Machine Vision and Applications (2004) 15: 139–148
Digital Object Identifier (DOI) 10.1007/s00138-004-0139-4

Machine Vision and Applications

Eye and gaze tracking for interactive graphic display

Zhiwei Zhu, Qiang Ji

Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, JEC 6219, Troy, NY 12180-3590, USA

Received: 21 July 2002 / Accepted: 3 February 2004
Published online: 8 June 2004 – © Springer-Verlag 2004

Abstract. This paper describes a computer vision system based on active IR illumination for real-time gaze tracking for interactive graphic display. Unlike most of the existing gaze tracking techniques, which often require assuming a static head to work well and require a cumbersome calibration process for each person, our gaze tracker can perform robust and accurate gaze estimation without calibration and under rather significant head movement. This is made possible by a new gaze calibration procedure that identifies the mapping from pupil parameters to screen coordinates using generalized regression neural networks (GRNNs). With GRNNs, the mapping does not have to be an analytical function and head movement is explicitly accounted for by the gaze mapping function. Furthermore, the mapping function can generalize to other individuals not used in the training. To further improve the gaze estimation accuracy, we employ a hierarchical classification scheme that deals with the classes that tend to be misclassified. This leads to a 10% improvement in classification error. The angular gaze accuracy is about 5° horizontally and 8° vertically. The effectiveness of our gaze tracker is demonstrated by experiments that involve gaze-contingent interactive graphic display.

Keywords: Eye tracking – Gaze estimation – Human–computer interaction – Interactive graphic display – Generalized regression neural networks

1 Introduction

Gaze determines a person's current line of sight or point of fixation. The fixation point is defined as the intersection of the line of sight with the surface of the object being viewed (such as the screen). Gaze may be used to interpret the user's intention for noncommand interactions and to enable (fixation-dependent) accommodation and dynamic depth of focus. The potential benefits of incorporating eye movements into the interaction between humans and computers are numerous. For example, knowing the location of a user's gaze may help a computer to interpret the user's request and possibly enable a computer to ascertain some cognitive states of the user, such as confusion or fatigue.

Eye gaze direction can express the interests of a user; it is a potential porthole into the current cognitive processes. Communication through the direction of the eyes is faster than any other mode of human communication. In addition, real-time monitoring of gaze position permits the introduction of display changes that are contingent on the spatial or temporal characteristics of eye movements. Such methodology is referred to as the gaze-contingent display paradigm. For example, gaze may be used to determine one's fixation on the screen, which can in turn be used to infer the information the user is interested in. Appropriate actions can then be taken such as increasing the resolution or increasing the size of the region where the user fixates. Another example is economizing on bandwidth by putting high-resolution information only where the user is currently looking.

Gaze tracking is therefore important for human–computer interaction (HCI) and intelligent graphics. Numerous techniques have been developed including some commercial eye gaze trackers. Basically, these can be divided into video-based techniques and non-video-based techniques. Usually, non-video-based methods use some special contacting devices attached to the skin or eye to catch the user's gaze. So they are intrusive and interfere with the user. For example, in [7], electrodes are placed on a user's skin around the eye socket to measure changes in the orientation of the potential difference between retina and cornea. This technique is too troublesome to be used for everyday use because it requires the close contact of electrodes to the user. Also in [3], a nonslipping contact lens is attached to the front of a user's eye. Although the direction of gaze can be obtained very accurately in this method, it is so awkward and uncomfortable that it is impossible for nonlaboratory tasks.

Recently, using a noncontacting video camera together with a set of techniques, numerous video-based methods have been presented. Compared with non-video-based gaze tracking methods, video-based gaze tracking methods have the advantage of unobtrusiveness and being comfortable during the process of gaze estimation. We will concentrate on the video-based approaches in this paper.

Correspondence to: Q. Ji (e-mail: qji@ecse.rpi.edu)

The direction of a person's gaze is determined by two factors: face orientation (face pose) and eye orientation (eye gaze). Face pose determines the global direction of the gaze, while eye gaze determines the local direction of the gaze. Global gaze and local gaze together determine the final gaze of the person. According to these two aspects of gaze information, video-based gaze estimation approaches can be divided into a head-based approach, an ocular-based approach, and a combined head- and eye-based approach.

The head-based approach determines a user's gaze based on head orientation. In [16], a set of Gabor filters is applied locally to the image region that includes the face. This results in a feature vector to train a neural network to predict the two neck angles, pan and tilt, providing the desired information about head orientation. Mukesh and Ji [14] introduced a robust method for discriminating 3D face pose (face orientation) from a video sequence featuring views of a human head under variable lighting and facial expression conditions. Wavelet transform is used to decompose the image into multiresolution face images containing both spatial and spatial-frequency information. Principal component analysis (PCA) is used to project a midfrequency, low-resolution subband face pose onto a pose eigenspace where the first three eigencoefficients are found to be most sensitive to pose and follow a trajectory as the pose changes. Any unknown pose of a query image can then be estimated by finding the Euclidean distance of the first three eigencoefficients of the query image from the estimated trajectory. An accuracy of 84% was obtained for test images unseen during training under different environmental conditions and facial expressions, and even for different human subjects. Gee et al. [6] estimated the user's gaze direction by head orientation from a single, monocular view of a face by ignoring the eyeball's rotation. Our recent efforts [11] in this area led to the development a technique that classifies 3D face poses based on some ocular parameters. Gaze estimation by head orientation, however, only provides a global gaze since one's gaze can still vary considerably given the head orientation. By looking solely at the head movements, the accuracy of the user's gaze is traded for flexibility.

The ocular-based approach estimates gaze by establishing the relationship between gaze and the geometric properties of the iris or pupil. One of the problems in the ocular-based approach is that only local information, i.e., the images of the eyes, is used for estimating the user's gaze. Consequently, the system relies on a relatively stable position of the user's head with respect to the camera, and the user should not rotate his head. Iris and pupil, two prominent and reliable features within the eye region, are often utilized in the gaze determination approach. The special character of the iris structure, namely, the transition from white to dark then dark to white, makes it possible to segment iris from the eye region reliably. The special bright pupil effect under IR illumination makes pupil segmentation very robust and effective. Specifically, the iris-based gaze estimation approach computes gaze by determining the iris location from the iris' shape distortions, while the pupil-based approach determines gaze based on the relative spatial positions between pupil and the glint (cornea reflection). For example, neural networks have been used in the past for this task [2,20]. The idea is to extract a small window containing the eye and feed it, after proper intensity normalization, to a neural network. The output of the network determines the coordinates of the gaze. A large training set of eye images needs to be collected for training, and the accuracy of it is not as good as for other techniques. Zhu et al. [21] proposed an eye gaze estimation method based on the vector from the eye corner to the iris center. First, one inner eye corner and the iris center are extracted from the eye image. Then a 2D linear mapping function from the vector between the eye corner and iris center to the gaze point in the screen is obtained by a simple calibration. But this simple linear mapping is only valid for a static head position. When the face moves, it will no longer work. Wang et al. [19] presented a new approach to measuring human eye gaze via iris images. First, the edges of the iris are located and the iris contour is fitted to an ellipse. The eye gaze, defined in their paper as the unit surface normal to the supporting plane of the iris, can be estimated from the projection of the iris contour ellipses. But in their method, calibration is needed to obtain the radius of the iris contour for different people. Also, a high-quality eye image is needed to obtain the iris contour, and the user should keep the eye fully open to avoid eyelid occlusion of the iris.

So far, the most common approach to ocular-based gaze estimation is based on the relative position between pupil and the glint (cornea reflection) on the cornea of the eye [4,8, 9,15,12,13,5,1]. Assuming a static head, methods based on this idea use the glint as a reference point; thus the vector from the glint to the center of the pupil is used to infer the gaze direction, assuming the existence of a simple analytical function that maps glint vector to gaze. While contact free and nonintrusive, these methods work well only for a static head, which is a rather restrictive constraint on the part of the user. Even a chin rest [1] is used to keep the head still because minor head movement can foil these techniques. This poses a significant hurdle to natural human–computer interaction (HCI). Another serious problem with the existing eye and gaze tracking systems is the need to perform a rather cumbersome calibration process for each individual. For example, in [13], nine points are arranged in a 3×3 grid on a screen, and the user is asked to fixate his/her gaze on a certain target point one by one. On each fixation, the pupil-glint vector and the corresponding screen coordinate are obtained, and a simple second-order polynomial transformation is used to obtain the mapping relationship between the pupil-glint vector and the screen coordinates. In their system, only slight head motion is allowed, and recalibration is needed whenever the head is moved or a new user wants to use it.

The latest research efforts are aimed at overcoming this limitation. Researchers from NTT in Japan proposed [15] a new gaze tracking technique based on modeling the eyeball. Their technique significantly simplifies the gaze calibration procedure, requiring only two points to perform the necessary calibration. The method, however, still requires a relatively stationary head, and it is difficult to acquire an accurate geometric eyeball model for each subject. Researchers from IBM [12] are also studying the feasibility of completely eliminating the need for a gaze calibration procedure by using two cameras and by exploiting the geometry of eyes and their images. Also, Shih et al. [17] proposed a novel method for estimating gaze direction based on 3D computer vision techniques. Multiple cameras and multiple point light sources are utilized in their method. Computer simulation shows promising results, but it seems too complicated for practical use. Other recent efforts

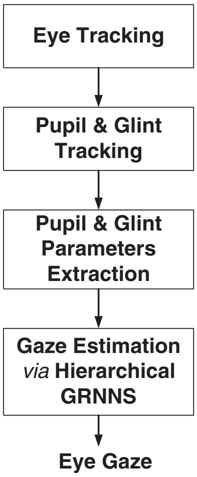

Eye Tracking
Pupil & Glint Tracking
Pupil & Glint Parameters Extraction
Gaze Estimation *via* Hierarchical GRNNS
Eye Gaze

Fig. 1. Major components of the proposed system

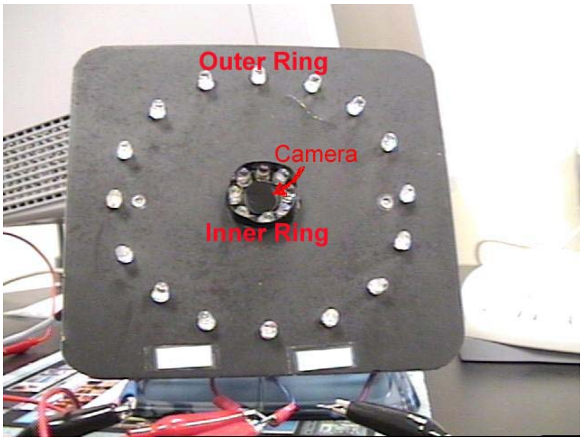


Fig. 2. Hardware setup: the camera with an active IR illuminator

[22,5] also focus on improving eye tracking robustness under various lighting conditions.

In view of these limitations, in this paper we present a gaze estimation approach that accounts for both the local gaze computed from the ocular parameters and the global gaze computed from the head pose. The global gaze (face pose) and local gaze (eye gaze) are combined together to obtain the precise gaze information of the user. A general approach that combines head pose information with eye gaze information to perform gaze estimation is proposed. Our approach allows natural head movement while still estimating gaze accurately. Another effort is to make the gaze estimation calibration free. New or existing users who have moved need not undergo a personal gaze calibration before using the gaze tracker. Therefore, compared with the existing gaze tracking methods, our method, though at a lower angular gaze resolution (about 5° horizontally and 8° vertically), can perform robustly and accurately without calibration and with natural head movements.

An overview of our algorithm is given in Fig. 1. In Sect. 2, we will briefly discuss our system setup and the eye detection and tracking method. In Sect. 3, the technology for pupil and glint detection and tracking is discussed. Also, the parameters extracted from the detected pupil and glint for gaze calibration are covered. In Sect. 4, gaze calibration using GRNNs is discussed. Section 5 discusses the experimental results and the operational volumes for our gaze tracker. The paper ends in Sect. 6 with a summary and a discussion of future work.

2 Eye tracking

Gaze tracking starts with eye tracking. For eye tracking, we track pupils instead. We use IR LEDs that operate at a power of 32 mW in a wavelength band 40 nm wide at a nominal wavelength of 880 nm. As in [10], we obtain a dark and a bright pupil image by illuminating the eyes with IR LEDs located off (outer IR ring) and on the optical axis (the inner IR ring), respectively. To further improve the quality of the image and to minimize interference from light sources other than the IR illuminator, we use an optical bandpass filter that has a wavelength pass band only 10 nm wide. The filter has increased the signal-to-noise ratio significantly compared with the case without using the filter. Figure 2 illustrates the IR illuminator consisting of two concentric IR rings and the bandpass filter.

Figure 3 summarizes our pupil detection and tracking algorithm, which starts with pupil detection in the initial frames, followed by tracking. Pupil detection is accomplished based on both the intensity of the pupils (the bright and dark pupils as shown in Fig. 4) and on the appearance of the eyes using the support vector machine (SVM). The use of the SVM avoids falsely identifying a bright region as a pupil. Specifically, candidates of pupils are first detected from the difference image resulting from subtracting the dark pupil image from the bright pupil image. The pupil candidates are then validated using the SVM to remove spurious pupil candidates. Given the detected pupils, pupils in the subsequent frames can be detected efficiently via tracking. Kalman filtering is used since it allows pupil positions in the previous frame to predict pupil positions in the current frame, thereby greatly limiting the search space. Kalman filtering tracking based on pupil intensity is therefore implemented. To avoid Kalman filtering going awry due to the use of intensity only, Kalman filtering is augmented by mean-shift tracking, which tracks an object based on its intensity

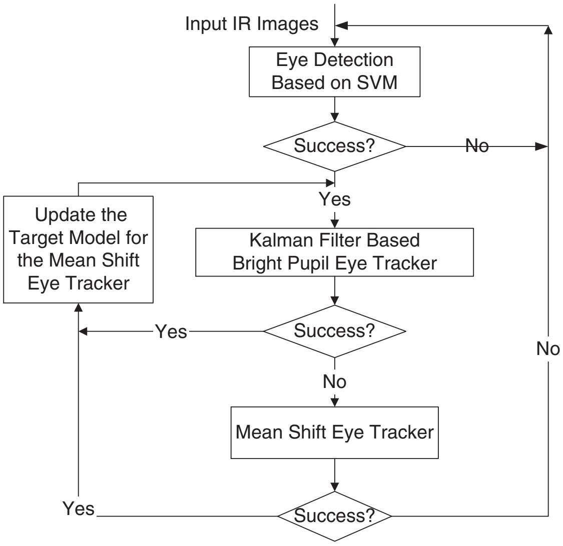

Input IR Images
Eye Detection Based on SVM
Success?
No
Yes
Update the Target Model for the Mean Shift Eye Tracker
Kalman Filter Based Bright Pupil Eye Tracker
Yes
Success?
No
No
Mean Shift Eye Tracker
Yes
Success?

Fig. 3. Flowchart of our pupil detection and tracking algorithm

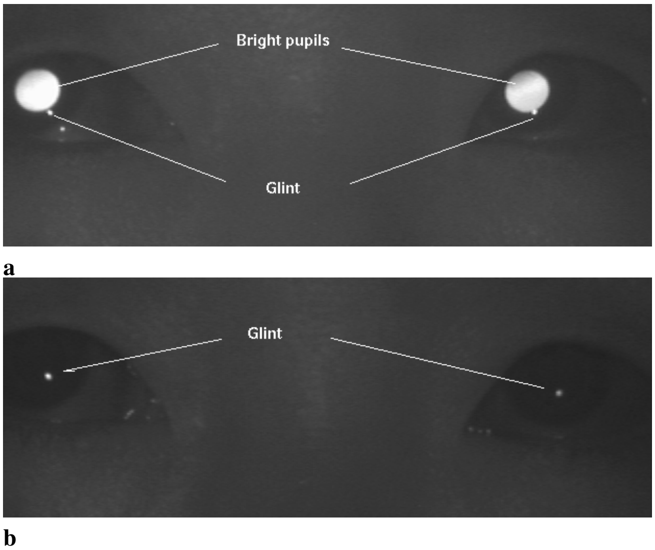


Fig. 4. Bright (**a**) and dark (**b**) pupil images with glints

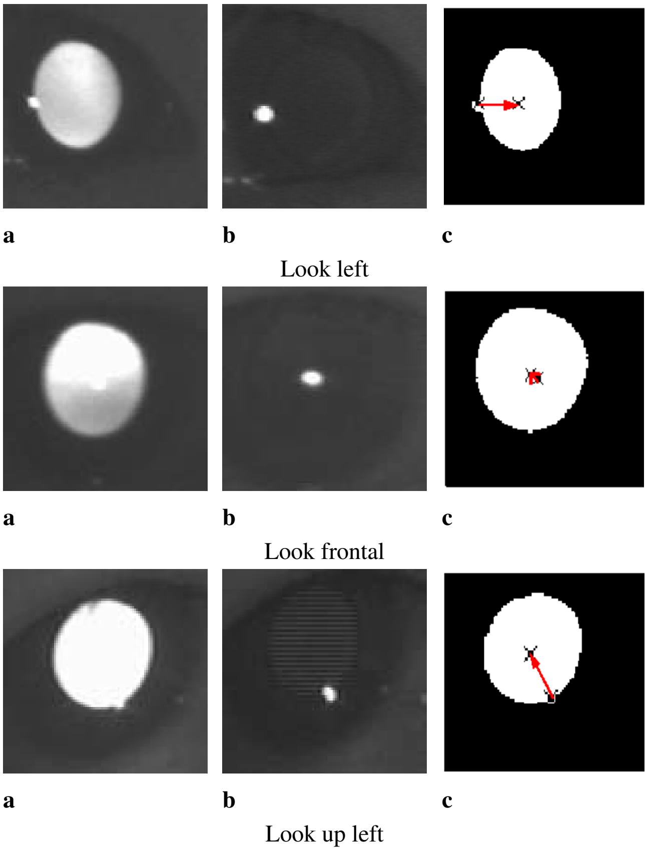


Fig. 5. Relative spatial relationship between glint and bright pupil center used to determine eye gaze position. **a** Bright pupil images. **b** Glint images. **c** Pupil–glint relationship generated by superimposing glint to the thresholded bright pupil images

distribution. Details on our eye detection and tracking may be found in [22].

3 Gaze determination and tracking

Our gaze estimation algorithm consists of three parts: pupil-glint detection and tracking, gaze calibration, and gaze mapping. In the following discussion, each part will be discussed in detail.

3.1 Pupil and glint detection and tracking

Gaze estimation starts with pupil and glint detection and tracking. For gaze estimation, we continue using the IR illuminator as shown in Fig. 2. To produce the desired pupil effects, the two rings are turned on and off alternately via the video decoder we developed to produce the so-called bright and dark pupil effect as shown in Fig. 4a and b.

Note that glint (the small brightest spot) appears on both images. Given a bright pupil image, the pupil detection and tracking technique described in Sect. 2 can be directly applied to pupil detection and tracking. The location of a pupil at each frame is characterized by its centroid. Algorithm-wise, glint is detected from the dark image since the glint is much brighter than the rest of the eye image, which makes glint detection and tracking much easier. The pupil detection and tracking technique can be used to detect and track glint from the dark images. Figure 5c shows the detected glints and pupils.

3.2 Local gaze calibration

Given the detected glint and pupil, a mapping function is often used to map the pupil–glint vector to gaze (screen coordinates). Figure 5 shows the relationship between gaze and the relative position between the glint and the pupil.

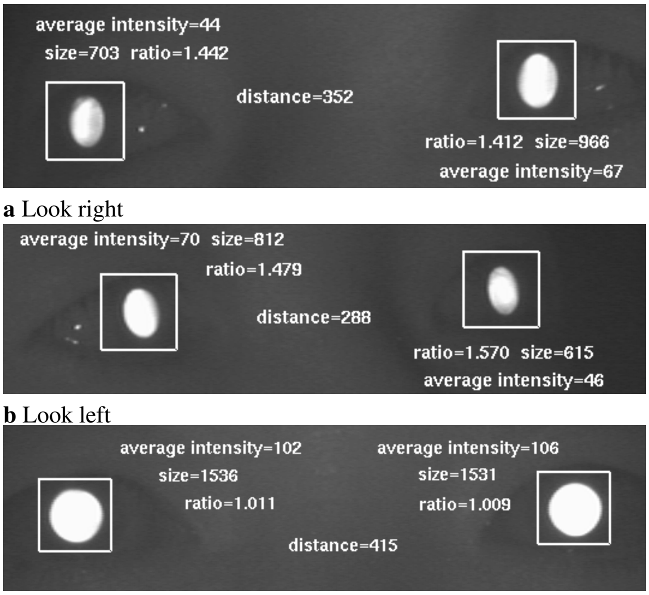
The mapping function is often determined via a calibration procedure. The calibration process determines the parameters for the mapping function given a set of pupil–glint vectors and the corresponding screen coordinates (gazes). The conventional approach to gaze calibration suffers from two shortcomings. First, most of the mapping is assumed to be an analytical function of either linear or second-order polynomial, which may not be reasonable due to perspective projection and the spherical surface of the eye. Second, only coordinate displacements between the pupil center and glint position are used for gaze estimation, which makes the calibration face orientation dependent. Another calibration is needed if the head has moved since the last calibration, even for minor head movement. In practice, it is difficult to keep the head still, and the existing gaze tracking methods will produce an incorrect result if the head moves, even slightly.

Therefore, head movement must be incorporated into the gaze estimation procedure.

3.3 Face pose by pupil properties

In our pupil tracking experiments, we had an interesting observation that the pupil appearances vary with different poses. Figure 6 shows the appearance changes of pupil images under different face orientations.

Our study shows that there exists a direct correlation between 3D face pose and properties such as pupil size, inter-



a Look right

b Look left

c Look front

Fig. 6. Changes of pupil images under different face orientations

pupil distance, pupil shape, and pupil ellipse orientation. It is apparent from these images that:

1. The interpupil distance decreases as the face rotates away from the frontal orientation.
2. The ratio between the average intensity of the two pupils either increases to over one or decreases to less than one as the face rotates away.
3. The shapes of the two pupils become more elliptical as the face rotates away or rotates up/down.
4. The sizes of the pupils also decrease as the face rotates away or rotates up/down.
5. The orientation of the pupil ellipse will change as the face rotates around the camera optical axis.

Based on the above observations, we can develop a face pose classification algorithm by exploiting the relationships between face orientation and these pupil parameters. We build a so-called pupil feature space (PFS) that is constructed by nine pupil features: interpupil distance, sizes of left and right pupils, intensities of left and right pupils, ellipse angles of left and right pupils, and ellipse ratios of left and right pupils. To make those features scale invariant, we further normalize those parameters by dividing by corresponding values of the front view. Figure 7 shows sample data projections in 3D PFS, from which we see clearly that there are five distinctive clusters corresponding to five face orientations (five yaw angles). Note that, although we can only plot 3D space here, PFS is constructed by nine features in which the clusters will be more distinctive. So a pose can be determined by the projection of pupil properties in PFS. Details on the face pose estimation based on pupil parameters may be found in [11].

3.4 Parameters for gaze calibration

PFS can capture relationships between 3D face pose and the geometric properties of the pupils, which proves that there exists a direct correlation between 3D face pose and the geometric properties of the pupils.

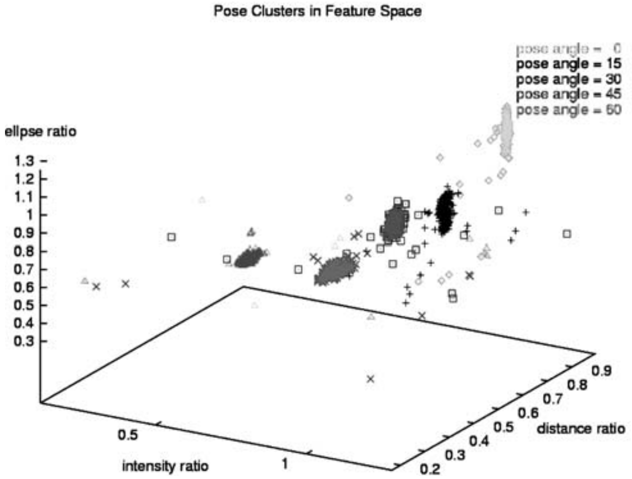


Fig. 7. Face pose clusters in pupil feature space

To incorporate the face pose information into the gaze tracker, the factors accounting for the head movements and those affecting the local gaze should be combined to produce the final gaze. Hence, six factors are chosen for the gaze calibration to get the mapping function: Δx, Δy, r, θ, g_x, and g_y. Δx and Δy are the pupil–glint displacement. r is the ratio of the major to minor axes of the ellipse that fits the pupil. θ is the pupil ellipse orientation, and g_x and g_y are the glint image coordinates. The choice of these factors is based on the following rationale. Δx and Δy account for the relative movement between the glint and the pupil, representing the local gaze. The magnitude of the pupil–glint vector can also relate to the distance of the subject to the camera. r is used to account for out-of-plane face rotation. The ratio should be close to 1 when the face is frontal. The ratio becomes larger or less than 1 when the face turns either up/down or left/right. Angle θ is used to account for in-plane face rotation around the camera optical axis. Finally, (g_x, g_y) is used to account for the in-plane head translation.

The use of these parameters accounts for both head and pupil movement since their movements will introduce corresponding changes to these parameters. This effectively reduces the head movement influence. Furthermore, the input parameters are chosen such that they remain relatively constant for different people. For example, these parameters are independent of the size of the pupils, which often vary among people. The determined gaze mapping function, therefore, will be able to generalize. This effectively eliminates the need to recalibrate for another person.

4 Gaze calibration via generalized regression neural networks (GRNNs)

Given the six parameters affecting gaze, we now need to determine the mapping function that maps the parameters to the actual gaze. In this study, one's gaze is quantized into eight regions on the screen (4×2), as shown in Fig. 8.

The reason for using neural networks to determine the mapping function is because of the difficulty in analytically

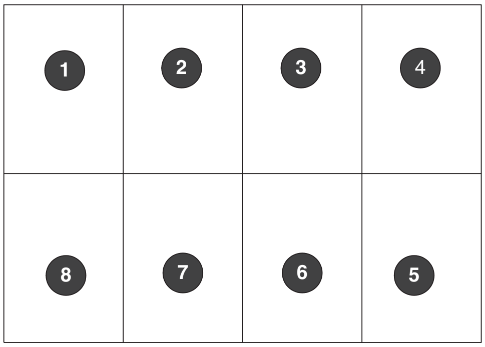


Fig. 8. Quantized eye gaze regions on a computer screen

deriving the mapping function that relates pupil and glint parameters to gaze under different face poses and for different persons. Given sufficient pupil and glint parameters, we believe there exists a unique function that relates gaze to different pupil and glint parameters.

Introduced in 1991 by Specht [18] as a generalization of both radial basis function networks (RBFNs) and probabilistic neural networks (PNNs), GRNNs have been successfully used in function approximation applications. GRNNs are memory-based feedforward networks based on the estimation of probability density functions. GRNNs feature fast training times, can model nonlinear functions, and have been shown to perform well in noisy environments given enough data. Our experiments with different types of neural networks also reveal superior performance of GRNN over the conventional feedforward backpropagation neural network.

The GRNN topology consists of four layers: input layer, hidden layer, summation layer, and output layer. The input layer has six inputs, representing the six parameters, while the output layer has one node. The number of hidden nodes is equal to the number of training samples, with one hidden node added for each set of the training sample. The number of nodes in the summation layer is equal to the number of output nodes plus 1. Figure 9 shows the GRNN architecture we use.

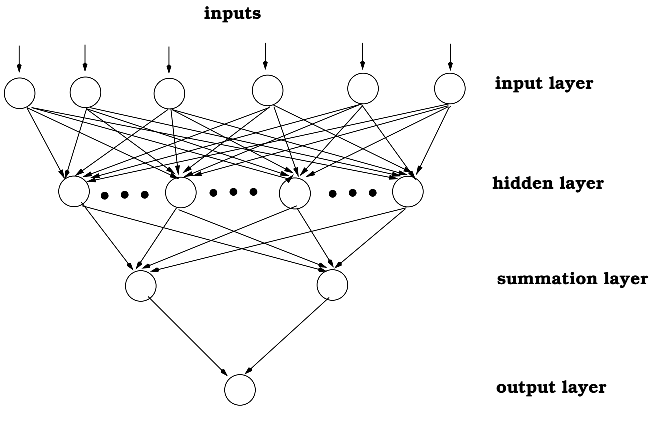


Fig. 9. GRNN architecture used for gaze calibration

Due to a significant difference in horizontal and vertical spatial gaze resolution, two identical GRNNs were constructed, with the output node representing the horizontal and vertical gaze coordinates s_x and s_y, respectively.

The parameters to use for the input layer must vary with different face distances and orientations to the camera. Specifically, the input vector to the GRNN is

$$\mathbf{g} = \left[\varDelta x \ \varDelta y \ r \ \theta \ g_x \ g_y \right] .$$

Before supplying to the GRNN, the input vector is normalized appropriately. The normalization ensures that all input features are in the same range.

A large amount of training data under different head positions is collected to train the GRNN. During the training data acquisition, the user is asked to fixate his/her gaze on each gaze region. On each fixation, ten sets of input parameters are collected so that outliers can be identified subsequently. Furthermore, to collect representative data, we use one subject from each race including an Asian subject and a Caucasian subject. In the future, we will extend the training to additional races. The subjects' ages range from 25 to 65. The acquired training data, after appropriate preprocessing (e.g., nonlinear filtering to remove outliers) and normalization, are then used to train the NN to obtain the weights of the GRNN. GRNNs are trained using a one-pass learning algorithm and are therefore very fast.

4.1 Gaze mapping and classification

After training, given an input vector the GRNN can then classify it into one of the eight screen regions. Figure 10 shows that there are distinctive clusters of different gazes in the three-parameter space. In this figure, we only plot 3D space. The clusters would be more distinctive if they were plotted by six features.

Although the clusters of different gazes in the gaze parameters are distinctive, the clusters sometimes overlap. This is especially a problem for gaze regions that are spatially adjacent to each other. Our experiment shows it is not always possible to map an input vector to a correct gaze class. Gaze misclassifications may occur. Our experiments confirmed this as shown by the confusion matrix shown in Table 1. An average of gaze classification accuracy of (85% accuracy) was

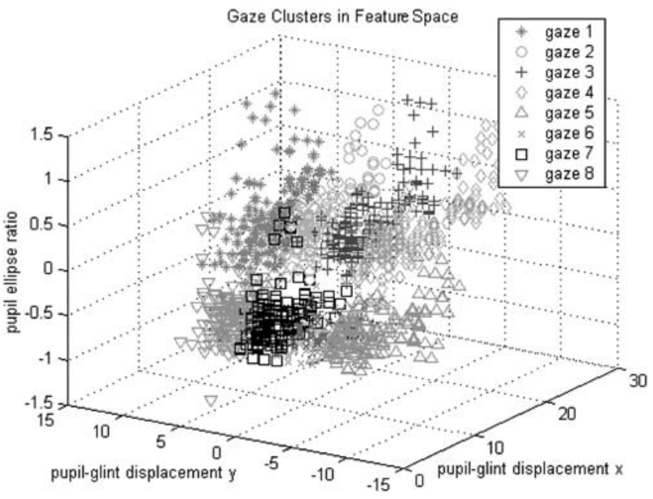


Fig. 10. Gaze clusters in feature space

Table 1. Gaze classification results for the one-level GRNN classifier. An average of gaze classification accuracy of 85% was achieved for 480 testing data not included in the training data for the one-level gaze classifier

Ground truth	Estimated result (mapping target #)								Correct-ness
(target #)	1	2	3	4	5	6	7	8	rate (%)
1	49	11	0	0	0	0	0	0	82
2	0	52	8	0	0	0	0	0	87
3	0	0	46	14	0	0	0	0	77
4	0	0	0	59	1	0	0	0	98
5	0	0	0	0	60	0	0	0	100
6	0	0	0	6	8	46	0	0	77
7	0	0	2	0	0	5	53	0	88
8	4	0	0	0	0	0	6	50	84

achieved for 480 testing data not included in the training data. Further analysis of this result shows significant misclassification occur between neighboring gaze regions. For example, about 18% of the gaze in region 1 are misclassified to gaze region 2 while about 24% gazes for region 3 are misclassified as gaze region 4. We therefore conclude misclassification almost exclusively occur among neighboring gaze regions.

4.2 Hierarchical gaze classifier

To reduce misclassification among neighboring gaze classes, we design a hierarchical classifier to perform additional classification. The idea is to focus on the gaze regions that tend to get misclassified and perform reclassification for these regions. Therefore, we design a classifier for each gaze region to perform the neighboring classification again. According to the regions defined in Fig. 8, we first identify the neighbors for each gaze region and then only use the training data for the gaze region and its neighbors to train the classifier. Specifically, each gaze region and its neighbors are identified as follows:

1. Region 1: neighbors: 2,8
2. Region 2: neighbors: 1,3,7
3. Region 3: neighbors: 2,4,6
4. Region 4: neighbors: 3,5
5. Region 5: neighbors: 4,6
6. Region 6: neighbors: 3,5,7
7. Region 7: neighbors: 2,6,8
8. Region 8: neighbors: 1,7

These subclassifiers are then trained using the training data consisting of the neighbors' regions only. The subclassifiers are subsequently combined with the whole classifier to construct a hierarchical gaze classifier as shown in Fig. 11.

Given an input vector, the hierarchical gaze classifier works as follows. First, the whole classifier classifies the input vector into one of the eight gaze regions; then, according to the classified region, the corresponding subclassifier is activated to reclassify the input vector to the gaze regions covered by the subclassifier. The result obtained from the subclassifier will be considered as the final classified result. Our expectation is that the final classification results will improve or remain the same, or at least will not get worse. Our experiments prove this.

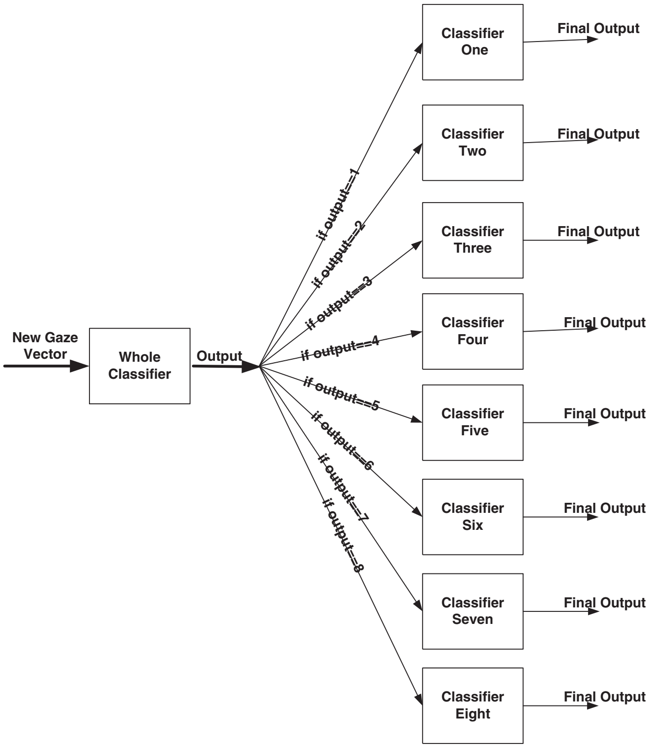


Fig. 11. Structure of hierarchical gaze classifier

5 Experimental results and analysis

To validate the performance of our gaze tracker, we perform a series of experiments that involves the use of gaze to interactively determine what to display on the screen.

The first experiment involves visual evaluation of our eye tracking system. A laser pointer is used to point at the different regions of the computer screen. As expected, the user gaze is able to accurately follow the movement of the laser pointer that moves randomly from one gaze region to another gaze region, even under natural head movement. A video demo of this experiment is available at `http://www.ecse.rpi.edu/~cvrl/Demo/demo.html`.

To quantitatively characterize the accuracy of our system, the second experiment studies the performance of our system under different face orientations and distances to the cameras and with different subjects. Table 2 summarizes the classification results. Compared with Table 1, which was produced based on the same data, we can see that the hierarchical gaze classifier can achieve an average of around 95% accuracy for a different subject, which improves the accuracy by around 10% over the existing one-level gaze classifier method. Specifically, the misclassification rate between neighbors 1 and 2 has decreased from 18% to about 8%, while the misclassification rate between gaze regions 3 and 4 has decreased to about 5% from the previous 24%. The classification errors for other gaze

Table 2. An average of gaze classification results (95% accuracy) was achieved for 480 testing data not included in the training data for the hierarchical gaze classifier

Ground truth (target #)	Estimated result (mapping target #)								Correctness rate (%)
	1	2	3	4	5	6	7	8	
1	55	5	0	0	0	0	0	0	92
2	0	58	2	0	0	0	0	0	97
3	0	0	57	3	0	0	0	0	95
4	0	0	0	59	1	0	0	0	98
5	0	0	0	0	60	0	0	0	100
6	0	0	1	5	5	49	0	0	82
7	0	0	2	0	0	5	53	0	88
8	3	0	0	0	0	0	2	55	92

regions have also improved or remained unchanged. The hierarchical classification therefore meets our expectation.

Our study, however, shows that the system has some difficulty with older people, especially for those who suffer from some vision problem such as farsightedness or nearsightedness.

Our experiments show that our system, working in near real-time (20 Hz) with an image resolution of 640 × 480 pixels on a Pentium III, allows about 15 cm. left/right and up/down head translational movement and allows ±20° left/right head rotation as well as ±15° up/down rotation. The distance to the camera ranges from 1 to 1.5 m. The spatial gaze resolution is about 5° horizontally and 8° vertically, which corresponds to about 10 cm. horizontally and 13 cm. vertically at a distance of about 1.25 m. from the screen.

Finally, we apply our gaze tracker to control graphic display on the screen interactively. This experiment involves user gazes at a region of the computer screen and then blinks three times, and the region being gazed at is then magnified to fill the screen. This repeats until the user can obtain enough details for the region of interest. One application may be a gaze-controlled map display as shown in Figs. 12, 13, and 14, which show a gaze-controlled map display at different levels of detail. We also applied our gaze tracking software to a gaze-controlled car-sale kiosk, where the user of the kiosk remotely (about 1.25 m. from the screen) controls what types of cars and how much detail to display on the screen using his/her gaze. Figures 15, 16, and 17 show sample images of the gaze-controlled kiosk. For a real-time demonstration of the gaze tracking software, please refer to `http://www.ecse.rpi.edu/~cvrl/Demo/demo.html`.

During our study we found that the vertical pupil movement range is much smaller than that of the horizontal range, causing the vertical pupil–glint vector measurement to be much more susceptible to external perturbation such as head movement. This leads to much lower SNR for the vertical data than that of the horizontal data, thereby leading to lower gaze vertical resolution. This explains why we used two separate neural networks for the vertical and horizontal gaze estima-

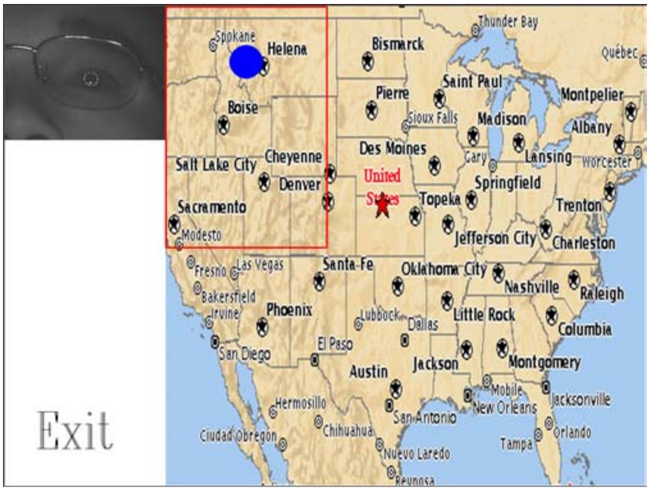


Fig. 12. Map of the United States, with the gaze-selected region as marked by the *shaded circle* and the associated *rectangle* around it

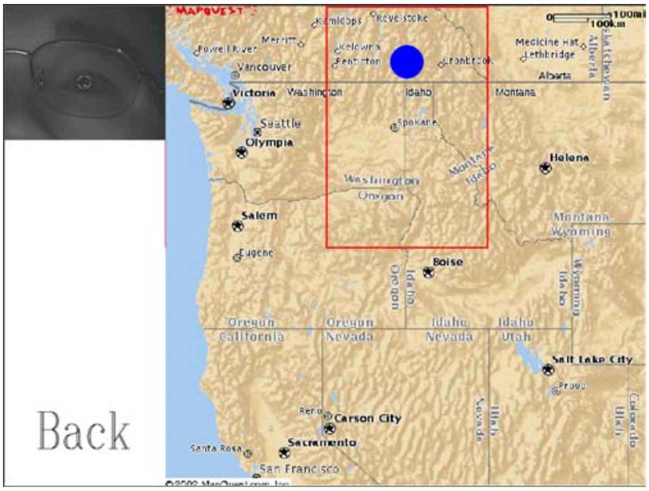


Fig. 13. Blown-up area for selected region in Fig. 12. Another selection is made by gazing on this image as indicated by the *shaded circle* and the associated *rectangle* around it

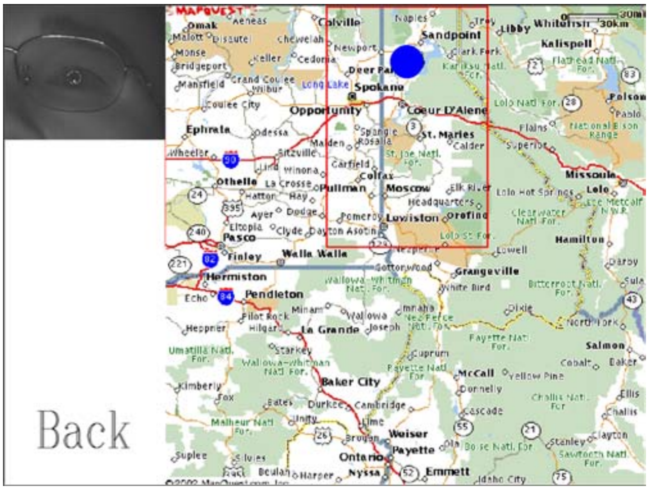


Fig. 14. Blown-up area for selected region in Fig. 13. Another selection is made by gazing on this image as indicated by the *shaded circle* and the associated *rectangle* around it

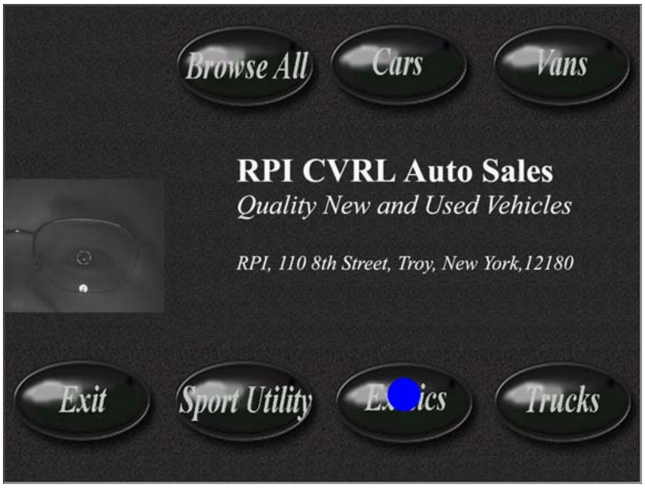
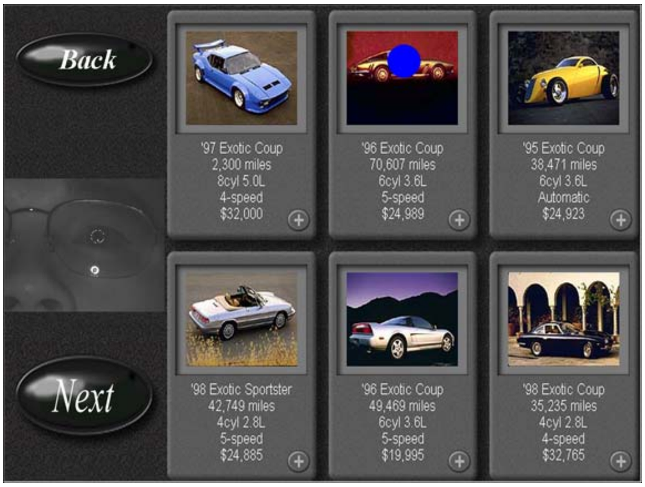

Browse All
Cars
Vans
RPI CVRL Auto Sales
Quality New and Used Vehicles
RPI, 110 8th Street, Troy, New York,12180
Exit
Sport Utility
Exotics
Trucks

Fig. 15. Main menu of gaze-controlled car-sale kiosk, with gaze-selected car option as marked by the *shaded circle* to indicate the "exotic" cars that the user wants to check



'97 Exotic Coup
2,300 miles
8cyl 5.0L
4-speed
$32,000

'96 Exotic Coup
70,607 miles
6cyl 3.6L
5-speed
$24,989

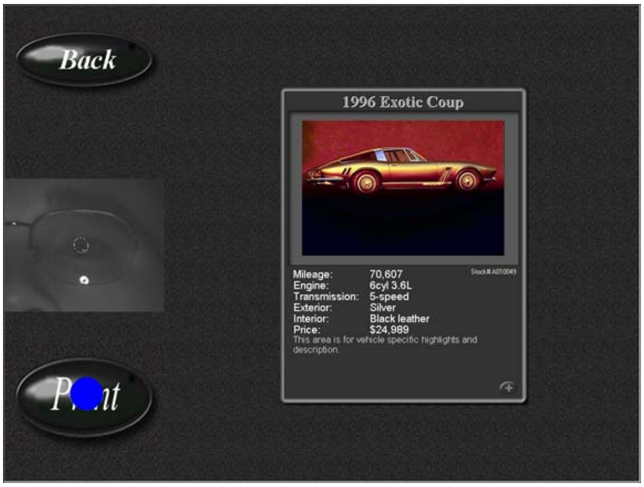
'95 Exotic Coup
38,471 miles
6cyl 3.6L
Automatic
$24,923

'98 Exotic Sportster
42,749 miles
4cyl 2.8L
5-speed
$24,885

'96 Exotic Coup
49,469 miles
6cyl 3.6L
5-speed
$19,995

'98 Exotic Coup
35,235 miles
4cyl 2.8L
4-speed
$32,765

Fig. 16. Cars for the selected car option in Fig. 15. Another selection is made by gazing on this image as indicated by the *shaded circle* to indicate the specific car that the user has an interest in. The user can view other "exotic" cars' information by looking at the "Next" button or browse other types of cars by looking at the "Back" button



1996 Exotic Coup

Stock# A010043

Mileage: 70,607
Engine: 6cyl 3.6L
Transmission: 5-speed
Exterior: Silver
Interior: Black leather
Price: $24,989
This area is for vehicle specific highlights and description.

Fig. 17. Detailed information for the selected car in Fig. 16. The user can print the car information by looking at the "Print" button as indicated by the *shaded circle* or browse other cars by looking at the "Back" button

tion. The current 4×2 gaze regions can be further refined to 4×3 or even 5×4. But this will lead to a decrease in tracking accuracy. This problem, however, can be overcome if we increase the image resolution.

6 Conclusions

In this paper, we present a new approach for gaze tracking. Compared with the existing gaze tracking methods, our method, though at a lower spatial gaze resolution (about $5°$), has the following benefits: no calibration is necessary, it allows natural head movement, and it is completely nonintrusive and unobtrusive while still producing relatively robust and accurate gaze tracking. The improvement is a result of using a new gaze calibration procedure based on GRNNs. With GRNNs, we do not need to assume an analytical gaze mapping function; therefore, we can account for head movement in the mapping. The use of hierarchical classification schemes further improves the gaze classification accuracy.

While our gaze tracker may not be as accurate as some commercial gaze trackers, it achieves sufficient accuracy even under large head movements and, more importantly, is calibration free. It has significantly relaxed the constraints imposed by most existing commercial eye trackers. We believe that, after further improvement, our system will find many applications including smart graphics, human computer interaction, nonverbal communication via gaze, and assistance for people with disabilities.

Acknowledgements. The research described in this report is supported by a grant from AFOSR.

References

1. (ASL) TASL () The applied system laboratory e210 eye tracking systems. http://www.a-s-l.com
2. Baluja S, Pomerleau D (1994) Non-intrusive gaze tracking using artificial neural networks. Technical Report CMU-CS-94-102, Carnegie Mellon University, Pittsburgh
3. Bour L (1997) Dmi-search scleral coil. Technical Report H2-214, Department of Neurology, Clinical Neurophysiology, Academic Medical Center, AZUA, Meibergdreef 9, 115AZ, Amsterdam
4. Ebisawa Y (1995) Unconstrained pupil detection technique using two light sources and the image difference method. In: Visualization and intelligent design in engineering and architecture II. Computational Mechanics Publications, Boston, pp 79–89
5. Ebisawa Y (1998) Improved video-based eye-gaze detection method. IEEE Trans Instrum Meas 47(2):948–955
6. Gee A, Cipoll R (1994) Determining the gaze of faces in images. Image Vision Comput 30:639–647
7. Gips J, Olivieri P, Tecce J (1993) Direct control of the computer through electrodes placed around the eyes. Human-computer interaction: applications and case studies. Elsevier, Amsterdam, pp 630–635

8. Hutchinson TE (1988) Eye movement detection with improved calibration and speed. United States patent [19] no. 4,950,069
9. Hutchinson TE, White K, Worthy JR, Martin N, Kelly C, Lisa R, Frey A (1989) Human-computer interaction using eye-gaze input. IEEE Trans Sys Man Cybern 19(6):1527–1533
10. Ji Q, Yang X (2001) Real time visual cues extraction for monitoring driver vigilance. In: Proceedings of the 2nd international workshop on computer vision systems (ICVS 2001), Vancouver, Canada
11. Ji Q, Yang X (2002) Real time 3D face pose discrimination based on active IR illumination. In: Proceedings of the international conference on pattern recognition
12. Koons D, Flickner M. IBM Blue Eyes project. http://www.almaden.ibm.com/cs/blueeyes
13. Morimoto CH, Koons D, Amir A, Flickner M (1999) Frame-rate pupil detector and gaze tracker. In: Proceedings of the IEEE ICCV'99 frame-rate workshop
14. Motwani MC, Ji Q (2001) 3D face pose discrimination using wavelets. In: Proceedings of the IEEE international conference on image processing (ICIP'2001), Thessaloniki, Greece, 7–10 October 2001
15. Ohno T, Mukawa N, Yoshikawa A (2002) Freegaze: a gaze tracking system for everyday gaze interaction. In: Proceedings of the symposium on eye tracking research and applications, 25–27 March 2002, New Orleans,
16. Rae R, Ritter H (1998) Recognition of human head orientation based on artificial neural networks. IEEE Trans Neural Netw 9(2):257–265
17. Shih S, Wu Y, Liu J (2000) A calibration-free gaze tracking technique. In: Proceedings of the 15th international conference on pattern recognition, Barcelona, Spain
18. Specht DF (1991) A general regression neural network. IEEE Trans Neural Netw 2:568–576
19. Wang J, Sung E (2001) Gaze determination via images of irises. Image Vision Comput 19(12):891–911
20. Yang G, Waibel A (1996) A real-time face tracker. In: Proceedings of the workshop on applications of computer vision, pp 142–147
21. Zhu J, Yang J (2002) Subpixel eye gaze tracking. In: Proceedings of the 5th IEEE international conference on automatic face and gesture recognition
22. Zhu Z, Fujimura K, Ji Q (2002) Real-time eye detection and tracking under various light conditions. In: Proceedings of the symposium on eye tracking research and applications, 25–27 March 2002, New Orleans

Zhiwei Zhu received his M.S. from the Department of Computer Science, University of Nevada at Reno, in August 2002. Currently, he is a Ph.D. candidate in the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY. His research interests include computer vision, pattern recognition, and human–computer interaction.

Dr. Ji received his Ph.D. in electrical engineering from the University of Washington in 1998. He is currently an assistant professor in the Department of Electrical, Computer, and Systems engineering at Rensselaer Polytechnic Institute in Troy, NY. His areas of research include computer vision, probabilistic reasoning for decision making and information fusion, pattern recognition, and robotics. Dr. Ji has published more than 60 papers in refereed journals and conferences. His research has been funded by NSF, NIH, AFOSR, ONR, DARPA, and ARO and Honda. His latest research focuses on applying computer vision and probabilistic reasoning theories to human computer interaction including human fatigue monitoring, user affect modeling and recognition, and active user assistance.