Improving Mobile Device Interaction by Eye Tracking Analysis

Carmelo Pino*, Isaak Kavasidis *
*Department of Electrical, Electronics and Informatics Engineering University of Catania,
Viale Andrea Doria, 6 - 95125 Catania, Italy

***Abstract*—This paper describes a non-intrusive eyetracking tool for mobile devices by using images acquired by the front camera of the iPhone and iPod Touch. By tracking and interpreting the user's gaze to the smartphone's screen coordinates the user can interact with the device by using a more natural and spotaneous way. The application uses a Haar classifier based detection module for identifying the eyes in the acquired images and subsequently the CAMSHIFT algorithm to find and track the eyes movement and detect the user's gaze. The performance of the proposed tool was evaluated by testing the system on 16 users and the results shown that in about 79% of the times it was able to detect correctly the users' gaze.**

I. Introduction

THE NEW generation of smartphones 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 [1], [2]. As mentioned in [3], the evolution of mobile phones to smartphones 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 [4] 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 [5]. Non-intrusive methods, instead, use images captured from a camera to estimate the gaze direction [6] or an infrared based approach to enhance the contrast between the pupil and the iris [7], [8], [9].

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 [10], a system capable of driving mobile applications using only the user's eye movements and actions is described, while in [11], 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 [12], 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 [13].

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. The remainder of the paper is as follows: in Section II we describe the system's general architecture and each block that composes it. In Section III, the performance of the system are discussed, while in the last section, concluding remarks are given.

II. System Architecture

The architecture of the implemented framework is based on a set of blocks that work asynchronously. Each of these blocks performs a specific function in order to capture the user's gaze.

The framework, called BAEyeTracking, uses the Apple Audio Video Foundation (AVFoundation) Framework to grab the video frames from the device's front camera. Every specific operation runs in a different queue using the Central Dispatch Queue Framework. In particular, there are four distinct modules:

- The data acquisition module aims at acquiring the images from the iPhone's front camera and at converting it into raw image data suitable for the subsequent processing steps. Given the small amount of memory, every operation must be as simple as possible, and must robustly release unneeded memory.
- For eye detection, a Haar classifier [13] is used in order to find the eyes and check if the rectangles of the eyes

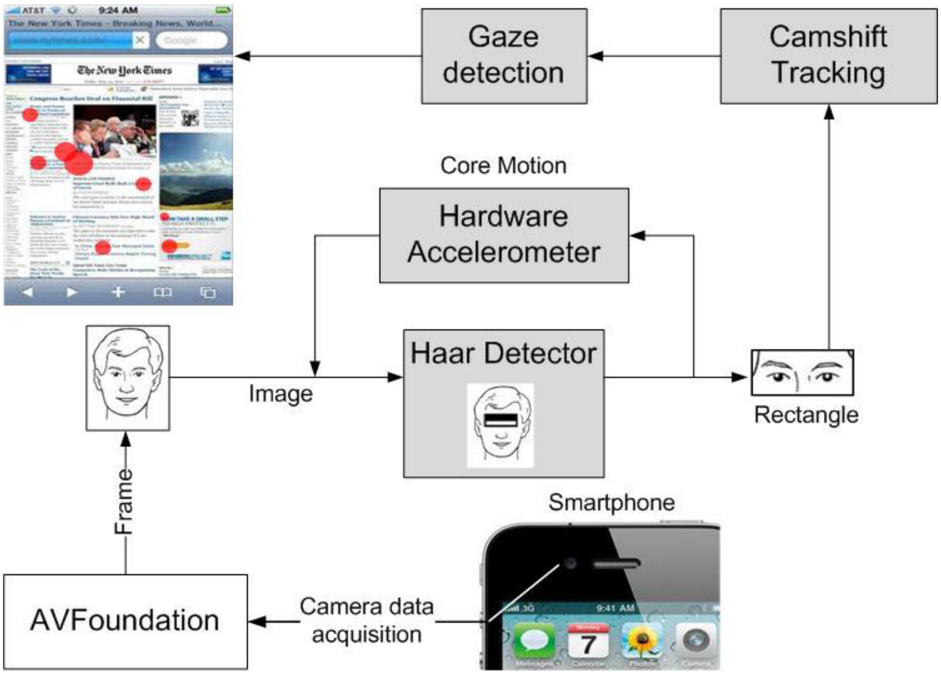

Fig. 1. System architecture.

are stable(i.e. the rectangles did not move significantly for three consecutive frames). Every time the Haar Detector starts, it checks the standard deviation between the previous rectangles and the current ones, and uses heuristic conditions to see whether the rectangles are really representing the eyes or not. Once the Haar Detector finds three stable samples, it enters in a stable state and lowers its execution frequency, until the rectangles destabilize. Using the device accelerometer, every time the Haar Detector is started, if the device was moved too far from its previous position, the detection will be forced to find new rectangles as soon as possible.

- To constantly keep track of the eyes position, an eye tracking module based on the CAMSHIFT algorithm was implemented. While being robust in static conditions, the Camshift algorithm must be configured accurately to be able to handle not only different lighting conditions but also different types of eyes.
- In order to interpret the acquired image in eye gaze location, a Gaze Detection module was used.

Every module is designed to work asynchronously in the background, and dispatches only the minimal amount of operations and data when needed on the main thread.

A. Haar-Like Eyes Detection module

In order to locate the eyes rectangle into the frame and to track any significant changes in the scene after the first recognition, a Haar classifier based on Haar-like features that are represented by adjacent rectangular regions (fig.2) at a specific location in a detection window, is used. To accomplish that, for each rectangle the sum of the pixel intensities is performed and its difference with the adjacent rectangle is calculated. This difference is then used to classify subsections of an image (eyes, nose etc...).

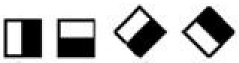

Fig. 2. Example of common Haar-like features.

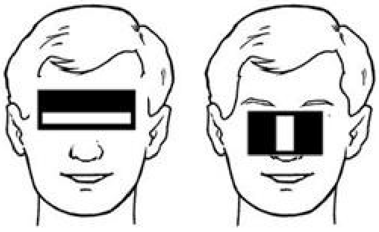

Fig. 3. Example of features overlay.

During the detection phase, a window of the target size is overlayed and moved over the input image (fig. 3) and for each subsection of the image the Haar-like features are calculated. The difference between the two rectangles is then compared to a threshold that allows to distinguish an object from another one.

Normally, the Haar Detector finds more than one eye in the scene, even if they are not present, and so it is necessary to filter these false positives out. To achieve that, from the set of the eye regions found, the two eye regions, which centroids have the minimum y-axis difference, are selected.

With this simple heuristic condition all the rectangles that do not stand on the same axis, are removed. Afterwards, the standard deviation between the current rectangle found and the previous one, in order to check whether the area has changed, is calculated. If it is greater than a threshold, the frame is classified as not stable and the algorithm starts again from the begining, otherwise the frame is classified as stable and the number of stable samples found is incremented. If the number of the consecutive stable samples is greater than a value, the detection module puts itself in the stable state, in which its execution frequency is reduced (1 per two seconds instead of 2 per second) until its state changes.

B. Hardware Accelerometer module

One of the biggest problems in implementing eye tracking on a mobile device is that the device (and its camera) is not adequately still and image misalignment occurs due to the phone's movement. Unfortunately, unless using a dock or position the phone on stable ground (e.g a table), these movements cannot be removed, and the problem must be dealed with other available sensor data. In particular, to face this problem, the Core Motion Framework was used to obtain gyroscope and accelerometer data. Every time the detection procedure is called, gyroscope and accelerometer data is sampled and compared to the previous values. If the difference is too high, it means that the device was moved from its previous position and, subsequently, the eyes will not be on the same position. In this case, the Haar Detector module switches to the non-stable state and initiates the eye detection procedure instantly.

C. CAMSHIFT Tracking module

In order to track the eye gaze, after the eyes' position has been identified, the CAMSHIFT algorithm [14] was used because it gives a great tradeoff between performance and efficiency. The CAMSHIFT algorithm tracks objects by matching the probability density functions of two consecutive frames. As in our case, as probability density function, generally, the hue plane histogram of the frames is chosen.

After the Haar Detector module goes into the stable state, the eye tracking module is used to monitor the eyes' movements and report their positions to the Gaze Detection module.

D. Gaze Detection module

By using the calculated data of the Eye Tracker module, the Gaze Detection module is used to translate the eye positions into the corresponding screen positions. While there are many algorithms that can be used for this stage (for example the Starburst algorithm [15]), it was preferred to use a simpler method for not overloading the device's processor and preserving energy.

The method consists in taking the eyes rectangles and calculating their centroids, in order to search for a correlation between these values and the user's gaze. Because the device and the eyes are not always in a fixed relative position, when both eyes are tracked, the eyes' rectangles are unified and represented in a different coordinate system. Using this method, the eyes rectangles are calculated always from the same point of origin if the following conditions hold true:

- The rectangle is calculated for every frame and must be as precise as possible.
- The eyes must stay on the same y axis.

Once the centroids are calculated, the Gaze Detection module stores the acquired data into a csv file.

III. Experimental Results

In order to assess the performance of the system and the effectiveness of the implemented modules, the application was tested on 16 subjects. The only criteria for a subject to be suitable to participate in the evaluation process were to have a visual acuity of 20/20 and to not have any disorder that affected their vision. For this reason, a complete ofthalmologic examination was performed on all subjects before the test's execution. The evaluation process was performed while the iPhone was seated on a dock station and the subject's position was comfortably adjusted in order to achieve a distance of about 30 cm between the subject's eyes and the iPhone's screen. Moreover, the application was tested when the iPhone was held by the subject at the same distance in order to evaluate the peroformance under normal conditions of the Eye Tracker and Haar Detector modules. All of the subjects underwent the following routine:

- Application calibration to calculate CAMSHIFT's parameters
- iPhone seated: For five minutes the subject was asked to visit a web page on the iPhone and to report out loud the object (image, text etc..) she was currently looking at.
- One minute pause
- iPhone on hand: For two minutes the subject was asked to visit the same web page on the iPhone and to report out loud the object (image, text etc..) he was currently looking at.
- The reported locations were registered together with the elapsed time from the experiment's starting time.
- At the end of the session, the produced csv file was compared to the registered notes.

The obtained performance, for all the 16 users, is reported in Table I. True positives (TP) represent the number of correctly detected fixations whereas false postives (FP) and false negatives (FN) represent the number of falsely detected fixations and the number of undetected fixations, respectively. RF and DF are the number of the user reported fixations and the number of the detected fixations, respectively. The accuracy of the proposed system reached about 79% when the iPhone was docked and 53% when the user held the iPhone on hand.

IV. Conclusion

In this paper, we presented a solution for integrating eye tracking capabilities on an iPhone. In particular, by using the

TABLE I
OBTAINED RESULTS.

Mode	RF	DF	TP	FP	FN	Accuracy (%)
Docked	741	624	603	21	138	79.1
On hand	373	268	224	44	149	53.5

phone's camera and the appropriate image processing techniques, our system is able to detect and track the eyes position in real-time and interpret it in screen coordinates. While the obtained results, especially when the iPhone is docked, look promising, more sophisticated and complex algorithms should be used in order to achieve better performance in real life normal conditions. This means that more processing power is needed in order to not overload the phone's processor, and to this end, we aim at integrating a Cloud computing which can provide computational resources on demand, such as the one in [16]. Finally, given the processing power of the currently available smartphones suffice for image processing applications, other types of image processing application can be developed as well, ranging from biometric applications like [17] to media retrieval applications [18], to object detection, tracking and classification applications [19], [20], [21], to medical image analysis tools [22], [23], [24], [25] and to specific image processing methods [26].

REFERENCES

[1] S. Agrawal, I. Constandache, S. Gaonkar, R. Roy Choudhury, K. Caves, and F. DeRuyter, "Using mobile phones to write in air," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, ser. MobiSys '11. New York, NY, USA: ACM, 2011, pp. 15–28. [Online]. Available: http://doi.acm.org/10.1145/1999995.1999998

[2] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," in *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, march 2009, pp. 1 –9.

[3] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, "A survey of mobile phone sensing," *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 140 –150, sept. 2010.

[4] M. Hayhoe and D. Ballard, "Eye movements in natural behavior," *Trends in Cognitive Sciences*, vol. 9, no. 4, pp. 188–194, Apr. 2005. [Online]. Available: http://dx.doi.org/10.1016/j.tics.2005.02.009

[5] A. B. e. a. Craig A. Chin, "Integrated electromyogram and eye-gaze tracking cursor control system for computer users withmotor disabilities," *Journal of Rehabilitation Research and Development*, vol. 45, 2008.

[6] S. Baluja and D. Pomerleau, "Non-intrusive gaze tracking using artificial neural networks," Pittsburgh, PA, USA, Tech. Rep., 1994.

[7] D. H. Yoo and M. J. Chung, "A novel non-intrusive eye gaze estimation using cross-ratio under large head motion," *Comput. Vis. Image Underst.*, vol. 98, pp. 25–51, April 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1061935.1649096

[8] A. Faro, D. Giordano, C. Spampinato, D. De Tommaso, and S. Ullo, "An interactive interface for remote administration of clinical tests based on eye tracking," in *Proceedings of the 2010 Symposium on Eye-Tracking Research &#38; Applications*, ser. ETRA '10. New York, NY, USA: ACM, 2010, pp. 69–72.

[9] A. Faro, D. Giordano, C. Pino, and C. Spampinato, "Visual attention for implicit relevance feedback in a content based image retrieval," in *Proceedings of the 2010 Symposium on Eye-Tracking Research &#38; Applications*, ser. ETRA '10. New York, NY, USA: ACM, 2010, pp. 73–76.

[10] E. Miluzzo, T. Wang, and A. T. Campbell, "Eyephone: activating mobile phones with your eyes," in *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, ser. MobiHeld '10. New York, NY, USA: ACM, 2010, pp. 15–20.

[11] H. Drewes, A. De Luca, and A. Schmidt, "Eye-gaze interaction for mobile phones," in *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology*, ser. Mobility '07. New York, NY, USA: ACM, 2007, pp. 364–371.

[12] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–511 – I–518 vol.1.

[13] P. I. Wilson and J. Fernandez, "Facial feature detection using haar classifiers," *J. Comput. Small Coll.*, vol. 21, pp. 127–133, April 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1127389.1127416

[14] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," 1998.

[15] D. Li, D. Winfield, and D. Parkhurst, "Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches," in *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, june 2005, p. 79.

[16] C.-W. Kao, C.-W. Yang, K.-C. Fan, B.-J. Hwang, and C.-P. Huang, "An adaptive eye gaze tracker system in the integrated cloud computing and mobile device," in *Machine Learning and Cybernetics (ICMLC), 2011 International Conference on*, vol. 1, july 2011, pp. 367 –371.

[17] A. Faro, D. Giordano, and C. Spampinato, "An automated tool for face recognition using visual attention and active shape models analysis," in *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, 30 2006-sept. 3 2006, pp. 4848 –4852.

[18] D. Giordano, I. Kavasidis, C. Pino, and C. Spampinato, "A semantic-based and adaptive architecture for automatic multimedia retrieval composition," in *Content-Based Multimedia Indexing (CBMI), 2011 9th International Workshop on*, june 2011, pp. 181 –186.

[19] C. Spampinato, "Adaptive objects tracking by using statistical features shape modeling and histogram analysis," in *Advances in Pattern Recognition, 2009. ICAPR '09. Seventh International Conference on*, feb. 2009, pp. 270 –273.

[20] C. Spampinato, D. Giordano, R. Di Salvo, Y.-H. J. Chen-Burger, R. B. Fisher, and G. Nadarajan, "Automatic fish classification for underwater species behavior understanding," in *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*, ser. ARTEMIS '10. New York, NY, USA: ACM, 2010, pp. 45–50.

[21] C. Spampinato, Y. heh Chen-burger, G. Nadarajan, and R. B. Fisher, "Detecting,tracking and counting fish in low quality unconstrained underwater videos," 2008.

[22] A. Faro, D. Giordano, C. Spampinato, and M. Pennisi, "Statistical texture analysis of mri images to classify patients affected by multiple sclerosis," vol. 29, pp. 272–275, 2010.

[23] A. Faro, D. Giordano, C. Spampinato, S. Ullo, and A. Di Stefano, "Basal ganglia activity measurement by automatic 3-d striatum segmentation in spect images," *Instrumentation and Measurement, IEEE Transactions on*, vol. 60, no. 10, pp. 3269 –3280, oct. 2011.

[24] D. Giordano, R. Leonardi, F. Maiorana, G. Scarciofalo, and C. Spampinato, "Epiphysis and metaphysis extraction and classification by adaptive thresholding and dog filtering for automated skeletal bone age analysis," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, aug. 2007, pp. 6551 –6556.

[25] D. Giordano, C. Spampinato, G. Scarciofalo, and R. Leonardi, "An automatic system for skeletal bone age measurement by robust processing of carpal and epiphysial/metaphysial bones," *Instrumentation and Measurement, IEEE Transactions on*, vol. 59, no. 10, pp. 2539 –2553, oct. 2010.

[26] F. Cannavo, G. Nunnari, D. Giordano, and C. Spampinato, "Variational method for image denoising by distributed genetic algorithms on grid environment," in *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006. WETICE '06. 15th IEEE International Workshops on*, june 2006, pp. 227 –232.