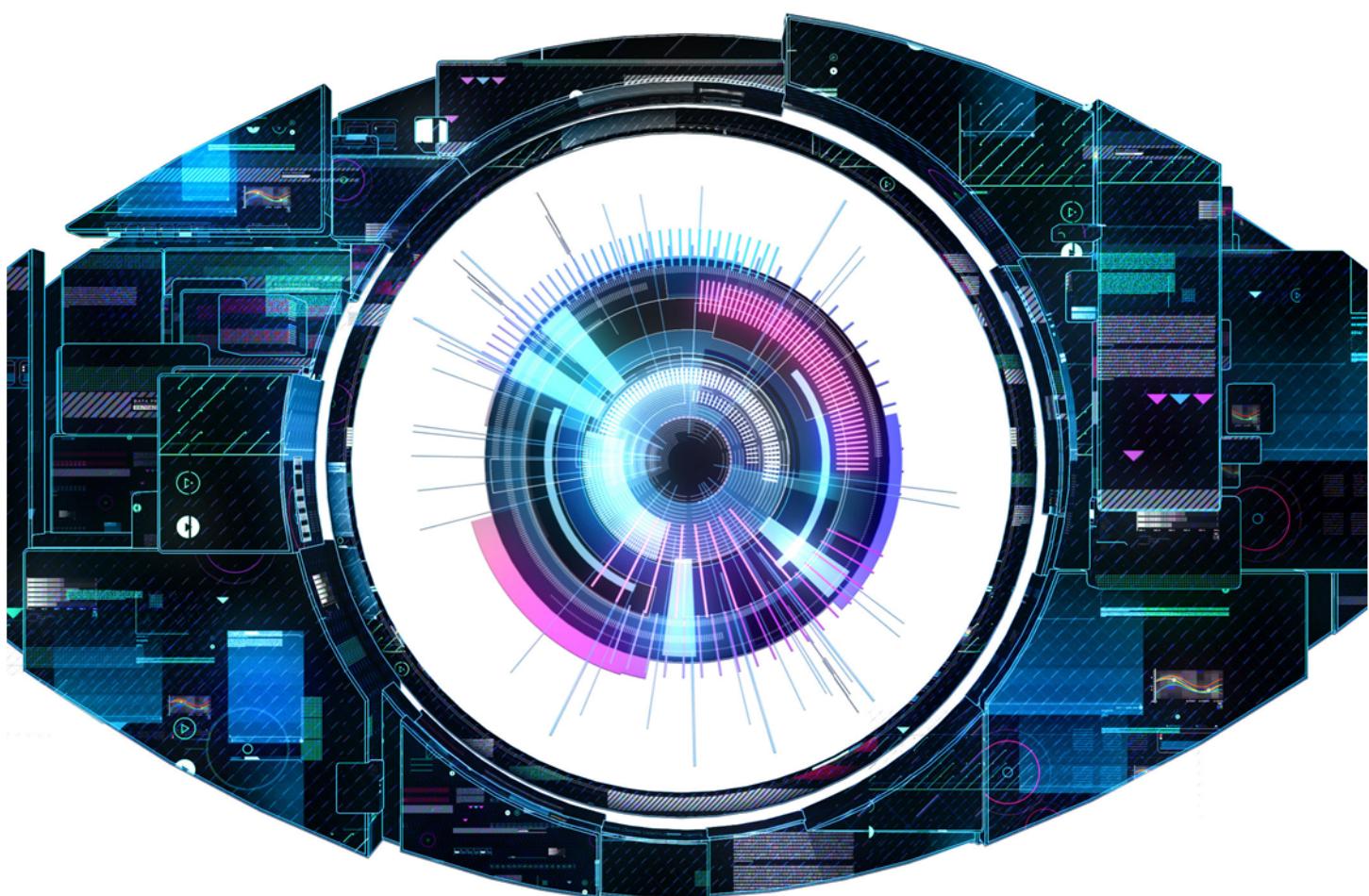


Google Glass Interaction Techniques for Deaf and Handicapped Users



Workgroup

Ashraf Saleh M. Aly
Mohamed Osama
Motaz Osama El-Shaer
Nourhan Abdelmoneam
Sherif El-Sayed Saleh
Yahia Attia Nassif

Supervisors

Prof.Dr Mohamed A. Ismail
Dr. Nagia M. Ghanem

First printing, July 2014



Contents

1	Introduction and Related Work	5
1.1	Introduction	5
1.2	Motivation	5
1.3	Related Work	6
1.3.1	Eye Tracking	6
1.3.2	Gesture Recognition	9
2	Discussion and Evaluation	13
2.1	Evaluation	13
2.2	Headset Design and Hardware	15
3	Conclusion and Future Work	17
3.1	Conclusion	17
3.1.1	Eye Tracking	17
3.1.2	Gesture Recognition	17
3.2	Future Work	18
	Index	21



1. Introduction and Related Work

1.1 Introduction

In this project we use two different HCI techniques; eye-tracking and gesture recognition. By using these techniques we aim to facilitate human computer interaction and provide a new way of accessibility for people with disabilities. Using Eye-tracking we detect the point of gaze which can be translated in some action. Gesture recognition is used to interpret hand gestures into simple actions like capturing a photo. Not to mention that gesture recognition can provide a notable advantage to the deaf since it is similar to sign language. In general we the 2 methods we are serving can be used to enhance user experience interacting with computers, specially wearable computers.

Through this book we will demonstrate 3 different eye tracking algorithms; Starburst [14], Pupil [12], and Mirt, in chapters ??, ?? and ?? respectively. Then we discuss the performance of these algorithms in chapter 2. In chapters ?? and ?? we discuss different approaches we use to recognize static and dynamic gestures. Finally we present our conclusion and future work in chapter 3.

1.2 Motivation

It is largely a software concern; when software, hardware, or a combination of hardware and software, is used to enable use of a computer by a person with a disability or impairment. Developing HCI techniques that ease interaction with computer has become a must. Traditional input methods like keyboards, mouses and even touch screens can be stumbling block in the way how persons with disability interact with computers.

Looking from a different prescriptive, wearable technology lately has been a very promising field. Ubiquitous computers are expected to be an essential part of our lifestyle in the near future. Introducing new pervasive wearable devices (i.e Google Glass, smart watches) opens a new area for development. Not only such devices have changed the way interact with computer, but also they leave the door open for new HCI

techniques. For example using voice commands is one of the most popular methods to interact with wearable gadget. Despite the simplicity of using voice commands, it might not be acceptable in some environments like libraries. Hence developing new techniques that cope together to serve a new user experience has become essential.

We believe that eye-tracking and gesture recognition based techniques can offer a new user experience and can ease the way people interact with wearable gadgets. Moreover, gesture recognition can be a very appealing way for the deaf to ease interaction with computers due to the similarity with sign language.

1.3 Related Work

1.3.1 Eye Tracking

Eye tracking is to collect data using either a remote or head-mounted 'eye tracker' connected to a computer. Most generally eye trackers include two common components: a light source and a camera. The light source(usually infra-red) is directed towards the eye. The camera tracks the reflection of the light source along with visible ocular features such as pupil. This data is usually used in estimating gaze of point.

There are many different methods for exploring eye tracking data. The most common is to analyze the visual path of one or more participants across an interface such as a computer screen. So, the presence or absence of eye data points in different screen areas can be examined. This type of analysis is used to determine which features are seen, when a particular feature captures attention, how quickly the eye moves, what content is overlooked and virtually any other gaze-related question.

Think of all the ways that you will use your eyes today. You might drive a car. You might read a magazine. You might surf the internet or play a video game or complete a training exercise or watch a movie or look at pictures on your mobile device. These are the applications of eye tracking. With very few exceptions, anything with a visual component can be eye tracked. We use our eyes almost constantly, and understanding how we use them has become an extremely important consideration in research and design.

Now days, the automotive and medical industries have applied eye tracking technology to make us safer. The fields of advertising, entertainment, packaging and web design have all benefited significantly from studying the visual behaviour of the consumer. Research with special populations has generated exciting breakthroughs in psychology. Every day, as eye tracking is used in creative new ways, the list of applications grows.

Today's Eye-tracking algorithms use two main approaches: feature-based and model-based approaches.

Feature-based approaches detect and localize image features related to position of the eye. There is an important property among all feature-based algorithm which is criteria (e.g. a threshold) needed to decide when a certain feature is present or not. Usually the determination of the criteria or the threshold left to the user. For example on the feature-based algorithm is Starburst because it depends on the corneal reflection. [14]

On the other hand, model-based approaches don't explicitly detect features but rather find the best fitting model that is consistent with the image. For example Pupil Detection algorithm is model-based approach. However, model-based approaches require searching a complex parameter space that can be fraught with local minima and thus cannot be used without a good initial guess of the model parameters. The gain in accuracy of a model-based approach is obtained at a significant cost in terms of computational speed and flexibility but it is so promise for real-time performance. [14]

Starburst

Starburst is a robust algorithm for video based eye tracking. We started with implementing Starburst Algorithm. This algorithm is more accurate than pure feature-based approaches yet is significantly less time consuming than pure model-based approaches. A validation study shows that the technique can reliably estimate eye position with an accuracy of approximately one degree of visual angle even in the presence of significant image noise.

The use of eye tracking has significant potential to enhance the quality of everyday human-computer interfaces. Two types of human-computer interfaces utilize eye-movement measures active and passive interfaces. Active interfaces allow users to explicitly control the interface through the use of eye movements. For example, eye-typing applications allow the user to look at keys on a virtual keyboard to type instead of manually pressing keys as with a traditional keyboard. Similarly, systems have been designed that allow users to control the mouse pointer with their eyes in a way that can support, for example, the drawing of pictures. Active interfaces that allow users with movement disabilities to interact with computers may also be helpful for healthy users by speeding icon selection in graphical user interfaces or object selection in virtual reality. Passive interfaces, on the other hand, monitor the user's eye movements and use this information to adapt some aspect of the display. For example, in video transmission and virtual-reality applications, gaze contingent variable-resolution display techniques present a high level of detail at the point of gaze while sacrificing level of detail in the periphery where the absence of detail is not distracting. While eye tracking has been deployed in a number of research systems and to some smaller degree consumer products, eye tracking has not reached its full potential.

Robust Realtime Pupil Tracking

In this paper, they present a real-time dark-pupil tracking algorithm designed for low-cost head-mounted active-IR hardware. Their algorithm is robust to highly eccentric pupil ellipses and partial obstructions from eyelashes, making it suitable for use with cameras mounted close to the eye. It first computes a fast initial approximation of the pupil position, and performs a novel RANSAC based ellipse fitting to robustly refine this approximation.

Pupil

Pupil is an open source platform for pervasive eye tracking and mobile gazed interaction. In this paper, they argue that affordability does not necessarily align with accessibility. They define accessible eye tracking platforms to have the following qualities: open

source components, modular hardware and software design, comprehensive documentation, user support, affordable price, and flexibility for future changes.

They have developed Pupil, a mobile eye tracking headset and an open source software framework, as an accessible, affordable, and extensible tool for pervasive eye tracking research. They explain the design motivation of the system, provide an in depth technical description of both hardware and software, and provide an analysis of accuracy and performance of the system.

Eye-based head gestures

Gestures are commonly used for interaction and are used to signify a particular command. For Eyes there are two types of gestures, namely eye and gaze gestures. Eye gestures such as wink and blinks. But actually eye gestures specially repetitive blinking for long-term use may create a feeling of nervous eye muscles.[5] Gaze gestures, on the other hand, are definable patterns of eye movements performed within a limited time interval.[11] But simple gaze gestures are not distinguishable from natural eye patterns and make unintended interaction similar to the Midas- touch problem. Complex gaze gestures consist of several simple gaze gestures are therefore needed for robust results.

Head nods and shakes are widely used in our daily conversation as a gesture to fulfill a semantic function and as conversational feedback (e.g., nodding instead of saying yes). [4]

So that is a novel for video-based head gesture recognition using eye information by an eye tracker has been proposed. The method uses a combination of gaze and eye movement to infer head gestures. Compared to other gesture-based methods a major advantage of the method is that the user keeps the gaze on the interaction object while interacting. This method has been implemented on a head-mounted eye tracker for detecting a set of predefined head gestures.[15]

Eye Tracking in Human-Computer Interaction and Usability Research

Eye-movement tracking is a method that is increasingly being employed to study usability issues in HCI contexts. So, it is a technique whereby an individual's eye movements are measured so that the researcher knows both where a person is looking at any given time and the sequence in which their eyes are shifting from one location to another. Tracking people's eye movements can help HCI researchers understand visual and display-based information processing and the factors that may impact upon the usability of system interfaces.

In this way, eye-movement recordings can provide an objective source of interface-evaluation data that can inform the design of improved interfaces. Eye movements can also be captured and used as control signals to enable people to interact with interfaces directly without the need for mouse or keyboard input, which can be a major advantage for certain populations of users such as disabled individuals. Also Eye tracking can be used in usability testing.[18]

1.3.2 Gesture Recognition

Traditional human-computer interaction devices such as the keyboard and mouse became ineffective for an effective interaction with the virtual environment applications because the 3D applications need a new interaction device. An efficient human interaction with the modern virtual environments requires more natural devices. Among them the “Hand Gesture” human-computer interaction modality has recently become of major interest. The main objective of gesture recognition research is to build a system which can recognize human gestures and utilize them to control an application. Hand gestures are a collection of movements of the hand and arm that vary from the static posture of pointing at something to the dynamic ones used to communicate with others. Recognizing these hand movements needs modeling them in the spatial and temporal domains. Hand posture is the static structure of the hand while its dynamic movement is called hand gesture and both are particularly crucial for human-computer interaction. The methods used for understanding these structures and movements are among the most classifying researches that still in progress.

Static Gesture Recognition

Many approaches have been developed for gesture recognition. Many researchers [17] and [2] have tried with different instruments and equipment to measure hand movements like gloves, sensors or wires, but in these techniques user have to wear the device which doesn't make sense in practical use. So people thought about a way of contactless gesture recognition that could be considered as a research area in Machine Vision or Computer Vision and which would be as natural as human to human interaction.

In [3],[9] and [13] approaches are divided based on the method used in it not on how it is treating the image. Approaches are divided into:

- Appearance Based Approaches.
- Model Based Approaches.
- Soft Computing Approaches.

Appearance Based Approaches:

- Based on detecting fingertips to construct the hand.
- Skin segmentation using Gaussian Model.
- Gaussian filters and freeman's algorithm to detect fingertips.
- Kalman filter for finding fingertips and their correct positions in a recursive manner.

Model Based Approaches:

- Using histogram for calculating probability for skin color.
- Genetic Algorithms approach.
- RGB and HSI color space model based algorithm for skin detection.

Soft Computing Approaches:

- Artificial Neural Networks.
- Fuzzy Logic.
- Genetic Algorithm.

There have been a number of research efforts on appearance based method in recent years. Freeman and Weissman recognized gestures for television control using normalized correlation [8]. This technique is efficient but may be sensitive to different users, deformations of the pose and changes in scale, and background. Lars and Lindberg used scale-space color features to recognize hand gestures [1]. In their method, gesture recognition method is based on feature detection and user independent while the authors showed real-time application only under uniform backgrounds.

Yikai Fang, Kongqiao Wang proposed "A REAL-TIME HAND GESTURE RECOGNITION METHOD" [7] inspired by Lars and Lindberg's works and HandVu, they present a robust real-time gesture recognition method. The main idea is to segment hand with color and motion cues generated by detection and tracking. Then scale-space feature detection method is used to recognize hand gestures. They partitioned their work into 3 modules as follows:

1. Firstly hand detection with Adaboost is used to trigger tracking and recognition.
2. Then Adaptive hand segmentation is executed during detection and tracking with motion and color cues.
3. Finally, scale-space features detection is applied to find palm-like and finger-like structures. Hand gesture type is determined by palm-finger configuration.

Serra, Giuseppe, Lorenzo and Benedetti proposed the paper “Hand Segmentation for Gesture Recognition in EGO-Vision” [20]. In their paper, they are interested in exploring the usage of ego-vision devices for cultural heritage domain: the museum experience, for example, could be enhanced by developing innovative human-machine interfaces such as new kinds of self-guided tour that can integrate information from the local environment, Web and social medias. They divided their work into two main contributions:

- define a novel hand segmentation algorithm, uses a super-pixel features, and integrate not only illumination invariance but also temporal and spatial consistency improves the state-of-the-art results in the publicly available EDSH dataset.
- Develop of a gesture recognition algorithm based on Exemplar SVM technique that, even with a few positive samples, permits to reach competitive results.

Our Work is summarized by implementing the following approaches for static gestures as detailed in chapter ??:

- Gridding Algorithm: A simple idea for Recognizing hand gestures based on their binary representation as described in [16].
- Predefined Features Extraction: More complicated approach. where, hand gesture is represented by a collection of features that summarize its properties as described in [23].
 1. Mean an variance of hand grey image.
 2. Hand area and perimeter.
 3. Hand orientation.
 4. Orientation Histogram: Avoid sensitivity to lighting conditions.
 5. Radial signature.

- **Bag-Of-Features:** SIFT features are extracted from images, then quantized using K-means clustering in order to be classified later.

In all static approaches, Classifiers are trained by the extracted Features. We used three classification techniques for evaluation (K-NN, Naive-Bayes, and SVM). Then, we agreed that SVM is most suitable technique for our gestures as detailed in chapter ??.

Dynamic Gestures Recognition

Hidden Markov Models (HMMs) are used in [6] for continuous gesture recognition. HMMs are statistical models capable of modeling spatio-temporal patterns where the same gesture can differ in shape and duration.

This approach assumes that the hand has been detected and its articulated motion is tracked. Although they have delivered promising results, the robustness of these approaches is dependent on the prior success of (frequently challenging) hand detection and motion tracking. In addition, it is both data intensive and computationally difficult to train these models before they can be applied in recognition.

The approach proposed by Xiaohui, Gang, in [21] visual representation for hand motions based on the motion divergence fields, which can be normalized to gray-scale images. Salient regions such as Maximum Stable Extremal Regions(MSER) are then detected on the motion divergence maps. From each detected region, a local descriptor is extracted to capture local motion patterns.

For our HCI purposes, we need relatively less computational methods suitable for applications on devices with limited capabilities. So, in our project, the second approach that estimate motion direction[21] is adapted for recognizing dynamic gestures as described in chapter ??.



2. Discussion and Evaluation

2.1 Evaluation

In the previous chapters we discussed Starburst [14], Pupil [12], and MIRT eye tracking algorithms. Both Starburst and Pupil use infrared spectrum imaging. On the other hand MIRT is using visible spectrum imaging. Using visible spectrum is more complicated due to uncontrolled lighting conditions, which contain multiple specular and diffuse components. Capturing infrared images requires either using infrared cameras or using infrared lighting source with a traditional camera. For example [10] describes how to hack a traditional USB webcam to capture infrared images by removing the infrared filter and adding an infrared LED on camera chip.

In MIRT we use visible spectrum which frees us from using special hardware. In visible spectrum images the corneal limbus (usually referred as limbus) is the strongest feature contour in the image, rather than the pupil which is the strongest feature contour when using infrared imaging.

To evaluate our implementation of Starburst and Pupil we used the the benchmark dataset by Swirski et al. [22]. Figure 2.1 shows performance comparison between our implementation of Starburst described in chapter ?? and our implementation of Pupil algorithm ???. As the figure indicates the performance of Starburst is relatively low compared to Pupil. The fall back in Starburst performance mainly results from the initial guess of the pupil center. A notable observation during the evaluation is that Starburst usually drops many frames until it detects the pupil, whenever the pupil is first detected the detection rate in successive frames increase significantly. We tried to short circuit the Starburst initial estimate of pupil center (image center) with the actual pupil location in the first frame but the detection rate wasn't much affected.

Moving to Pupil it can be clearly seen that Pupil overcomes Starburst. Pupil achieves about 48% detection rate when evaluated using the complete dataset. If we exclude the dataset p1-right, that contains eye images recorded at the most extreme angles, the detection rate rise to 64%.

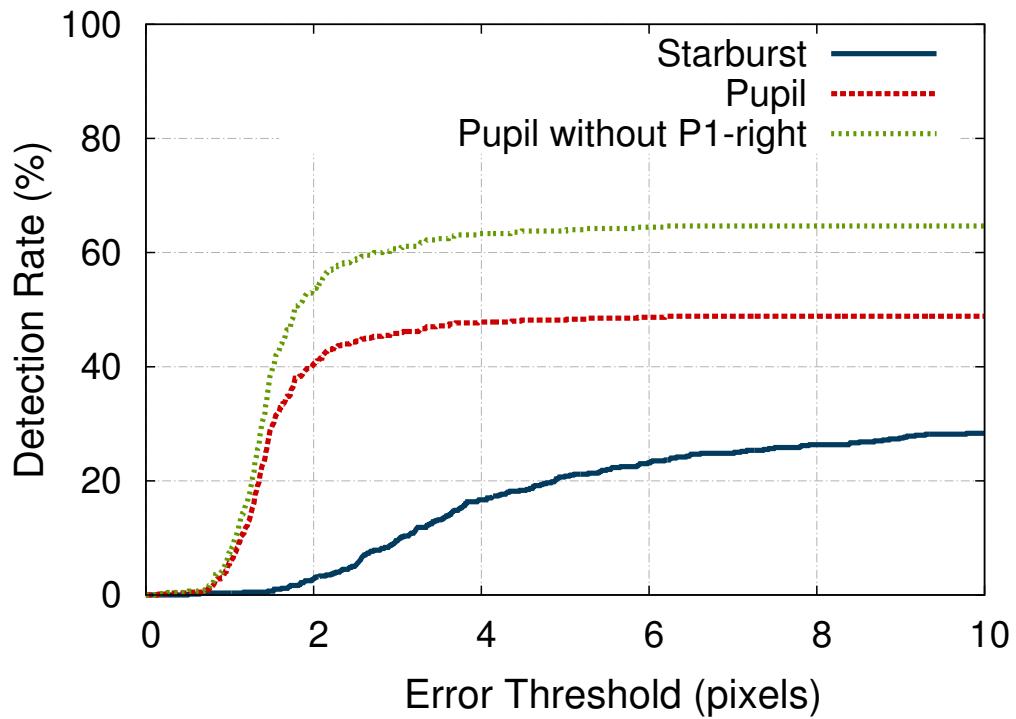


Figure 2.1: Comparison of pupil detection rate between our implementation of Pupil algorithm and our implementation of Starburst algorithm using benchmark dataset by Swirski et al.

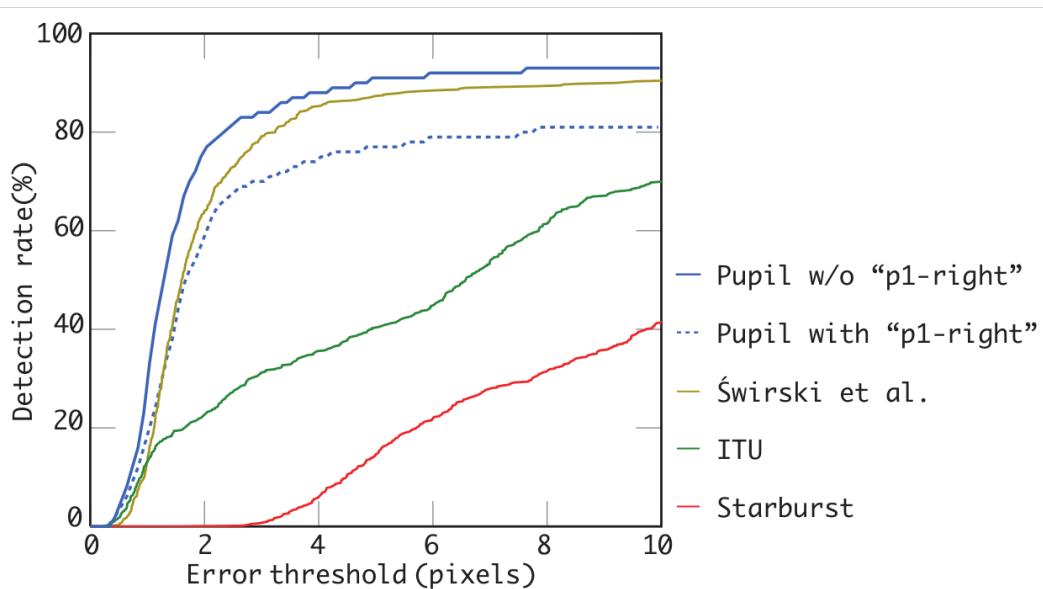


Figure 2.2: Comparison of pupil detection rate for Pupil's algorithm, the stock algorithm proposed by Swirski et al., the ITU gaze tracker and Starburst. Figure from Pupil [14]

Figure 2.2 shows the evaluation on the same dataset mentioned in Pupil paper [12]. This figure shows the performance difference between our implementation of Starburst and Pupil and actual implementations of both algorithms. Starburst we almost achieved the same performance level. On the other hand our implementation of pupil is about 20% less accurate than the actual implementation.

2.2 Headset Design and Hardware

In our system we build a cheap (do it yourself - DIY) head mounted eye-tracker. Figure 2.3 shows the prototype of our eye tracker. We used a traditional glasses frame after removing the lenses. Then we attached an internal laptop webcam to the frame. The webcam we used is a USB HP webcam. The chip comes with 2 integrated microphones, hence the camera chip has 6 pins while the USB has 4 wires. Correct pinout can be found at [19].

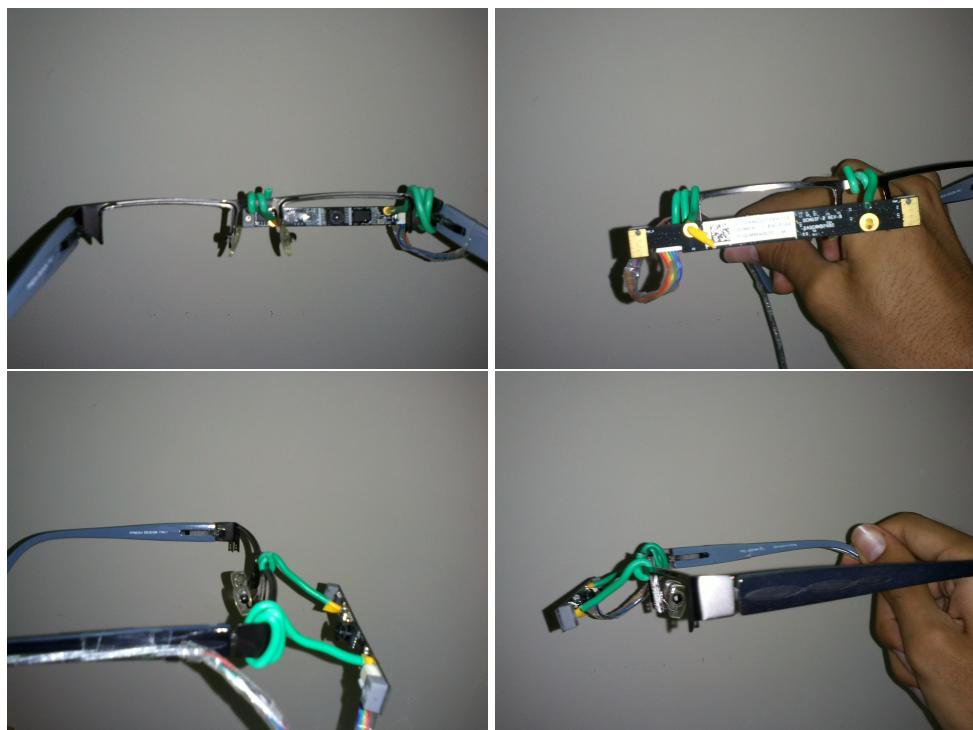


Figure 2.3: Prototype of our cheap eye-tracking head set

Conclusion

Eye Tracking

Gesture Recognition

Future Work

3. Conclusion and Future Work

3.1 Conclusion

Though this book we developed two HCI techniques that offer a new experience interacting with wearable smart gadgets. It is largely a software concern; when software, hardware, or a combination of hardware and software, is used to enable use of a computer by a person with a disability or impairment. We believe that that the techniques we discussed in this book will ease the accessibility of wearable computers.

3.1.1 Eye Tracking

Using eye tracking we implemented different algorithms; Starburst [14] and Pupil [12]. Starburst combines feature-based and model-based approaches to achieve a good trade off between run-time performance and accuracy for dark-pupil infrared illumination. The goal of the algorithm is to extract the location of the pupil center and estimate point of gaze. Pupil maintains the same goal, however pupil is more robust and accurate. A notable remark is that both algorithms locate the dark pupil in the IR illuminated eye camera image. Hence both algorithms can not be used outdoors in daytime due to the ambient infrared illumination.

Then we proposed our novel eye-tracking method; MIRT: Morphological Based Iris Tracking. Unlike the previously discussed algorithms, MIRT uses visible imaging to track the iris. We didn't try MIRT in outdoor environments. We expect that the algorithm shall work outdoors after some minor modifications since we use visible spectrum imaging using traditional camera.

3.1.2 Gesture Recognition

We implemented different algorithms for Static Gestures; Gridding, Predefined Feature Extraction, SIFT and Bag Of Words. Gridding gives higher accuracy with lower running time. But it has high sensitivity to dataset used for training, and light conditions.

Predefined Feature Extraction describe gesture with many features related to gesture size,

its orientation, HOG, and Radial signature. It gives higher accuracy with the advantage of the in-sensitivity to changes in pixel's intensities due to variation in light conditions.

SIFT and Bag of Feature Method gives a good accuracy, However it is the slowest one in running time compared to the other approaches.

Hand gesture recognition is a difficult problem and the current work is only a small step towards achieving the results needed in the field. The results obtained also proved that the feature selection and data preparation phase is an important one, especially with low resolution images.

3.2 Future Work

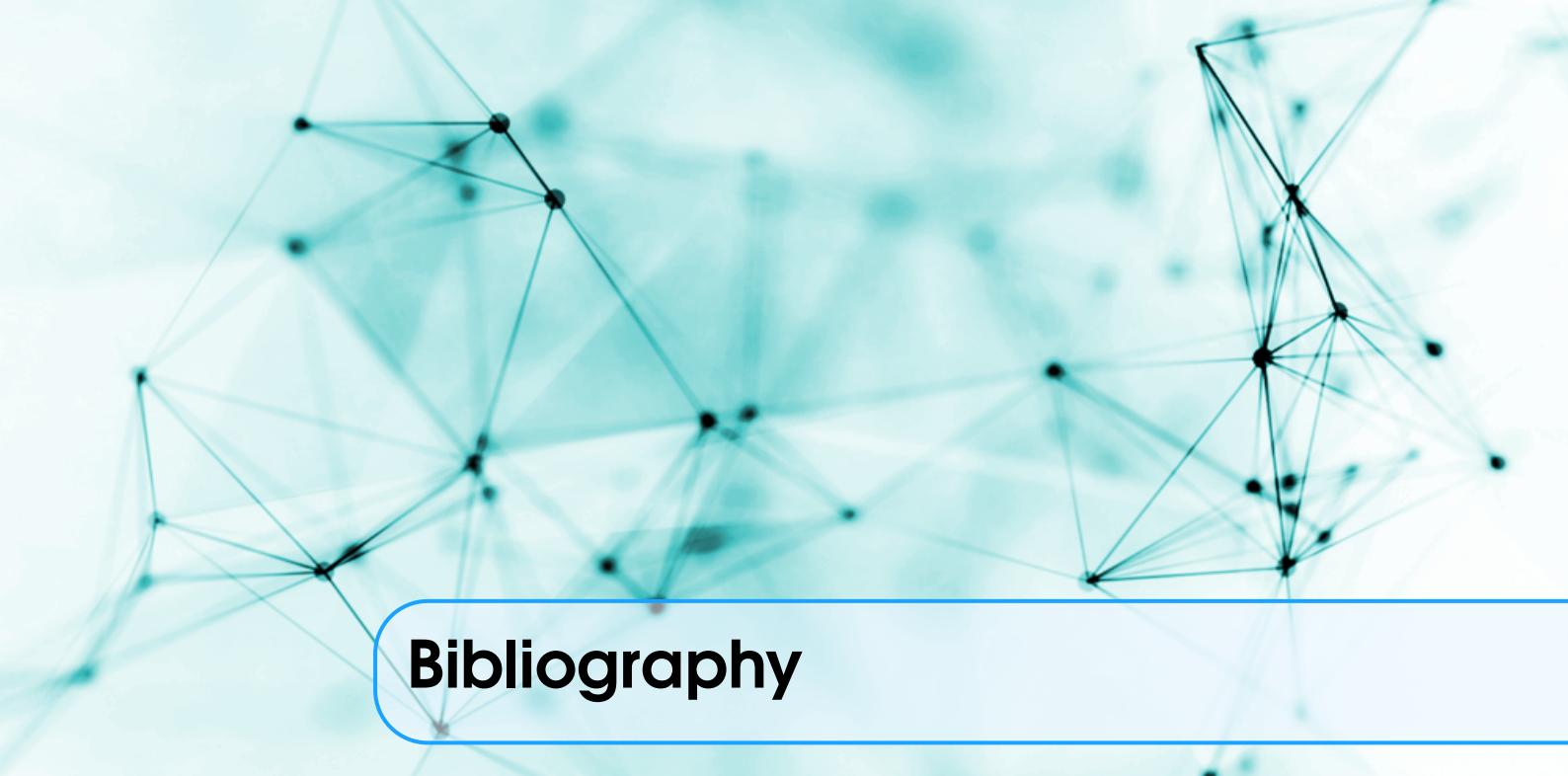
A number of improvements to MIRT algorithm can be made. For example the algorithm could be updated to work in outdoor environments. Moreover the algorithm implementation can be produced as a package that offer a standard API so as to be integrated in other applications. Speaking of the head set, we developed our cheap eye-tracking headset. During development we used a custom hand made prototype. We are willing to design the headset such that parts that hold the camera to the frame could be 3D printed.

Extending current Bag-Of-Features to dynamic gesture recognition, which is done by extracting local spatio-temporal features from a sequence of frames. Local space-time features have recently become a popular video representation for action recognition as in [24].

Bag-Of-Features Video Representation

- Videos are treated as Spatio-Temporal volumes:
 1. Salient local regions are extracted using an interest point detector.
 2. The video content in each of the detected regions is described with local features.
 3. Orderless distributions of features are computed for the entire video sample.
- 3D-Harris Detector: Salient motion is detected using a multi-scale 3D generalization of the Harris operator.
- HOF: Histogram of flow.
- HOG: Hostogram of gradient.

Last but not least, the whole system can be integrated to work with hand held smart devices. We already began to migrate to Android. No to mention that most wearable computers are Android based (i.e Google glass). In Google I/O 2014, Google released a development kit for wearable gadgets. Our system can be modified to work with the new development kit.



Bibliography

- [1] Lars Bretzner, Ivan Laptev, and Tony Lindeberg. “Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering”. In: *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on.* IEEE. 2002, pp. 423–428.
- [2] Ankit Chaudhary et al. “Intelligent approaches to interact with machines using hand gesture recognition in natural way: a survey”. In: *arXiv preprint arXiv:1303.2292* (2013).
- [3] Ankit Chaudhary et al. “Intelligent approaches to interact with machines using hand gesture recognition in natural way: a survey”. In: *arXiv preprint arXiv:1303.2292* (2013).
- [4] Charles Darwin. *The expression of the emotions in man and animals*. Oxford University Press, 1998.
- [5] Heiko Drewes. “Eye gaze tracking for human computer interaction”. PhD thesis. lmu, 2010.
- [6] Mahmoud Elmezain et al. “A hidden markov model-based continuous gesture recognition system for hand motion trajectory”. In: *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on.* IEEE. 2008, pp. 1–4.
- [7] Yikai Fang et al. “A real-time hand gesture recognition method”. In: *Multimedia and Expo, 2007 IEEE International Conference on.* IEEE. 2007, pp. 995–998.
- [8] William T Freeman and Craig Weissman. “Television control by hand gestures”. In: *Proc. of Intl. Workshop on Automatic Face and Gesture Recognition.* 1995, pp. 179–183.
- [9] Haiying Guan, Rogério Schmidt Feris, and Matthew Turk. “The isometric self-organizing map for 3d hand pose estimation”. In: *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on.* IEEE. 2006, pp. 263–268.

- [10] Dapper Vision Inc. *WearScript EyeTracking - Headset*. <http://www.wearscrip.com/en/latest/eyetrackin> 2014.
- [11] Howell Istance et al. “Designing gaze gestures for gaming: an investigation of performance”. In: *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*. ACM. 2010, pp. 323–330.
- [12] Moritz Kassner, William Patera, and Andreas Bulling. “Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-based Interaction”. In: *arXiv preprint arXiv:1405.0006* (2014).
- [13] Makoto Kato, Yen-Wei Chen, and Gang Xu. “Articulated hand tracking by pca-ica approach”. In: *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*. IEEE. 2006, pp. 329–334.
- [14] Dongheng Li, David Winfield, and Derrick J 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*. IEEE. 2005, pp. 79–79.
- [15] Diako Mardanbegi, Dan Witzner Hansen, and Thomas Pederson. “Eye-based head gestures”. In: *Proceedings of the symposium on eye tracking research and applications*. ACM. 2012, pp. 139–146.
- [16] Adam Goss. Ming Cai. “Hand Gesture Recognition and Classification”. In: 2013.
- [17] Vladimir I Pavlovic, Rajeev Sharma, and Thomas S. Huang. “Visual interpretation of hand gestures for human-computer interaction: A review”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19.7 (1997), pp. 677–695.
- [18] Alex Poole and Linden J Ball. “Eye tracking in HCI and usability research”. In: *Encyclopedia of human computer interaction* 1 (2006), pp. 211–219.
- [19] Ashraf Saleh. *codingrush blog: laptop internal webcam to USB webcam*. <http://codingrush.wordpress.com/internal-webcam-to-usb-webcam/>. 2014.
- [20] Giuseppe Serra et al. “Hand segmentation for gesture recognition in EGO-vision”. In: *Proceedings of the 3rd ACM international workshop on Interactive multimedia on mobile & portable devices*. ACM. 2013, pp. 31–36.
- [21] Xiaohui Shen et al. “Dynamic hand gesture recognition: An exemplar-based approach from motion divergence fields”. In: *Image and Vision Computing* 30.3 (2012), pp. 227–235.
- [22] Lech Świrski, Andreas Bulling, and Neil Dodgson. “Robust real-time pupil tracking in highly off-axis images”. In: *Proceedings of the Symposium on Eye Tracking Research and Applications*. ACM. 2012, pp. 173–176.
- [23] Paulo Trigueiros, Fernando Ribeiro, and Luis Paulo Reis. “A comparison of machine learning algorithms applied to hand gesture recognition”. In: *Information Systems and Technologies (CISTI), 2012 7th Iberian Conference on*. IEEE. 2012, pp. 1–6.
- [24] Heng Wang et al. “Evaluation of local spatio-temporal features for action recognition”. In: *BMVC 2009-British Machine Vision Conference*. 2009.



Index

C

Conclusion 19
Conclusion and Future Work 19

D

Dynamic Gestures Recognition 13

E

Evaluation 15
Eye Tracking 8, 19
Eye Tracking Discussion and Evaluation
15

F

Future Work 20

G

Gesture Recognition 11, 19

H

Headset Design and Hardware 17

I

Introduction 7

M

Motivation 7

P

Pupil 9

R

Related Work 8
Robust Realtime Pupil Tracking 9

S

Starburst 9
Static Gesture Recognition 11