

# openEyes: a low-cost head-mounted eye-tracking solution

Dongheng Li, Jason Babcock, and Derrick J. Parkhurst\*  
The Human Computer Interaction Program  
Iowa State University, Ames, Iowa, 50011

## Abstract

Eye tracking has long held the promise of being a useful methodology for human computer interaction. However, a number of barriers have stood in the way of the integration of eye tracking into everyday applications, including the intrusiveness, robustness, availability, and price of eye-tracking systems. To lower these barriers, we have developed the openEyes system. The system consists of an open-hardware design for a digital eye tracker that can be built from low-cost off-the-shelf components, and a set of open-source software tools for digital image capture, manipulation, and analysis in eye-tracking applications. We expect that the availability of this system will facilitate the development of eye-tracking applications and the eventual integration of eye tracking into the next generation of everyday human computer interfaces. We discuss the methods and technical challenges of low-cost eye tracking as well as the design decisions that produced our current system.

**CR Categories:** H.5.2 [Information Interfaces and Presentation]: User Interfaces—Interaction styles

**Keywords:** video-based eye-tracking, human computer interaction, consumer-grade off-the-shelf parts

## 1 Introduction

Eye tracking has been used for close to a century as a tool to study the cognitive processes of humans performing a wide variety of tasks ranging from reading to driving (for review see [Duchowski 2002]). Only more recently, has the potential integration of eye movements in human computer interfaces been seriously investigated (initially by [Jacob 1991]). In spite of the promise of this research, eye-tracking technology is not used in everyday computer interfaces. The absence of eye tracking in consumer-grade human computer interfaces can be attributed to the significant intrusiveness, lack of robustness, low availability, and high price of eye-tracking technology.

Much research has indicated the potential of eye tracking to enhance the quality of everyday human-computer interfaces. For example, eye-tracking interfaces have been implemented that allow users to directly control a computer using only eye movements. In one such application, eye typing, users with movement disabilities can type by looking at keys on a virtual keyboard instead of providing manual input [Majaranta and Raiha 2002]. 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 [Hornof et al. 2004]. These interfaces have also been helpful for healthy users by speeding icon selection in graphical user interfaces [Sibert and Jacob 2000] or object selection in virtual reality [Tanriverdi and Jacob 2000]. Furthermore, eye tracking promises to enhance the quality of video-transmission and virtual-reality applications by selectively presenting a high level of detail at the point of gaze while sacrificing level of detail in the periphery where its absence is not distracting [Parkhurst and Niebur 2002; Parkhurst and Niebur 2004].

Although numerous eye-tracking technologies including electro-oculography, magnetic eye-coil tracking and video-based tracking, have been available for many years [Young and Sheena 1975], these techniques have all been limited in a number of important ways. The primary limitation, especially relevant for application in consumer products, is the invasiveness of eye-tracking systems. Some techniques require equipment such as special contact lenses, electrodes, chin rests, bite bars or other components that must be physically attached to the user. These invasive techniques can quickly become tiresome or uncomfortable for the user. Video-based techniques have minimized this invasiveness to some degree. Video-based techniques capture an image of the eye from a camera either mounted on head gear worn by the user or mounted remotely. The recent miniaturization of video equipment has greatly minimized the intrusiveness of head-mounted video-based eye trackers [Pelz et al. 2000; Babcock and Pelz 2004]. Furthermore, remotely located video-based eye-tracking systems can be completely unobtrusive (e.g., see [Haro et al. 2000; Morimoto et al. 2002]), although at some cost to the robustness and quality of the eye tracking.

The cost and availability of eye-tracking technology also limits its application. Until only recently, eye trackers were custom made upon demand by a very few select production houses. Even today, eye-tracking systems from these sources range in price from 5,000 to 40,000 US dollars, and thus limit their application to high-end specialty products. It is important to note however that the bulk of this cost is not due to hardware, as the price of high-quality camera technology has dropped precipitously over the last ten years. Rather the costs are mostly associated with custom software implementations, sometimes integrated with specialized, although inexpensive, digital processors, to obtain high-speed performance. Moreover, customer support can also contribute significantly to these final purchase prices.

It is clear that to reap the potential benefits of eye tracking in everyday human-computer interfaces, the development of inexpensive and robust eye-tracking systems will be necessary. Towards this goal, we have undertaken the development of an eye tracker that can be built from low-cost off-the-shelf components. We have iterated through a number of system designs and in this paper we describe these systems as well our successes and failures in this process. We have arrived at a minimally invasive, digital head-mounted eye tracker capable of an accuracy of approximately one degree of visual angle. Aside from a desktop or laptop computer to process video, the system costs approximately 350 US dollars to construct. Our analysis also indicates the need for the development of widely available, reliable and high-speed eye-tracking algorithms that run on general purpose computing hardware. Towards this goal, we have also developed a novel video-based eye-tracking algorithm.

\*Email: derrick@iastate.edu

Copyright © 2006 by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions Dept, ACM Inc., fax +1 (212) 869-0481 or e-mail [permissions@acm.org](mailto:permissions@acm.org).

ETRA 2006, San Diego, California, 27–29 March 2006.

© 2006 ACM 1-59593-305-0/06/0003 \$5.00

We refer to our eye tracker as the openEyes system because we make freely available both the hardware constructions plans and the software that implements the algorithm. The open-hardware design is available in a detailed step by step tutorial on our website (<http://hcvl.hci.iastate.edu/openEyes>). The software is also freely available in the form of an open-source package licensed under the General Public License. We hope that the availability of software, ease of construction and open design of the openEyes system will enable interface designers to begin exploring the potential benefits of eye tracking for human computer interfaces. Furthermore, the flexibility provided by our open approach should allow system designers to integrate eye tracking directly into their system or product. We expect that the availability of the openEyes system will significantly enhance the potential that eye tracking will be incorporated into the next generation of human-computer interfaces.

## 2 Video-based eye tracking

Two types of imaging approaches are commonly used in eye tracking, visible and infrared spectrum imaging [Hansen and Pece 2005]. Visible spectrum imaging is a passive approach that captures ambient light reflected from the eye. In these images, it is often the case that the best feature to track is the contour between the iris and the sclera known as the limbus. The three most relevant features of the eye are the pupil - the aperture that lets light into the eye, the iris - the colored muscle group that controls the diameter of the pupil, and the sclera, the white protective tissue that covers the remainder of the eye. Visible spectrum eye tracking is complicated by the fact that uncontrolled ambient light is used as the source, which can contain multiple specular and diffuse components. Infrared imaging eliminates uncontrolled specular reflection by actively illuminating the eye with a uniform and controlled infrared light not perceivable by the user. A further benefit of infrared imaging is that the pupil, rather than the limbus, is the strongest feature contour in the image (see e.g., Figure 1(l)). Both the sclera and the iris strongly reflect infrared light while only the sclera strongly reflects visible light. Tracking the pupil contour is preferable given that the pupil contour is smaller and more sharply defined than the limbus. Furthermore, due to its size, the pupil is less likely to be occluded by the eye lids. The primary disadvantage of infrared imaging techniques is that they cannot be used outdoors during daytime due to the ambient infrared illumination.

Infrared eye tracking typically utilizes either a bright-pupil or dark-pupil technique (however see [Morimoto et al. 2002] for the combined use of both bright and dark pupil techniques). The bright-pupil technique illuminates the eye with a source that is on or very near the axis of the camera. The result of such illumination is that the pupil is clearly demarcated as a bright region due to the photoreflective nature of the back of the eye. Dark-pupil techniques illuminate the eye with an off-axis source such that the pupil is the darkest region in the image. While the sclera, iris and eye lids all reflect relatively more illumination. In either method, the first-surface specular reflection of the illumination source off of the cornea (the outer-most optical element of the eye) is also visible. The vector between the pupil center and the corneal reflection center is typically used as the dependent measure rather than the pupil center alone. This is because the vector difference is less sensitive to slippage of the head gear - both the camera and the source move simultaneously.

Both visible spectrum and infrared spectrum imaging techniques have been applied in the context of remote video-based eye tracking. The single most attractive reason for using a remote eye-tracking system is that its use can be completely unobtrusive. How-

ever, a limitation of a remote system is that it can only track eye movements when the user is within a relatively confined area of operation. The design of remote eye-tracking systems must consider the three way trade-off between cost, flexibility and quality. For example, the flexibility to track eye movements over a wide area can be improved by using a pan-tilt camera, but such cameras are quite expensive. Furthermore, the quality of eye tracking can be improved by capturing a high-resolution image of the eye using a zoom camera, with the trade-off of a reduced operational area and higher cost. Although, there are a number of promising remote eye tracking approaches (e.g., see [Haro et al. 2000; Morimoto et al. 2002]), it currently appears that a head-mounted system has a greater potential to achieve a reasonable compromise between all of these factors.

The innovative work of Jeff Pelz and colleagues [Pelz et al. 2000; Babcock and Pelz 2004] at the Rochester Institute of Technology (RIT) on the construction of low-cost minimally invasive head-mounted eye trackers is particularly noteworthy. In their system, analog cameras are mounted onto safety glasses (in a similar configuration as that shown in Figure 1(a)) and video of the user's eye and the user's field of view are interleaved in a single interlaced video frame and recorded using a mini-DV camcorder stowed in a backpack. Point of gaze computation is then performed off-line using proprietary hardware and software purchased from a production house. Given our goal to integrate eye movement measurements into human computer interfaces, this dependence on high-cost proprietary equipment is a serious limitation of their approach. Furthermore, the off-line nature of the system is another limitation as some degree of real-time performance will be necessary in many HCI applications. However, their innovation in head gear design and low-cost approach is laudable and we adopt both in our own efforts.

## 3 The openEyes system

The motivation for this research stems from the recognition in the eye-tracking and human computer interaction communities of the need for robust inexpensive methods for eye tracking. The openEyes system addresses this need by providing both an open-hardware design and a set of open-source software tools to support eye tracking. The open-hardware design details a procedure to construct a minimally invasive, digital head-mounted eye tracker from low-cost off-the-shelf components capable of an accuracy of approximately one degree of visual angle. The open-source software tools provide a ready to use implementation of a robust eye-tracking algorithm that we developed. This implementation can be run on general-purpose hardware and thus can be widely employed in everyday human-computer interfaces.

## 4 Open-hardware design

In this section, the design of the openEyes eye-tracking hardware is described in a way that shows the evolution of the system to its final form. This approach provides insight into principles, decisions, benefits and limitations of the system. The description is limited to the most important construction details given that an extensive description of the system construction is available on the openEyes website. This description includes a step by step tutorial on head-gear construction as well as a detailed parts list accompanied by hyperlinks to vendor web sites.

The first design consideration after having chosen to use a head-mounted system was the configuration of the head gear. The most significant issue was where to mount the cameras. Given that until recently cameras were quite large, a number of commercial systems place the cameras either above the eyes, on top of the head or above the ears, primarily for ergonomic reasons. These configurations necessitate the integration of a mirror or prism in the camera's optical path. Instead of taking this approach, we adopt the solution developed at RIT of placing the eye camera on a boom arm such that there is a direct line of sight between the camera and the eye (see Figure 1(a)). The primary advantage of this design is that it avoids the need for expensive optical components. Half-silvered infrared-reflecting mirrors or prisms can be expensive and glass components can pose significant danger of eye damage in near-eye applications. We were unable to locate an inexpensive source of half-silvered infrared-reflecting mirrors constructed of plexiglass. Such mirrors are typically used in commercial systems but must be purchased in bulk to achieve a reasonable price. The primary disadvantage of a boom arm design is that a portion of the visual field is blocked by the camera and the armature. Given the small extent and peripheral positioning of the camera/boom, we view this as an acceptable compromise. In fact, because these components are attached to the head gear and thus static in the user's visual field, they are easily ignored just as the frames of normal eye glasses are ignored.

The second design consideration concerned finding a way to capture and process digital images for real-time eye tracking. The RIT system used inexpensive low-resolution CMOS cameras to generate analog video output. The cameras that they used are among the smallest available on the market and, in general, analog cameras are available in smaller sizes than digital cameras. We considered a number of analog image-capture solutions to use in combination with analog cameras, but all such solutions were overly expensive (i.e. many hundreds of dollars), would require considerable fabrication expertise (e.g., the use of an A/D chip), or were not applicable in the mobile context (i.e. required a desktop computer). We therefore considered only solutions that utilized digital cameras with a readily available means of capture to a standard laptop computer. For example, a number of small inexpensive USB web cameras were investigated but the resolution and frame rates were limited by the bandwidth of USB. We failed to find any inexpensive USB 2.0 compatible web cameras that utilized the full bandwidth of USB 2.0. Ultimately, we settled upon using inexpensive IEEE-1394 web cameras. The bandwidth of these cameras (400Mbit/sec) is sufficient to capture video simultaneously from two cameras at a resolution of 640x480 pixels with a frame rate of 30hz. Two additional benefits of IEEE-1394 cameras include the fact that cameras on the same bus will automatically synchronize themselves and that the IEEE-1394 standard is well supported under Linux with the 1394-based DC Control Library.

We examined a number of inexpensive IEEE-1394 cameras available on the market. Initially, the Apple I-sight camera was considered because of its unique construction. The optics have an auto-focus feature and the CCD is mounted on a flat flex cable approximately one inch long that leads to the main processing board. However, after much investigation, we failed to find a way to extend this cable in a reasonable way. Any modifications would have required extremely difficult soldering of surface mount connectors. We finally settled on using the comparably priced Unibrain Fire-i IEEE-1394 web camera. One advantage of using this camera for our application is that more than one camera can be daisy chained together and thus share a single power source (see Figure 1(f)). The disadvantage of this camera is that the CCD sensor is soldered directly to processing board and without removal, the entire board would be too cumbersome to mount on a head gear. Therefore a technique was developed to detach the CCD sensor from the camera

board and solder a multi-conductor cable of some length between the board and the chip. When done carefully, the sensor remains undamaged and the lens and mount can be re-attached so that the camera functions as before. Note, however, that a degree of noise is induced in the captured images (see Figures 1(l&m)). Much of the work subsequent to this initial design decision has been to find a way to reduce this noise (see below).

#### 4.1 Generation 1

The first generation prototype is shown in Figures 1(a-c) and, as can be seen, the profile is small and unobtrusive. The Sony CCD and lens mount assembly standard with the Fire-i camera were extended from the camera processing boards and mounted on a pair of modified safety glasses which have had the plastic lenses cut mostly away. Very fine unshielded wire was used to extend the CCD and when routed above the ear and back to the processing boards mounted on the backpack, its presence was hardly noticeable. Moreover, the lightness of the lenses and boom arm did not add to the perceivable weight of the glasses when worn. The presence of the eye tracker was not overly disturbing in spite of the fact that the camera occluded a portion of the visual field.

The design of the first generation system had three major limitations. First, the CCDs for this system were removed using a soldering iron. Given the small size of the chip and the proximity of other components on the board, this was a procedure that we believe damaged the chips and/or board. Second, the thin unshielded wire lead to significant noise in the captured images when both cameras were operated simultaneously. The amount of noise was amplified when the 14 lines for each CCD were run adjacent to each other down to the processing boards on the backpack. The degree of noise was unpredictable and tended to change as the wearer shifted their head and body. The final limitation of this approach was that we employed visible spectrum imaging. Due to the low sensitivity of these consumer-grade cameras, we were often unable to image the eye with the user indoors. Furthermore, the presence of specular reflections from various ambient light sources made digitally extracting a reliable measure of eye movements particularly difficult.

#### 4.2 Generation 2

In the second generation prototype, we attempted to redress many of the limitations of the first generation prototype. Most significantly, we moved to an infrared imaging approach. As can be seen in Figures 1(e&d), we placed an infrared LED on the boom armature off-axis with respect to the eye camera. This configuration produces an illumination that allows the discrimination of the pupil from the rest of the eye. The LED was powered from a free USB port on the laptop. Unfortunately, this design decision also required a new lens mount assembly on the eye camera. The Fire-i cameras come with a small, non-standard mount and lens combination which has an infrared cut-filter coated on the sensor side of the lens that could not be removed. To solve this problem, we salvaged the somewhat larger lens mount and lens from an OrangeMicro i-Bot web camera. The infrared blocking filter was removed from this lens and replaced with an 87c Wratten filter to block visible light and allow only infrared light to pass. The image captured using infrared illumination can be seen in Figure 1(l). Note that the infrared illumination strongly differentiates the pupil from the the iris in the image. Also note the presence of a specular reflection of the LED. This is an important benefit as the corneal reflection can be tracked and used to compensate for head gear slippage.

The second major modification that we made to the system was to use shielded cables between the CCD and the processing boards in order to reduce the noise. While the noise was reduced to some degree, its presence was still noticeable and continued to depend on the positioning of the cables. Unfortunately, a second type of strong noise appeared in this system which was much more problematic although sporadic. For example, when the head gear was nudged, touched or the user turned their head abruptly, significant but transient line noise was induced. We suspected that the CCD and processing boards were damaged or that the solder joints were weak due to the de-soldering and re-soldering. Although we could still maintain a relatively ergonomic cable configuration, the cables extending over the ear were much more noticeable to the user than in the previous generation. Furthermore, the additional stiffness of the cables sometimes induced the head gear to shift when the user turned their head. To minimize this slippage of the head gear, we employed the use of an elastic head band specially designed for glasses.

### 4.3 Generation 3

Having produced a prototype that was capable of infrared eye tracking (albeit with a large degree of noise which induced frequent tracking errors), we were encouraged to proceed. Shown in Figures 1(g-i) is the third prototype which utilized the same basic design but with a number of important modifications. First, thin double-shielded cables were employed to reduce noise. These cables added a significant degree of stiffness and consequentially the only reasonably ergonomic configuration of the head gear was for the scene camera to be mounted on the left side of the glasses (see Figure 1(i)). Second, a Unibrain monochrome Fire-i board-level camera was used for the eye camera in order to take advantage of its overall greater sensitivity to infrared light. Third, we extracted the CCDs from the processing boards using a solderless technique to minimize heat damage and developed an interlocking socket assembly (see Figure 1(h)) on which to mount the CCD sensors to minimize joint stress on the chip. Together, these modifications completely eliminated the sensitivity of the camera to spurious noise during head movements or adjustments to the head gear and significantly reduced the amount of overall image noise.

Because we used the I-bot 4.5 mm lens in the second generation prototype, the portion of the image that was occupied by the eye was quite small. Given that the accuracy of eye tracking is related to the size of the eye in the image, we employed a 12mm lens in the third generation system to obtain a much closer image of the eye. While this is clearly beneficial for achieving high-accuracy eye measurements, this design decision carried consequences. First, the depth of field in the image is smaller and consequentially more attention is necessary to obtain a correct focus. Furthermore, the restricted field of view of the camera requires proper alignment that results in a greater sensitivity to head gear slippage. Depending on the particular application, the choice of a lens between 4 and 12 mm should be made based on the trade-off between accuracy and flexibility.

A socket assembly was also constructed for the LED and positioned in a more central location in order to maximize the ability to detect the corneal reflection when gaze is non-central. A scene camera with a wider field of view was also used to track a greater range of eye movements. Notably however wide field of view lenses introduce radial distortion, which if not digitally removed, can lead to reduced eye tracking accuracy (see below).

In an attempt to improve the modularity of the system, both image processing boards were housed in a single plastic case and sepa-

rated from the head gear using a single multi-pin connector that routed cables from both cameras. Unfortunately, this design decision was a serious misstep because we experienced significantly more noise than we had previously. This was due entirely to the interference between the cameras given that when only a single camera was used, the images were entirely noise free. To eliminate this problem, the image processing boards were separated into shielded metal cases and connected using shielded metal connectors.

### 4.4 Generation 4

As is shown in the Validation Section (below), the third generation prototype tracked eye movements with an accuracy of approximately 1 degree of visual angle. However, we noticed that this level of accuracy was restricted to when the system was tested at the same distance that it was calibrated. This is due to that fact that the scene camera is not in the same optical path as the tracked eye. Thus, depending on the difference between the calibrated distance and the fixated distance, the parallax between the eye and the scene camera introduces tracking error. We found that the error was tolerable only over a one-foot discrepancy between the calibrated and fixated distances for the third generation prototype. In the fourth generation prototype, the scene camera was moved from the left side of the system (6.5 inches from the tracked eye) to the right side of the system (1.5 inches from the tracked eye). Consequentially, the tolerance to discrepancies was greatly improved. We found that for discrepancies as great as two feet between the calibration and test distance, that the average error just after calibration remained under one degree of visual angle. This degree of error is appropriate for desktop eye-tracking applications. This introduced error could be further reduced by placing the scene camera directly over the tracked eye, however we decided against this configuration for ergonomic reasons.

## 5 Open-source software

A robust eye-tracking algorithm was needed for use with the openEyes hardware design due to the presence of noise caused by the low-cost hardware construction. The traditional dual-threshold algorithm, which takes a low threshold to get the pupil center and a high threshold to get the corneal reflection, was overly susceptible to this noise and resulted in extremely poor eye tracking quality. Therefore, we developed the Starburst algorithm, which combines feature-based and model-based image processing approaches [Li et al. 2005]. The algorithm has been implemented on general-purpose hardware and has been tuned for the run-time performance and accuracy necessary for everyday human-computer interfaces. We provide both a cross-platform Matlab implementation and a C implementation that runs on the Linux operating system as open-source software packages that can be downloaded from the openEyes website.

### 5.1 The Starburst algorithm

**Noise reduction** The goal of the algorithm is to extract the locations of the pupil center and the corneal reflection so as to relate the vector difference between these location to coordinates in the scene image. The algorithm begins by reducing the shot noise and line noise in the eye image. We reduce the shot noise by applying a  $5 \times 5$  Gaussian filter with a standard deviation of 2 pixels. The line noise is spurious and a normalization factor can be applied line

by line to shift the mean intensity of the line to the running average derived from previous frames.

**Corneal reflection detection** The corneal reflection is located using an adaptive brightness-thresholding technique. We lower the threshold until the ratio between the area of the largest candidate region and the average area of other regions begins to grow. The location of the corneal reflection is then given by the geometric center of the largest region in the image using the adaptively determined threshold. The radius of the corneal reflection is obtained by a model-based approach. The corneal reflection is then removed by radial intensity interpolation, meaning that for each pixel between the center and the contour, the pixel intensity is determined via linear interpolation.

**Feature detection** The pupil edge points are located using an iterative two-stage feature-based technique. We find the feature points by computing the derivatives along rays extending radially away from a starting point, until a threshold is exceeded. In the first stage, the candidate feature points are detected from a starting point. In the second stage, for each of the candidate feature points, the feature-detection process is repeated using the candidate feature points as the starting point. The second stage tends to increase ratio of the number of feature points on the pupil contour over the number of feature points not on the pupil contour. This two-stage process iterates by replacing the starting point with the center of the detected feature points until the position of the center converges.

**Ellipse fitting** Given a set of candidate feature points, the next step of the algorithm is to find the best fitting ellipse. If least-squares approach is used to fit an ellipse to all the feature points, gross errors made in the feature detection stage can strongly influence the accuracy of the results. To address this problem, we utilize the Random Sample Consensus (RANSAC) fitting paradigm [Fischler and Bolles 1981]. RANSAC is used to fit an ellipse in the presence of an unknown percentage of outliers among the candidate feature points. In detail, RANSAC is an iterative procedure that selects many small but random subsets of the feature points, uses each subset to fit an ellipse, and finds the ellipse that has the largest agreement with the entire set of candidate feature points. The parameters from this ellipse are then used to initialize a local model-based search that optimizes the fit to the image data on the contour of the ellipse.

**Calibration** To calculate the point of gaze of the user in the scene image, a mapping between locations in the scene image and an eye position must be determined. The typical procedure in eye-tracking methodology is to measure this relationship through a calibration procedure [Stampe 1993]. During calibration, the user is required to look at a number of scene points for which the positions in the scene image are known. While the user is fixating each scene point, the eye position is measured. Then a mapping between the two sets of points is generated using a polynomial mapping. The user's point of gaze in the scene for any frame can then be established using this mapping.

## 5.2 cvHAL: computer vision Hardware Abstraction Layer

cvHAL is a Linux-based open-source computer vision software package that we developed to provide an automated system for

discovery, configuration, networking of video cameras. The software allows a developer to focus on computer vision algorithm development by abstracting away from hardware-specific camera issues. cvHAL is an always-on daemon that processes requests for video streams from clients on the network. While there exists other similar software, cvHAL is targeted at the computer-vision community by implementing advanced functionality such as multiple-camera synchronization, color-format transformations and the ability to provide server-side pre-processing on video streams. A major advantage of cvHAL is that with the recent availability of low-cost gigabit networking and high-speed wireless networking, consumer-grade off-the-shelf cameras can be easily turned into "smart cameras" by connecting them to any networked computer. cvHAL provides camera abstraction for the openEyes system and can be downloaded from the openEyes website.

## 6 Validation Study

Table 1	FOV	1st	2nd	3rd
Generation 3	Wide	1.16	1.38	1.71
Generation 4	Wide	0.68	1.11	1.37
Generation 4	Narrow	0.60	1.03	1.04

An eye-tracking evaluation was conducted in order to validate the performance of the algorithm. Video was recorded from third and fourth generation prototypes while the two authors and one research assistant viewed two movie trailers presented on a laptop computer. Prior to and after viewing each trailer, the users placed their head in a chin rest and fixated a series of nine calibration marks on a white board positioned approximately 60 cm away. The evaluation was conducted twice for each user in the case of the fourth generation prototype. During the second evaluation, the wide field of view lens (111° FOV, and significant radial distortion) used on the scene camera was replaced with a narrow field of view lens (56° Field of View (FOV)) to evaluate the potential increase in eye-tracking quality attributable to using a lens without significant radial distortion.

Shown in Table are the accuracy estimates derived from the first, second and third viewings of the calibration grid separately. Accuracy is measured as the distance between the estimated point of gaze and the actual location of the calibration marks in the scene image averaged over all nine calibration points. The first viewing of the grid is used to calibrate the eye tracker. The results show that the average eye-tracking error is very low in all conditions and is easily on par with much more expensive, commercially available eye tracking systems. A small decrease in accuracy is seen over the course of the validation, which can be attributed to some slippage of the head gear. An improvement in accuracy is seen with the fourth generation prototype, especially when using a lens without significant radial distortion.

## 7 Discussion

The goal of this research was the development of a high-quality but low-cost eye tracking system that is capable of robust real-time measures of the user's point of gaze for application to desktop and mobile applications. We expect that given the combination of open-source eye-tracking software with low-cost eye-tracking hardware built from off-the-shelf components, motivated interface designers will be able to explore the potential of eye movements for improving interface design and that this will lead to an increased role for eye tracking in the next generation human computer interfaces.

A number of improvements could be readily made to improve the current system design if cost was less of a concern. First, the entire system could be made more mobile with the use of a smaller lighter-weight computer. Computers with sufficient computational power to perform eye tracking are already available in form factors that would easily fit in a shirt or jacket pocket. These computers typically cost a factor of three more than a similarly powerful laptop. Second, high resolution digital cameras are also readily available in a form factor comparable to our solution, but cost a factor of ten more than the off-the-shelf camera that we utilized. Notably, however, the superior resolution in combination with a wide field of view lens could simultaneously improve accuracy and flexibility given that there is a trade-off between the size of the eye in the image and the quality of eye tracking. Third, a higher speed camera could be employed. An issue with all low frame-rate eye-tracking systems is that point of gaze estimates during eye movements can be poor. This is due to the motion blur induced by the long CCD integration times associated with the low frame rates and low sensitivity to infrared light of off-the-shelf cameras. Fortunately, eye movements are very rapid lasting on the order of 10 milliseconds while fixations are much longer (hundreds of milliseconds). Thus only 5-10% of the captured images show the eye in motion and for many of these frames, the motion blur is small enough that an accurate estimate of the point of gaze can still be obtained. Fourth, further consideration could also be given to selecting thin and flexible cable, devising specialized electronics to remove the noise, or moving to a wireless solution. We expect that these considerations would help minimize a degree of head gear slippage and increase the overall comfort of wearing the system.

While we have made significant progress in designing a robust low-cost eye-tracking system, there still is much work to do to facilitate the integration of eye tracking into applications. We expect that this task will not necessarily be trivial but its difficulty will necessarily depend on the particular application. For example, using eye movements to monitor the attentiveness of a user through blink rate and scan path analysis would require only post processing of the eye movements data provided by our system. However, to control a cursor on a computer screen would require additional information. Because the user is free to make head movements, the relationship between the scene camera and the computer screen must be known. One way is to track the the user's head with a magnetic or optical tracker. Such a measurement would then allow the eye movements recorded in the coordinate frame of the user's head to be transformed to the coordinate frame of the monitor. A more attractive alternative that we are currently exploring is to use image processing techniques to extract the location of markers in the scene that have known location and orientation and to infer the pose and location of the scene camera. We expect that this approach will become part of the openEyes system in the future.

## 8 Acknowledgments

We wish to thank David Winfield for contributing to the hardware construction and the open-hardware design.

## References

- BABCOCK, J., AND PELZ, J. 2004. Building a lightweight eye-tracking headgear. In *Eye Tracking Research & Applications Symposium*, 109–114.
- DUCHOWSKI, A. 2002. A breadth-first survey of eye-tracking applications. *Behavior Research Methods, Instruments and Computers* 34, 4, 455–470.
- FISCHLER, M., AND BOLLES, R. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24, 6, 381–395.
- HANSEN, D., AND PECE, A. 2005. Eye tracking in the wild. *Computer Vision and Image Understanding* 98, 1, 155–181.
- HARO, A., FLICKNER, M., AND ESSA, I. 2000. Detecting and tracking eyes by using their physiological properties, dynamics, and appearance. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 163–168.
- HORNOF, A. J., CAVENDER, A., AND HOSELTON, R. 2004. Eye-draw: A system for drawing pictures with eye movements. In *ACM SIGACCESS Conference on Computers and Accessibility, Atlanta, Georgia*, 86–93.
- JACOB, R. 1991. The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Transactions on Information Systems* 9, 2, 152–169.
- LI, D., WINFIELD, D., AND PARKHURST, D. J. 2005. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In *Proceedings of the IEEE Vision for Human-Computer Interaction Workshop at CVPR*, 1–8.
- MAJARANTA, P., AND RAIHA, K. 2002. Twenty years of eye typing: systems and design issues. In *Proceedings of the symposium on Eye tracking research and applications*, 15–22.
- MORIMOTO, C., AMIR, A., AND FLICKNER, M. 2002. Detecting eye position and gaze from a single camera and 2 light sources. In *Proceedings. 16th International Conference on Pattern Recognition*, 314–317.
- PARKHURST, D., AND NIEBUR, E. 2002. Variable resolution displays: a theoretical, practical and behavioral evaluation. *Human Factors* 44, 4, 611–29.
- PARKHURST, D., AND NIEBUR, E. 2004. A feasibility test for perceptually adaptive level of detail rendering on desktop systems. In *Proceedings of the ACM Applied Perception in Graphics and Visualization Symposium*, 105–109.
- PELZ, J., CANOSA, R., BABCOCK, J., KUCHARCZYK, D., SILVER, A., AND KONNO, D. 2000. Portable eyetracking: A study of natural eye movements. In *Proceedings of the SPIE, Human Vision and Electronic Imaging*, 566–582.
- SIBERT, L., AND JACOB, R. 2000. Evaluation of eye gaze interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 281–288.
- STAMPE, D. M. 1993. Heuristic filtering and reliable calibration methods for video-based pupil-tracking systems. *Behavior Research Methods, Instruments, and Computers* 25, 2, 137–142.
- TANRIVERDI, V., AND JACOB, R. 2000. Interacting with eye movements in virtual environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 265–272.
- YOUNG, L., AND SHEENA, D. 1975. Survey of eye movement recording methods. *Behavior Research Methods and Instrumentation* 7, 397–429.

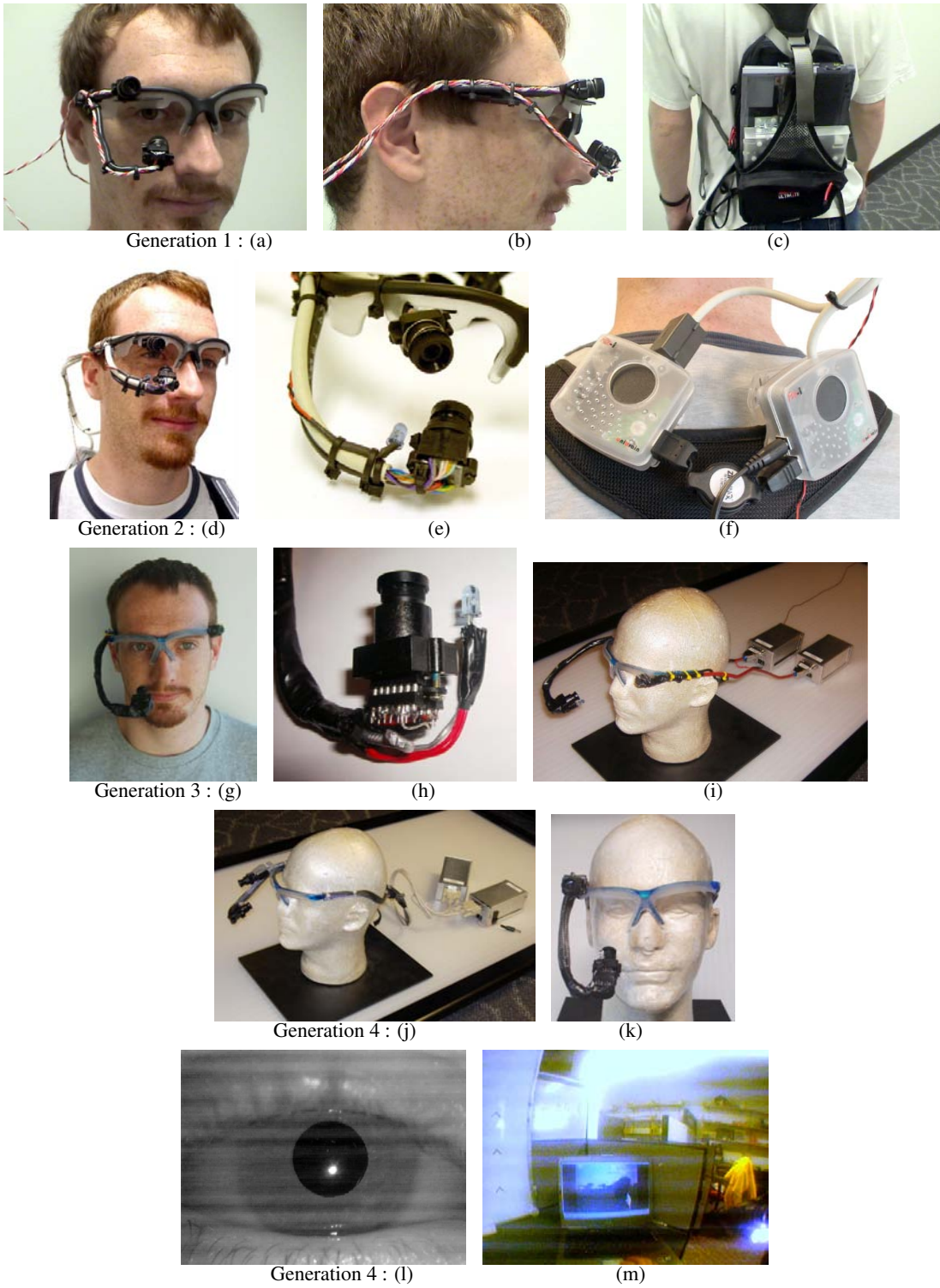


Figure 1: openEyes Hardware