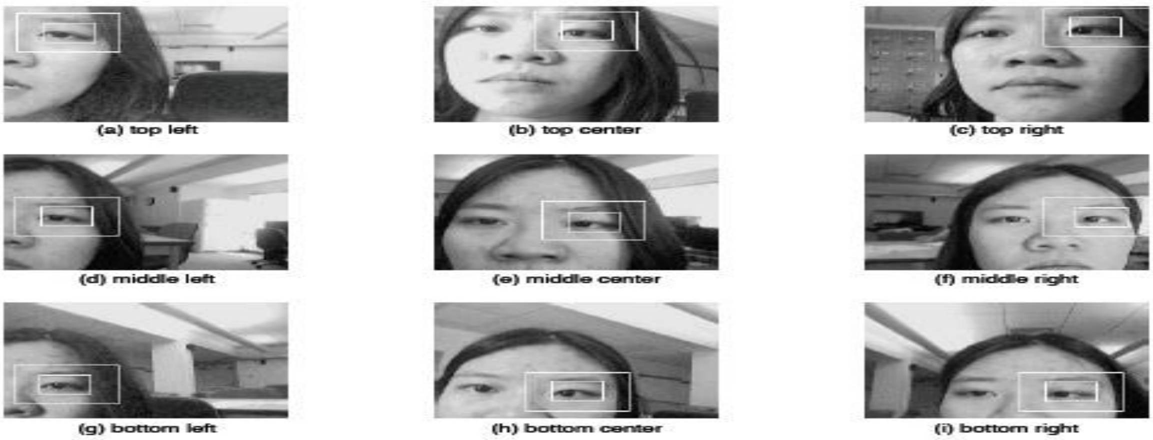

Figure : Eye capture using the nokia n810 front camera running the eyephone system. The inner white box surrounding the right eye is used to discriminate the nine positions of the eye on the phone's display. The outer box encloses the template matching region.

3.3 Eye Tracking

The eye tracking algorithm is based on template matching. The template matching function calculates a correlation score between the open eye template, Created the first time the application is used, and a search window. In order to reduce the computation time of the template matching function and save resources, the search window is limited to a region which is twice the size of a box enclosing the eye.

These regions are shown in Figure, where the outerbox around the left eye encloses the region where the correlation score is calculated. The correlation coefficient we rely on, which is often used in template matching problems, is the normalized correlation coefficient defined in .This coefficient ranges between -1 and 1. From our experiments this coefficient guarantees better performance than the one used in. If the normalized correlation coefficient equals 0.4 we conclude that there is an eye in the search window. This threshold has been verified accurate by means of multiple experiments under different conditions (e.g., bright, dark, moving, not moving).

3.4 Blink Detection

To detect blinks we apply a thresholding technique for the normalized correlation coefficient returned by the template matching function as suggested in. However, our algorithm differs from the one proposed in. In the authors introduce a single threshold T and the eye is deemed to be open if the correlation score is greater than T, and closed vice versa. In the EyePhone system, we have two situations to deal with :

- The quality of the camera is not the same as a good USB camera,
- The phone's camera is generally closer to the person's face than is the case of using a desktop and USB camera.

Because of this latter situation the camera can pick up iris movements, i.e., the interior of the eye, due to eyeball rotation. In particular, when the iris is turned towards the corner of the eye, upwards or downwards, a blink is inferred even if the eye remains open. This occurs because in this case the majority of the eye ball surface turns white which is confused with the color of the skin. We derive e four thresholds: $\mathbf{T_1^{min} = 0.64, T_1^{max} = 0.75, T_2^{min} = -0.53}$ **and** $\mathbf{T_2^{max} = -0.45}$.These thresholds are determined experimentally and again under different experimental conditions as discussed previously. If we indicate with cmin and cmax, respectively, the minimum and maximum normalized correlation coefficient values returned by the template matching function, the eye is inferred to be closed if $\mathbf{T_1^{min} \leq c_{max} \leq T_1^{max}}$ **and** $\mathbf{T_2^{min} \leq c_m}$.It is inferred to be open otherwise .The results of the blink detection algorithm.

Legend

- DS = eye tracking accuracy measured in daylight exposure and being steady;
- AS = eye tracking accuracy measured in artificial light exposure and being steady;
- DM = eye tracking accuracy measured in daylight exposure and walking;
- BD = blink detection accuracy in daylight exposure**.**

Table : EyePhone average eye tracking accuracy for different positions of the eye in different lighting and movement conditions and blink detection average accuracy.

Eye position	DS	AS	DM	BD
Top Left	76.73%	74.50%	82.81%	84.14%
Top center	79.74%	97.78%	79.16%	78.47%
Top right	80.35	95.06%	60%	82.17%
Middle Left	98.46%	97.19%	70.99%	74.72%
Middle center	99.31%	99.31%	76.52%	79.55%
Middle right	99.42%	75.79%	65.15%	80.1%
Bottom Left	98.36%	93.22%	78.83%	74.53%
Bottom center	90.76%	71.46%	85.26%	67.41%
Bottom right	84.91%	93.56%	78.25%	72.89%

4. EYE TRACKING

Eye tracking is a technique that captures eye behavior in response to a visual stimulus (e.g., computer interface, photograph, page in a newspaper, TV commercial). When using eye tracking, we tend to assume that where people look is where they focus their attention. This is not always true, as everyone can recall looking at something but not really "seeing it" or noticing something without having to look at it directly. These are some of the caveats that we need to be aware of when studying eye movements. However, in most cases, the "eye-mind" hypothesis holds (with visual attention being slightly ahead of the eye) and, by examining people's eye movements, we can gain insight into their cognitive processes and learn more about what they find important, interesting, complex, or confusing.

Saccadic eye movements, which consist of saccades and fixations, are the most common eye movements we make. Information processing occurs during fixations, where the eyes are relatively stationary and focused on a particular location of the stimulus. Saccades are rapid eye jumps from one place of the stimulus to another, during which the eyes are practically blind. However, because saccades are so short (less than one tenth of a second), the fact that we cannot see usually goes unnoticed. The device used to capture eye movements is called an eye tracker. The eye tracker determines the position of one or both eyes multiple times per second. It then superimposes those positions over a recording of the stimulus to determine where the person was looking at any given time. .Commonly used eye trackers differ in physical form, setup procedures, and tracking methodology. Some are most appropriate for studies involving stimuli presented on the computer screen, such as Web sites, software applications, or graphic images.

Table : Eye Tracking

Usability Testing The interaction between the user and the screen is mediated by the eye. We evaluate usability based on visual behavior as well as clicks, errors and interview feedback. ***Marketing Research*** The eye is the window to the mind of the consumer. We use advanced eye tracking technology along with traditional methods to meet your marketing research objectives. ***Scientific Applications*** The potential uses for eye tracking are limitless. As industry leaders, we have applied our technology and expertise in a range of important fields of science. ***Eye works™*** Eye Works™ is the only eye tracking software developed by researchers for researchers. This powerful package is compatible with more eye trackers than any other platform and includes all the tools you'll need to conduct advanced research in any field.	***Software*** The success of your eye tracking study depends on the effectiveness of your software. We have developed the tools to take your research to the next level. ***Cognitive Workload*** The ability to measure mental effort is invaluable. Our patented cognitive workload algorithm captures a direct signal from the brain as it is transmitted through the pupil. ***Capabilities*** The researchers at Eye Tracking, Inc. wear many hats. From marketing and usability to software and science, we have driven innovation in eye tracking for over a decade.

Some are better suited for studies that require interaction with physical objects, such as computer peripherals, handheld devices, or packaging. Others still are best for tracking how people view their surroundings when moving around freely, for example, while playing sports or navigating.

ADVANTAGES:

- Eye movement is faster than other current input media
- No training or particular coordination is required of normal users
- Can determine where the user's interest is focused automatically
- Helpful for usability studies to understand users interact with their environments

Potential Applications

- Eye gaze correction for videoconferencing
- Maximizing controllers' efficiency, minimizing dangers in air traffic displays
- Developing video games and graphics
- Marketing research and E-commerce website development
- Potentially could provide new and more effective methods of computer-human interaction.

EASY:

- Very quick set up. No complicated hardware to configure, simply install the PCI card in your computer and load the software
- Intuitive **user interface**
- **Real-time feedback** of eye movement data
- The Auto-threshold feature quickly provides luminance threshold values for discriminating both the pupil and glint automatically
- No calibration is required for many applications
- Easy Slip Correction feature and re-presentation of stray calibration points
- Automatic blink detection and suppression
- The powerful SDK makes it very easy to interface eye tracking with your other applications
- **Slide Show** -- super-easy stimulus presentation for experiments
- Intuitive graphical analysis of data with fixation-time representation

FLEXIBLE:

- Select between 30 Hz and 60 Hz operation modes
- Robust against different lighting conditions
- Pupil-only tracking, Glint-only tracking, or choose Pupil-Glint Vector tracking that provides tolerance to head movements
- Choose between Dark Pupil or Bright Pupil tracking
- Run ViewPoint on the same computer as your stimulus presentation or on a remote computer
- Real-time interface between the computer running ViewPoint and other computers (PC or Mac) provides maximum flexibility
- Software Developers Kit provides complete external control of ViewPoint
- Visual Range +/- 44° of visual arc horizontal and +/- 20° vertical

5. IMPROVING MOBILE DEVICE INTERACTION BY EYE TRACKING ANALYSIS

The new generation of smart phones has been revolutionized with the introduction of technologies like touch screen, accelerometer, gyroscope, photo camera, etc. These innovations in conjunction with the increase in hardware performance, allows a different approach in the use of these devices improving user experience and interaction. Several recent research projects demonstrate how the interaction with mobile phone technologies improved. As mentioned in, the evolution of mobile phones to Smartphone opened new horizons for the implementation of innovative types of mobile applications, like using the phone's camera for more specialized sensing activities, such as tracking the user's eye movement across the phone's display as a means to activate applications.

In fact, eye gaze sensing is an important method in human computer interfacing. The eye gaze is a more natural method to interact with a device than a mouse or keyboard. Eye movement is reflective of cognitive processes and eye gaze interaction could be a convenient way for controlling mobile devices. The methods for eye tracking can be classified into two categories: ***Intrusive and Non-Intrusive. Intrusive*** methods require direct interaction with the user. The user needs to wear head-mounted equipment resulting in discomfort and restricting their movement range. ***Non-Intrusive*** methods, instead, use images captured from a camera to estimate the gaze direction or an infrared based approach to enhance the contrast between the pupil and the iris.

In contrast to eye tracking systems for computers, mobile devices suffer from several drawbacks like: intensity of light (indoor or outdoor use), camera resolution, calibration issues (caused by head movements and mobile device movements).Eye tracking technology for interaction with mobile phones is not yet available as a stable and usable application. One reason is the lack of infrared devices for accurate eye detection. The data captured form a camera must be sufficient to understand the gaze movement. This implies the use of complex and heavy computational techniques which collide with the lack of processing power to handle video streams on these devices in real-time.

Various systems have been implemented that integrate eye tracking capabilities into a mobile phone. In a system capable of driving mobile applications using only the user's eye movements and actions is described, while in, different approaches, in particular dwell-time method and gaze gestures, are compared in order to investigate how gaze interaction can be used to control applications on mobile phone. The implementation of an eye tracking system using a smartphone and images captured from its camera, requires a robust method to detect the eyes location. In particular in the authors introduced Haar classifiers to accurately and rapidly detect faces within an image and can be adapted to accurately detect facial features, like eyes. In this work, we present a system architecture for eye tracking using an iPhone by processing the images captured from the device's front camera.

6. EVALUATION

In this section, we discuss initial results from the evaluation of the EyePhone prototype. We implement EyePhone on the Nokia N810. The N810 is equipped with a **Processor (400 MHz), RAM (128 MB), Operating (Maemo 4.1, a Unix based platform).** In system is Maemo 4.1, a Unix based platform on which we can install both the C Open CV (Open Source Computer Vision) library and our EyePhone algorithms which are cross compiled on the Maemo scratchbox. To intercept the video frames from the camera we rely on GStreamer, the main multimedia framework on Maemo platform. In what follows, we first present results relating to average accuracy for eye tracking and blink detection for different lighting and User movement conditions to show the performance of Eye-Phone under different experimental condition. We also report system measurements, such as CPU and memory usage, battery consumption and computation time when running EyePhone on the N810.All experiments are repeated five times and average results are shown.

- Daylight Exposure Analysis for a Stationary Subject
- Artificial Light Exposure for a Stationary Subject.
- Daylight Exposure for Person Walking.
- Impact of Distance Between Eye and Tablet
- System Measurements

6.1 Daylight Exposure Analysis For A Stationary Subject

The first experiment shows the performance of Eye-Phone when the person is exposed to bright daylight, i.e., in a bright environment, and the person is stationary. The inner white box in each picture, which is a frame taken from the front camera when the person is looking at the N810 display while holding the device in their hand, represents the eye position on the phone display.It is evident that nine different positions for the eye are identified.

These nine positions of the eye can be mapped to nine different functions and applications as shown in Figure 4. Once the eye locks onto a position (i.e., the person is looking at one of the nine buttons on the display), a blink, acting as a mouse click, launches the application corresponding to the button. The accuracy of the eye tracking and blink detection algorithms are reported in Table 1. The results show we obtain good tracking accuracy of the user's eye. However, the blink detection algorithms accuracy oscillates between ~67 and 84%. We are studying further improvements in the blink detection as part of future work.

6.2 Artificial Light Exposure for a Stationary Subject

In this experiment, the person is again not moving but in an artificially lit environment (i.e., a room with very low daylight penetration from the windows). We want to verify if different lighting conditions impact the system's performance.The results, shown in Table 1, are comparable to the daylight scenario in a number of cases. However, the accuracy drops. Given the poorer lighting conditions, the eye tracking algorithm fails to locate the eyes with higher frequency.

6.3 Daylight Exposure for Person Walking

We carried out an experiment where a person walks outdoors in a bright environment to quantify the impact of the phone's natural movement; that is, shaking of the phone in the hand induced by the person's gait. We anticipate a drop in the accuracy of the eye tracking algorithm because of the phone movement.This is confirmed by the results shown in Table 6.2, column4. Further research is required to make the eye tracking algorithm more robust when a person is using the system on the move.

6.4 Impact of Distance between Eye and Tablet.

Since in the current implementation the open eye template is created once at a fixed distance, we evaluate the eye tracking performance when the distance between the eye and the tablet is varied while using EyePhone.

We carry out the measurements for the middle-center position in the display (similar results are obtained for the remaining eight positions) when the person is steady and walking.

Table: Average CPU usage, RAM usage, and computation time for one video frame. The front camera supports up to 15 frames per second. The last column reports the percentage of used battery by EyePhone after a three hour run of the system.

CPU	RAM	COMPUTATION TIME	BATTERY USED AFTER 3h
65.4%	56.51%	~100 msec	40%

As expected, the accuracy degrades for distances larger than 18-20 cm (which is the distance between the eye and the N810 we currently use during the eye template training phase). The accuracy drop becomes severe when the distance is made larger (e.g., ~45 cm).

These results indicate that research is needed in order to design eye template training techniques which are robust against distance variations between the eyes and the phone.

6.5 System Measurements

In Table we report the average CPU usage, RAM usage, battery consumption, and computation time of the EyePhone system when processing one video frame – the N810 camera is able to produce up to 15 frames per second.EyePhone is quite lightweight in terms of CPU and RAM needs. The computation takes 100 msec/frame, which is the delay between two consecutive inference results.In addition, the EyePhone runs only when the eye pair is detected implying that the phone resources are used only when a person is looking at the phone's display and remain free otherwise. The battery drain of the N810 when running the EyePhone continuously for three hours is shown in the 4th column of Table .

Although this is not a realistic use case, since a person does not usually continuously interact with their phone for three continuous hours, the result indicates that the EyePhone algorithms need to be further optimized to extend the battery life as much as possible.

7. APPLICATIONS

7.1 Eye Menu

An example of an EyePhone application is Eye Menu as shown in Figure. Eye Menu is a way to shortcut the access to some of the phone's functions. The set of applications in the menu can be customized by the user.

The idea is the following:

- ❖ The position of a person's eye is mapped to one of the nine buttons.
- ❖ A button is highlighted when EyePhone detects the eye in the position mapped to the button.
- ❖ If a user blinks their eye, the application associated with the button is lunched.
- ❖ Driving the mobile phone user interface with the eyes can be used as a way to facilitate the interaction with mobile phones or in support of people with disabilities.

Figure : Eye Menu on the Nokia N900. While looking the display, a button is highlighted if it matches the eye position on the display. The highlighted button is ready to be "clicked" with a blink of the eye.In this example, the user is looking at the SMS button and the SMS keypad is launched by blinking the eye.

7.2 Car Driver Safety

Eye Phone could also be used to detect drivers drowsiness and distraction in cars. While car manufactures are developing technology to improve drivers safety by detecting drowsiness and distraction using dedicated sensors and cameras. Eye Phone could be readily usable for the same purpose even on low-end cars by just clipping the phone on the car dashboard.

8. CONCLUSION

In this paper, we have focused on developing a HPI technology solely using one of the phone's growing number of on board sensors, i.e., the front-facing camera. We presented the implementation and evaluation of the EyePhone prototype. The EyePhone relies on eye tracking and blink detection to drive a mobile phone user interface and activate different applications or functions on the phone. Although preliminary, our results indicate that EyePhone is a promising approach to driving mobile applications in a hand-free manner.

REFERENCES

[1] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan. 'uWave: Accelerometer-Based Personalized Gesture Recognition and its Applications. In Pervasive and Mobile Computing', vol. 5, issue 6, pp. 657-675, December 2009.

[2] E. Miluzzo, C.T. Cornelius, A. Ramaswamy,T. Choudhury, Z. Liu, A.T. Campbell. Darwin Phones: The Evolution of Sensing and Inference on Mobile Phones. In Eighth International ACM Conference on Mobile Systems, Applications, and Services (MobiSys '10), San Francisco, CA, USA, June 15-18, 2010.

[3] Nokia n810 tablet. http://www.nokiausa.com/findproducts/phones/nokia-n810.

[4] Bulling, J.A. Ward, H. Gellersen, and G. Troster. Eye Movement Analysis for Activity Recognition. In 11th International Conference on Ubiquitous Computing, pages 41–50. ACM,

[5] Advantages Of Using Eye Tracking : http://www.electronicink.com/think and http://www.simpleusability.com/services/usability/eye-tracking

[6] Http://www.zdnet.com/could-the-next-iphone-be-an-eye-tracking-phone-7000019411/

[7] Http://www.amazon.co.uk/movement-disorders-agnes-wong-m-d/dp/0195324269

[8] http://www.amazon.com/eye-movement-disorders-agnes-wong/dp/0195324269

[9] http://ciit-atd.edu.pk/admin/newsfiles/130330358861069032055605598.pdf

[10] http://www.ncbi.nlm.nih.gov/pmc/articles/pmc3339072/

[11] http://www.zdnet.com/could-the-next-iphone-be-an-eye-tracking-phone-7000019411/

[12] http://www.cs.dartmouth.edu/~campbell/papers/eyephone.pdf

[13] http://www.hotmobile.org/2012/papers/hotmobile12-final72.pdf