Nonlinear Eye Gaze Mapping Function Estimation via Support Vector Regression

Zhiwei Zhu
Sarnoff Corporation
zzhu@sarnoff.com

Qiang Ji
Department of ECSE, RPI
qji@ecse.rpi.edu

Kristin P. Bennett
Department of Math Sci., RPI
bennek@rpi.edu

Abstract

We propose a novel method for tracking eye gaze that allows natural head movement. Most existing remote eye gaze trackers cannot work under natural head movement due to the difficulty of building a gaze mapping function that can incorporate head motion information. Therefore, the user is required to hold his/her head unnaturally still, possibly with the use of chin-rest. In addition, before each usage of the tracking system, a cumbersome calibration procedure must be performed to obtain a gaze mapping function. Our proposed method significantly improves the conventional Pupil Center Corneal Refection (PCCR) technique to permit natural head movement and to minimize calibration. Support vector regression (SVR) is used to construct a highly nonlinear generalized gaze mapping function that accounts for head movement. As the head moves naturally in front of the camera, the associated gaze mapping function with each new head position will be obtained automatically by the learned generalized gaze mapping function. Once learned, the generalized gaze mapping function can be used by other users via a simple personal adaptation without retraining. Experiments for multiple users show that eye gaze can be estimated accurately under natural head movement via the proposed technique.

1 Introduction

Eye gaze, representing where a person is looking, reveals a person's focus of attention and interest. Eye gaze has been used in various applications such as Human Computer Interaction [1], [2], Human Cognition Study [3], etc. However, its use is confined to controlled environments and its usability under natural environments still needs to be improved.

Various techniques [4], [1], [5] have been proposed to estimate eye gaze based on eye images. Usually, these techniques are composed of two major components: feature extraction and gaze mapping function approximation. In feature extraction, the features that typically characterize the eyeball movements are extracted from the eye images. These features usually include pupil position, limbus position, relative positions of purkinje images, etc. The gaze mapping function is approximated to encode the relationship between the extracted features and the gaze points in an object. Once the gaze mapping function is obtained, the gaze points of the users are estimated from the extracted features via the derived gaze mapping function.

Most existing gaze tracking techniques differ only in the input features and models utilized for the gaze mapping function. For example, in [6], a linear mapping function from the vector between the eye corner and the iris center to the gaze angle is utilized. In [4], a second order polynomial mapping function from the vector between the pupil center and the glint (corneal reflection) center to the gaze point is constructed. In [7], even a higher order polynomial function is employed to model the gaze mapping function. Unfortunately, these extracted eyeball movement features, such as the corner-iris vector, the pupil-glint vector, vary significantly with head movement. But, the head motion information is not considered as an input to the gaze mapping functions built by most of the existing gaze tracking techniques. Therefore, existing methods are very sensitive to head motion and require the user to either have the ability of keeping the head fixed [4] or use a chin-rest or bite-bar to constrain the head movement [7].

The few previous attempts to integrate the head movement information into the gaze mapping function do not estimate the eye gaze under natural head movement with high accuracy. In [8], Artificial Neural Networks (ANN) are utilized to obtain a gaze mapping function by using the eye image as the input. Unfortunately, the eye image varies significantly with eyelid movement, individuals, head position, face orientation, illumination, etc. Therefore, the obtained gaze mapping function based on the eye image is hard to function well. In [2], Generalized Regression Neural Networks (GRNN) are constructed to obtain a gaze mapping function based on a set of pupil geometric features. Part of features are expected to account for the head movement. However, these features can only infer the 3D face orientation rather than the 3D head position. The 3D head position is the key factor that affects the gaze mapping function. Thus, the obtained gaze mapping function is still sensitive to the head motion, which explains why the approximate 5° low gaze accuracy is achieved by this proposed technique.

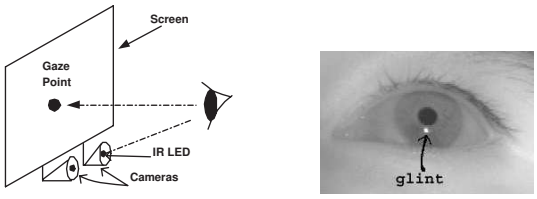
Our approach builds an eye gaze tracking system that allows natural head movement without gaze mapping function calibration. First, the factor that accounts for the head movement effect is extracted. Then, the head movement factor is utilized as an input to build a generalized gaze mapping function. Since the generalized gaze mapping function is highly nonlinear and its complicated nonlinear structure can not be assumed in advance, traditional regression methods are inadequate. However, the Support Vector Regression

Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

(SVR) technique [9], [10] can be used to model highly nonlinear functions efficiently and accurately by using kernel functions. Therefore, SVR models are utilized to approximate the nonlinear gaze mapping function. This generalized gaze mapping function can be readily adapted to a new user using a simple personal adaptation procedure. Experiment results demonstrate that the SVR approach produces gaze mapping functions that track the eye gaze very well.

2 Generalized PCCR Technique Under Natural Head Movement

Among the existing gaze tracking techniques, the PCCR technique is the most popular approach due to its simplicity and reasonable accuracy. The gaze point on the screen, an (x, y) coordinate, is calculated by tracking the relative position of the pupil center and a speck of IR light reflected from the eye cornea, technically known as the "glint" as shown in Figure 1 (b). Specifically, first, computer vision techniques are used to extract the pupil center and the glint center accurately [11] from the eye images. Then, the pupil center and the glint center are connected to form a 2D pupil-glint vector V. After extracting the pupil-glint vectors, a calibration procedure is proposed to acquire a gaze mapping function that can map the extracted pupil-glint vector to the user's gaze point on the screen. Usually, the gaze mapping function is modelled as a linear or nonlinear equation [4], [5]. The coefficients of the gaze mapping function are estimated from a set of pairs of pupil-glint vectors and screen gaze points with known positions. These pairs are collected in the calibration procedure, in which the user is required to visually follow a shining dot as it displays at these predefined locations on the computer screen, while keeping his head as still as possible. If the user doesn't move his/her head significantly after the gaze calibration, the user's gaze point in the screen can be estimated accurately from the extracted pupil-glint vector via the calibrated gaze mapping function.



(a) (b)

Fig. 1. (a) The configuration of eye gaze tracking system. (b) Eye image with corneal reflection (glint)

Several systems [1], [4] have been built via the above scheme, including most commercially available eye gaze tracking systems [5]. High accuracy of the eye gaze tracking results can be achieved if there is little or no head movement. But as the head moves away from the original position where the user performed the calibration, the accuracy of these gaze tracking systems drops dramatically. The detail data in [4] show how the calibrated gaze mapping function decays as the head moves away from its original position.

2.1 Head Movement Effect Compensation

Specifically, when the user is looking at the same point S in the computer screen at two different 3D eye positions O_1 and O_2, the generated pupil-glint vectors V_1 and V_2 in the eye images are significantly different. In fact, the head movement is responsible for this pupil-glint vector variation or difference. If the pupil-glint vector V_2 is used as an input to the gaze mapping function at the position O_1 where the calibration is performed, inaccurate gaze point will be estimated. Hence, the head movement effect on the pupil-glint vectors cannot be ignored during the eye gaze estimation.

However, if the gaze mapping function f_{O_2} at the eye position O_2 can be obtained automatically, then the screen point can be estimated accurately from V_2 via the gaze mapping function f_{O_2}. This is equivalent to finding a function F that can provide a gaze mapping f_{O_i} for each eye position O_i automatically. The function F is called as a generalized gaze mapping function, and it is expressed as follows:

$$S = F(V, O_i) = f_{O_i}(V) \tag{1}$$

Since the generalized gaze mapping function F will provide the gaze mapping function f_{O_i} for each new eye position O_i dynamically as the head moves, the screen gaze can be estimated accurately under the natural head movement. In the next section, the SVR regression technique is proposed to find the generalized gaze mapping function F successfully.

3 Nonlinear Gaze Mapping Function Approximation via SVR

Intuitively, the generalized gaze mapping function F is a highly nonlinear function with a complicated structure that is not know *a priori*. However, the Support Vector Regression (SVR) technique is an effective tool for the nonlinear function approximation without the assumption of a prior parametric model. Therefore, SVR is utilized to approximate generalized gaze mapping function F. First, SVR is briefly reviewed as follows.

3.1 Support Vector Regression

Support Vector Machines have been successfully used for nonlinear regression and function approximation [9], [10]. The training set is denoted $(x_1, y_1), ..., (x_N, y_N)$, where $x_i \in X$, $y_i \in R$, N is the size of training data, and X denotes the space of the input samples. In SVR, the data are mapped to a high-dimensional feature space and a linear regression function is computed in the feature space corresponding to a nonlinear function in the original space. The regression function has the following form:

$$y = f(x) = (w \cdot \phi(x)) + b. \tag{2}$$

where $(\cdot)$ denotes the inner product in the mapped feature space of the input space via a nonlinear map function ϕ, which can have a very high dimensionality. The training algorithm constructs the parameters, w and b, by minimizing the regularized regression risk for the training data, R_{reg}, based on the empirical risk:

$$R_{reg}(f) = \frac{1}{2}\|w\|^2 + C\frac{1}{N}\sum_{i=1}^{N}\tau(f(x_i) - y_i) \quad (3)$$

where C is a constant determining the trade-off, and $\tau()$ is a cost function that measures the empirical risk; here, the ϵ-insensitive loss function is used as the cost function [10].

The optimal regression risk function 3 can be found using quadratic programming in the dual form, which will transform the estimation of function 2 into the estimation of the function as follows:

$$f(x) = \sum_{i=1}^{N}\alpha_i K(x_i, x) + b \quad (4)$$

where α_i is the Lagrange multipliers associated with each training example x_i, and K is the kernel function that describes the inner product in the feature space: $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$. Since the gaze mapping function being estimated is highly nonlinear, the commonly used non-linear kernels are preferred, such as the Gaussian radial basis function (RBF) kernel: $K(x_i, x_j) = exp(\frac{\|x_i - x_j\|^2}{2\sigma^2})$.

3.2 Parameters for SVR

The gaze mapping function maps the pupil-glint vector V and 3D eye position O to an (s_x, s_y) screen coordinate. Usually, O is represented by the 3D eyeball center, which cannot be observed directly in the eye image. However, the pupil center P can be observed from the image and it is less than 8 mm away from O [12]. Hence, the 3D pupil center P is used to represent the 3D eye position O. Thus, the input gaze data vector X_g is represented as $X_g = [d_x, d_y, p_x, p_y, p_z]$, where $[d_x, d_y]$ is the pupil-glint vector V, and $[p_x, p_y, p_z]$ is the 3D coordinate of the pupil center P.

Two video cameras are mounted under the monitor screen in our eye gaze tracking system as shown in Figure 1 (a). An IR illuminator is mounted in the front of one camera to produce the glint in the eye image. Therefore, the pupil-glint vector can be extracted from the captured eye images. In addition, both cameras are calibrated to form a stereo vision system so that the 3D coordinate of the pupil center can be computed. The computed 3D pupil center will concatenate with the extracted 2D pupil-glint vector to serve as the input for the gaze mapping function.

A large amount of training data under different head positions in an approximate $150 \times 150 \times 200mm^3$ head motion volume centered at around 360mm to the camera, is collected to train the SVR for the generalized gaze mapping function. During the training data acquisition, a user is asked to position his/her head at different locations in front of the cameras, generating the data with different 3D eye positions. At each location, the user is asked to gaze at a moving object that will display at 15 predefined locations around the computer screen. In total, 2757 samples composed of the input gaze parameter vector X_g and its corresponding screen gaze point (s_x, s_y) are collected for training. In addition, another set of 30 different locations around the screen are defined, and a set of 832 samples are collected while the user is asked to gaze at them one by one under different head positions. These 832 samples will serve as testing samples.

Given sufficient training samples, we believe that a unique nonlinear function that maps the input gaze parameter vectors to the screen gaze points hidden in the training samples can be captured by the SVR model. Since we don't know how complicated the mapping function will be, different kernels and their associated parameter settings need to be manually tested so that the optimal kernel and its parameter settings can be selected to estimate the gaze mapping function accurately.

4 Experiment Results

In this section, we investigated two types of result. In the first set of experiments, we show how the initial generalized gaze mapping function is estimated and investigate its accuracy for various types of support vector machines. In the second set of experiments, we demonstrate how this generalized gaze mapping function can be successfully used by other users without retraining.

4.1 Gaze Mapping Function Estimation

In our experiments, the training set includes 2757 samples while the testing set includes 832 samples. The accuracy of the approximated generalized gaze mapping function by SVR model is characterized by the errors of the estimated gaze points, which are represented by the absolute value of the difference between the actual screen gaze point and the estimated screen gaze point. For simplicity, the mean μ and standard deviation σ are computed to characterize the errors.

Due to a significant difference in horizontal and vertical spatial gaze resolution, two different SVR models are trained for the horizontal and vertical gaze coordinates s_x and s_y, respectively. Table I lists the results on the testing set for three different SVR kernels, whose parameter settings were adjusted to produce the highest accuracy on the training set. It clearly tells that the best accuracy is achieved with the use of Gaussian RBF kernel.

TABLE I

Experiment results for different kernels

	X-coordinate (mm)	Y-coordinate (mm)
kernel type	Errors ($\mu \pm \sigma$) on Testing Set	Errors (u,σ) on Testing Set
linear	(11.5 $\pm$ 9.3)	(11.1 $\pm$ 8.4)
poly.	(6.9 $\pm$ 5.3)	(8.0 $\pm$ 6.7)
RBF	(5.2 $\pm$ 4.0)	(6.6 $\pm$ 6.3)

Therefore, via the obtained Gaussian RBF kernel with its optimal parameter settings, the generalized gaze mapping function can be approximated very well. The average horizontal and vertical errors are approximately 5.2 mm and 6.6 mm respectively as shown in Table I. Since the testing samples are significantly different from the training samples, the testing error provides a good estimate of the accuracy of the estimated gaze mapping function in future use.

4.2 Personal Adaptation

The size of human eyeball varies considerably among individuals, but the optical function for each individual is modelled similarly. Specifically, the eye cornea is modelled as a convex mirror [12] that has the equivalent power of the eye. In addition, the rotation angles of the eyeballs with different sizes are approximately equivalent when they are located at the same 3D position in front of the camera while looking at a same screen point.

The rotation angle of the eyeball is characterized by the pupil-glint vector in the eye image. Because of size difference among the human eyeballs, their generated pupil-glint vectors will be different from individuals, even though the rotation angles are exactly same for them. In the following, we propose a simple calibration procedure to eliminate the size difference for individual eyeball. Specifically, during calibration, each new user is required to gaze at a predefined screen point S_r while positioning his/her eye at a predefined 3D position P_r, assuming a pupil-glint vector V_n extracted. Since the pupil-glint vector V_r for the reference eyeball (the one that used to build the generalized gaze mapping function) is known in advanced, a personal adaptation scale factor k can be computed by $k = \frac{V_r}{V_n}$. For each individual, this personal adaptation scale factor k can be used to compensate for the effect of the eyeball size difference on the generated pupil-glint vector. As a result, the learned generalized gaze mapping function F can be still utilized to estimate the gaze points from the scaled pupil-glint vectors. No retraining of F is required.

To test the accuracy of the proposed gaze estimation technique, six new individuals were asked to participate in the experiment. During the experiment, the proposed personal adaptation procedure is done first so that the generalized SVR gaze mapping function can be personalized to each participant. Subsequently, the participant is asked to follow a moving object that will display around the computer screen. With the use of personal adaptation, the gaze accuracy can be improved significantly for a new individual. For example, for the first participant, the average gaze estimation error decreases dramatically from 15.8 mm to 5.7 mm horizontally and from 11.6 mm to 5.5 mm vertically. Table II provides the average gaze estimation errors for all the participants in the experiment. In general, for a new individual, the gaze estimation error is less than 10 mm (approximately 1.5°), which is sufficiently accurate for most Human Computer Interaction applications.

TABLE II
The gaze estimation errors for different subjects (mm)

Sub.	X ($\mu \pm \sigma$)	Y ($\mu \pm \sigma$)	Sub.	X ($\mu \pm \sigma$)	Y ($\mu \pm \sigma$)
1	5.7 ± 5.3	5.5 ± 5.2	4	9.3 ± 8.9	10.1 ± 9.2
2	7.4 ± 6.9	8.2 ± 7.3	5	10.2 ± 9.3	11.5 ± 10.6
3	8.1 ± 7.5	8.9 ± 8.6	6	12.0 ± 11.1	12.6 ± 11.3

5 Conclusion

We propose a novel method to track eye gaze under natural head movement without gaze mapping function calibration, which represents a significant advance over prior techniques. Unlike existing eye tracking techniques, the user does not need to keep his head unnaturally still; our system can track the eye gaze while the user moves naturally. Previous systems required cumbersome calibration or retraining for each user to obtain a person-dependent gaze mapping function. Our system builds a person-dependent gaze mapping function by individualizing a generalized gaze mapping function approximated via SVR to a new user automatically through a simple personal adaptation. Experiment results show that approximate 1.5° high gaze accuracy under natural head movement is achieved via the proposed technique.

References

[1] R.J.K. Jacob and K. S. Karn, "Eye tracking in human-computer interaction and usability research: Ready to deliver the promises," 2003, Oxford, Elsevier Science.

[2] Z. Zhu and Q. Ji, "Eye and gaze tracking for interactive graphic display," *Machine Vision and Applications*, vol. 15, no. 3, pp. 139–148, 2004.

[3] S.P. Liversedge and J.M. Findlay, "Saccadic eye movements and cognition," *Trends in Cognitive Science*, vol. 4, no. 1, pp. 6–14, 2000.

[4] C.H. Morimoto and M. R.M. Mimica, "Eye gaze tracking techniques for interactive applications," *CVIU, special issue on eye detection and tracking*, vol. 98, no. 1, 2005.

[5] LC Technologies Inc, "The eyegaze development sytem," http://www.eyegaze.com.

[6] J. Zhu and J. Yang, "Subpixel eye gaze tracking," in *International Conference on AFG*, 2002.

[7] Z. Cherif, A. Ali, J. Motsch, and M. Krebs, "An adaptive calibration of an infrared light device used for gaze tracking," in *IEEE Conference on Instrumentation and Measurement Technology*, 2002.

[8] S. Baluja and D. Pomerleau, "Non-intrusive gaze tracking using artifical neural networks," in *Technical Report CMU-CS-94-102,School of Computer Science, CMU*, 1994.

[9] V. N. Vapnik, "The nature of statistical learning theory," *Springer Verlag, New York*, 1995.

[10] A. J. Smola and B. Scholkopf, "A tutorial on support vector regression," in *NeuroCOLT Technical Report NC-TR-98-030,Royal Holloway College, University of London,UK*, 1998.

[11] Z. Zhu and Q. Ji, "Robust real-time eye detection and tracking under variable lighting conditions and various face orientations," *CVIU, special issue on eye detection and tracking*, vol. 98, no. 1, 2005.

[12] Clyde W. Oyster, "The human eye: Structure and function," *Sinauer Associates, Inc.*, 1999.