Real-Time Eye Gaze Tracking With an Unmodified Commodity Webcam Employing a Neural Network

Weston Sewell
Texas State University-San Marcos
Department of Computer Science
601 University Drive
San Marcos, TX 78666-4616
weston.sewell@gmail.com

Oleg Komogortsev
Texas State University-San Marcos
Department of Computer Science
601 University Drive
San Marcos, TX 78666-4616
ok11@txstate.edu

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.
ACM 978-1-60558-930-5/10/04.

Abstract

An eye-gaze-guided computer interface could enable computer use by the seriously disabled but existing systems cost tens of thousands of dollars or have cumbersome setups. This paper presents a methodology for real-time eye gaze tracking using a standard webcam without the need for hardware modification or special placement. An artificial neural network was employed to estimate the location of the user's gaze based on an image of the user's eye, mimicking the way that humans determine where another person is looking. Accuracy measurements and usability experiments were performed using a laptop computer with a webcam built into the screen. The results show this approach to be promising for the development of usable eye tracking systems using standard webcams, particularly those built into many laptop computers.

Keywords

Eye tracker, neural network, human computer interaction, gaze estimation, webcam.

ACM Classification Keywords

H5.2. Input devices and strategies – Evaluation/methodology.

General Terms

Human Factors.

Introduction

An eye tracking system using an unmodified webcam could enable severely disabled people to interface with computers without specialized equipment. It could also enable a user interface to be sensitive to the attention of a user. Sibert and Jacob [8] showed that eye tracking interfaces are both usable and superior to mouse driven interfaces for some metrics.

Eye gaze tracking is typically achieved using specialized equipment which generally costs several times that of a personal computer. Many laptop computers and LCD monitors come equipped with built-in webcams presenting an opportunity for a much wider acceptance of eye-gaze-driven interfaces. Agustin et al. developed a webcamera based eye tracking interface but it required modification to and special placement of the camera [3]. The aim of this research was to develop an eye tracker that could use a personal computer's built-in webcam, without any modification to the camera hardware.

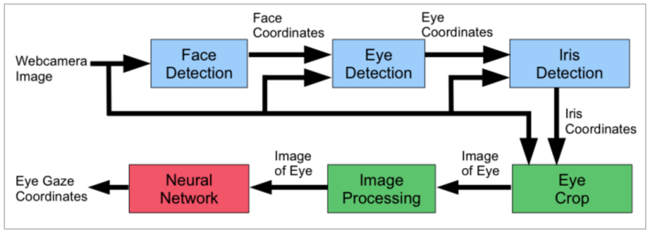

Figure 1. Eye Tracking Employing a Neural Network

Conventional methodologies for eye tracking rely on the tracker's ability to detect and track the movement of the pupil and highlights/reflections of the eye's anatomical structures. Because an unmodified webcamera does not have the resolution and/or appropriate lighting to find the pupil, a different methodology was chosen. We employed an artificial neural network (ANN) to estimate the location of the user's gaze on the screen, based on the difference in how the user's eye appeared in images from the camera. The goal of our work was to determine if this method yields real-time performance capabilities to warrant further research. The developed eye tracker was required to provide high degree of responsiveness and provide an accuracy that was within an order of magnitude of state of the art webcamera-based eye trackers.

In [5] the use of an artificial neural network for eye tracking was shown to be fruitful but the type and placement of the camera used were not discussed. The methodology proposed by Bäck [4] closely matches ours and results in an eye tracker with an accuracy of between two and four degrees. A notable distinction from our method is that Bäck chose to find points of interest within the eye and employ measurements on those points as the input to the neural network. Instead, our method employs an image of the eye directly due to a neural network's ability to operate on imprecise and noisy data.

Eye Tracking Employing a Neural Network

An overview of our method for eye tracking using a neural network is depicted in Figure 1. The image from the webcamera is first downsampled to grayscale to reduce complexity in the subsequent stages. The user's face is detected followed by detection of the eye and pupil. The image of the eye is cropped from the original image and processed to enhance detail and reduce complexity. The processed image is sent to the neural network where the eye gaze coordinates are estimated. As a simplification only one eye was used for tracking.

Face and Eye Detection

The face and eye were detected using HAAR classifiers. The HAAR classifier is a pattern matching technique that uses sums of the intensities of the pixels within a region to produce a score [9]. The scores for an object of interest (i.e. a face or an eye) are encoded hierarchically into a template with each scored region being broken into smaller scored regions. We used the

OpenCV [2] library for face and eye detection with the provided example HAAR classifier templates. The detection of the eye was constrained to the region of the top-left quadrant of the detected face rectangle to improve accuracy.

Iris Detection and Eye Cropping

Neural networks perform best when each input has an equivalent designation throughout its operation. This constraint could not be met if the image of the eye were taken from the position found by the HAAR classifier. The nature of the HAAR classifier results in the detected position of the eye moving by several pixels from one image to the next. To overcome this problem we employed the highlight within the user's iris (of the reflection of the computer screen). Iris detection was constrained to the region of the detected eye and accomplished using simple pattern matching against a template image of a pupil taken from a grayscale image of a user's eye. The template matching provisions of the OpenCV library were used which matches based on the sum of the squares of the differences between the intensities of each pixel in the template and test region. It was assumed that the iris was always present within the detected eye and the region with the lowest squared difference was selected as the location of the iris. The image of the eye to be used as input to the neural network was centered about the center of the detected iris. A size of 26x22 pixels was chosen for the image of the eye based on pictures of the eyes of various users while they sat at a normal distance (approximately 75cm) from the screen.

Eye Image Processing

Two image processing operations were performed on each image of the eye before being sent to the neural network, to improve the performance of the eye tracker. Figure 2 shows an example of the effects of the image processing. The first operation was histogram equalization which enhances contrast differences, resulting in a brightened sclera and darkened eye boundary.

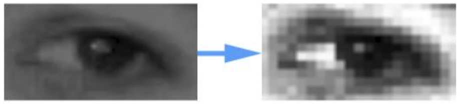

Figure 2. Example of Image Processing Effects

The second image processing operation was a downsampling (resizing) of the image to a size of 13x11 pixels using bicubic interpolation. The result was a four-fold decrease in the amount of pixels (143 vs. 572). Without the reduction of the number of pixels input to the neural network, the network required multiple minutes to complete training with an acceptable error. We chose to reduce the number of pixels such that the training time was typically less than 60 seconds.

Neural Network Gaze Tracking

The intensity of each pixel in the image of the eye was used as an input to the neural network. The network's two outputs corresponded to the X and Y locations of the user's gaze on the screen. Assuming linear separability would not pose a problem, we employed a feedforward, two-layer neural network. Previous implementations of neural network based eye trackers used networks with at least three layers [5]. The pixel intensities were converted to floating point representation for input to the network. The network outputs were floating point values representing the X and Y screen coordinates as a fraction of the screen's relevant dimension (0 representing left or top of screen and 1 representing right or bottom of screen).

Methodology

All experiments were performed on a commodity laptop computer with a built-in webcamera (dual-core 2GHz processor, 2GB of RAM, a 1280x800 pixel 13" LCD screen and a webcamera with a resolution of 1280x1024 pixels). The full resolution of the webcamera was not available to our experiments; instead the images attained were 640x480 pixels. The OpenCV library was used for interfacing with the webcamera and for all image processing tasks [2]. The FANN library was used to implement the neural network [1]. A chinrest was employed to immobilize the user's head. The lighting conditions for all experiments were similar to those found in an office or classroom.

Neural Network Training

The neural network was implemented using the Fast Artificial Neural Network Library [1]. Training was accomplished by displaying 48 visual markers on the screen for the user to look at. The markers were organized into eight columns and six rows, evenly spaced across the screen and covering the edges and corners. At each marker, eight images of the eye were recorded and associated with the X and Y coordinates of the marker. When all 48 markers had been shown the neural network was trained on the data collected. For network training we chose the FANN_TRAIN_RPROP back propagation training method which was devised by Riedmiller et al. and improved by Igel et al. [6,7]. Training was concluded when the error dropped below 0.002. When 40,000 epochs were completed before the error dropped below 0.002, the training was restarted and the user was shown the markers again. This was not a rare occurrence but always attributable to a failure in detection of the iris.

Error Calculation

The error of the eye tracker was computed in the X and Y directions as a percentage of the relevant screen dimension. For comparison to existing eye gaze trackers, errors in degrees are calculated according to the following equation.

$$Error_{degrees} = arct\left[\frac{\left(Error_{percent} * ScreenDimension_{Cm}\right)}{2 * UserDistanceFromScreen_{cm}}\right] * 360/\pi$$

The distance of the of the subject's eyes from the screen was 75 cm, the screen width was 28 cm and the screen height was 18 cm.

Results

Qualitative Results

Five subjects, consisting of males and females, ages 9 to 66 (mean=34.8, SD= 20.39) participated in the qualitative testing. All but one user had normal vision. One user, a 66 year old male, was far-sighted but could see the markers well enough with his glasses off to participate. Glasses were not permitted in the experiments because they disrupted the eye detection. None of the users had prior experience with eye tracking. The neural network was used to estimate the location of each user's gaze which was drawn as a green marker on the screen. All users indicated that they felt the green marker was "following" their eye movements. Two users reported the green marker being exactly where they looked while keeping their focus within one quadrant of the screen.

All users made the following observations:

- Head movement affected the accuracy of the gaze location considerably
- The marker was "jumpy" even when keeping their eyes still
- Inability to move the green marker to one or more regions of the screen

The sensitivity to head position was expected because of the neural network's dependence on the appearance of the eye in the image and the appearance of the eye changes as the user moves his or her head. Inconsistencies in finding the iris highlight and noise within the webcamera image were the likely cause of the "jumpy" marker in the tests. Inspection of the image of the eye showed a significant amount of noise from the webcamera's sensor. Changes in this noise from frame to frame appeared to be responsible for the "jumpy" behavior of the marker even when detection of the iris highlight was stable.

Quantitative Testing

A preliminary quantitative error assessment was made where one user was shown the same set of points used in training. After network training, the set of 48 markers was shown for the subject to follow with her eyes. At each marker one image of the eye was used as input to the gaze tracker and the error (in both the X and Y directions) was recorded. The average errors for the X and Y directions were 1.38° (±1.01°) and 1.63° (±0.75°). The results show an average accuracy improvement of 40-80% compared to previous research reported by Bäck [4].

Five subjects, consisting of males and females, ages 28 to 66 (mean=43.6, SD=17.9) participated in a second experiment to test the eye tracker's error. Each user was shown 50 points randomly placed around the screen. The average errors for all users in the X and Y directions were 2.60° (±3.43°) and 2.61° (±2.45°), providing the average distance error of 3.68º (±4.24). The large errors compared to those from the tests using training points suggest that the neural network had trouble extrapolating. Adding hidden layers to the neural network would possibly rectify this problem.

Discussion, Conclusion and Future Work

Analysis

Our research produced an eye tracker, created with an unmodified commodity webcam. The eye tracker works with a low enough error (<3.68º) indicated by the accuracy results. The users of the system subjectively confirmed the responsiveness of the eye tracker. There are some areas, listed below, where our methodology could be improved, which could enable our eye tracker to become more accurate and responsive.

Better Eye Location

A large source of error was the inconsistency of the location of the iris highlight resulting in an inconsistent image being sent to the neural network. More constant and useful data could be produced from having the image of the eye's position static with respect to the placement of the eye socket.

Head Position

The largest source of difficulty experienced by the users was the sensitivity of the gaze tracker to head movements. The orientation and distance of the head caused significant errors in the eye tracking. We believe this problem can be reduced or eliminated by taking measurements on the position of the head and using them as inputs to the neural network, similar to the method employed by Bäck [4].

Neural Network Pre-Training

The training step in our gaze tracker required 30-60 seconds to complete. Longer training may be required to increase the accuracy. One way to reduce user training time is to pre-train the neural network using a large number of subjects. The neural network would be primed for the general appearance of a human eye across the various gaze positions. The user would then perform training to adjust the neural network to the user’s eye.

Higher Resolution Camera

The camera used in this research was capable of a resolution of 1280x1024 pixels but the programming interface was limited to 640x480. A higher resolution image would improve several aspects of the gaze tracker's operation. Location of the eye and iris (and possibly pupil) would be improved as would image processing including edge detection of the structures in the eye (i.e. iris, pupil, and sclera). These effects would be beneficial even if the neural network continued to use a subsequently resized image.

Conclusion

We have shown that it is plausible for an unmodified webcamera to be used for eye tracking. If further investigation overcomes the areas we have outlined, a usable eye tracking interface could be implemented which requires no special hardware or setup. Such an interface would be very useful to the disabled and could make existing user interfaces more context aware.

The results presented in our research indicate a 40% to 80% improvement of such a device compared to the previous work. We believe our work makes another step towards making affordable eye-guided computer interfaces for the disabled a reality.

References

[1] FANN. [Online] [Cited: December 2, 2009.] http://leenissen.dk.

[2] OpenCV. [Online] [Cited: December 2, 2009.] http://opencv.willowgarage.com.

[3] Low-Cost Gaze Interaction: Ready to Deliver the Promises. Agustin, Javier San, et al. : ACM, 2009. Proceedings of the 27th international conference extended abstracts on Human factors in computing systems. pp. 4453-4458.

[4] Bäck, David. Neural Network Gaze Tracking using Web Camera. Välkommen till institutionen för medicinsk teknik : Masters Thesis, 2005.

[5] Non-Intrusive Gaze Tracking Using Artificial Neural Networks. Baluja, Shumeet and Pomerleau, Dean. 1994, Technical Report: CS-94-102.

[6] Task-Dependent Evolution of Modularity in Neural Networks. Igel, Christian, Toussaint, Marc and Hüsken, Michael. 2002, Connection Science.

[7] A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. Riedmiller, Martin and Braun, Heinrich. 1993. IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS.

[8] Sibert, L.E. and Jacob, R.J. Evaluation of eye gaze interaction. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '00. ACM, New York, NY, 281-288.

[9] Rapid Object Detection using a Boosted Cascade of Simple Features. Viola, Paul and Jones, Michael. s.l. : IEEE, 2001. Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. pp. I-511- I-518 vol.1.