-- --

An Eigenspace Approach to Eye-Gaze Estimation

George Bebis† and Kikuo Fujimura‡

†Department of Computer Science, University of Nevada, Reno, NV
‡Honda R&D, Fundamental Research Labs, Mountain View, CA

Abstract

In this paper, we present a new approach to eye-gaze estimation using eigenspaces. The context of our application is monitoring car drivers. Within this context, eye-gaze information will be an important component in the development of systems that monitor the driver and alert him/her of any insecure driving conditions. Based on the requirements of our application, accurate gaze estimates are not really necessary. On the other hand, speed is very important. The approach we have taken involves classifying gaze into five directions ("straight", "up", "down", "left", and "right"). Each gaze direction is modeled by a distinct eigenspace. Novel gaze directions are classified as one of the above five gaze directions by computing the distance of the novel direction from each eigenspace. The eigenspace associated with the smallest distance determines the classification result. The proposed approach is very simple and fast. Also, it is not intrusive and does not required extracting any special features. Our preliminary results illustrate that it is a promising approach.

Keywords: eye-gaze estimation, principal components analysis, eigenspace.

1. Introduction

The problem of eye-gaze estimation has received a lot of attention lately mainly because of its potential applications in building advanced interaction devices to improve human-machine communication [1]-[3]. The goal of eye-gaze estimation is to determine where a subject is looking from the appearance of the subject's eyes. The context of our application is using eye-gaze for monitoring car drivers. Eye-gaze information regarding the movement of a driver's line of sight may have the potential to indicate a driver's intention and his/her physical or mental conditions. For normal driving, the line of sight is front. When people are drowsy or drunken, for example, their visual awareness cannot cover a wide enough area. Consequently, eye-gaze information will be an important component in the development of systems that monitor the driver and alert him/her of any insecure driving conditions.

In general, the problem of eye-gaze estimation involves two steps: first, the orientation of the subject's head needs to be estimated ("global" gaze direction) and second, the orientation of the subject's eyes within their sockets needs to be estimated ("local" gaze direction). The overall gaze direction can be obtained by integrating the global and local directions. Most techniques in the literature deal with the "local" direction only, assuming that the position of the head is fixed. Although this might be a valid assumption in certain applications (e.g., medical applications), it is a rather restrictive assumption in many other applications. To build a general purpose eye-gaze estimation system, both problems need to be tackled. Here, we concentrate on the estimation of the "local" gaze direction only.

Methods for eye-gaze estimation usually rely on techniques such as measuring the reflection of some light that is shone onto the eye, measuring the electric potential of the skin around the eyes or applying special contact lenses [4][5]. Although these techniques work relatively well, they are intrusive, thus, not acceptable in every application domain. Recently, a lot of emphasis has been on using no-intrusive computer vision techniques [1]-[3]. These techniques are aimed at extracting visual characteristics from images of the subject. Several critical steps are involved when gaze estimation is based on image data. First, the face of the subject needs to be detected. Methods based on neural networks [6] and skin-color [3] are very popular in this case. Then, the locations of the eyes need to be found within the face region. Among the numerous methods proposed for this in the literature [1]-[3], the method in [7] is fast and reliable since it is based on the fact that humans must periodically blink to keep their eyes moist. After the eyes have been detected, the last step is the gaze estimation step.

The most common probably approach to gaze estimation is based on the reflection of some light shone onto the eye. Specifically, the gaze direction is estimated from the relative position between the pupil and the glint (i.e., the brightest light spot on the eye due to light reflection) [2][5][8]. This approach is not completely non-intrusive and requires good localization of the pupil and the glint. To avoid localizing these features explicitly, the use of neural networks has been proposed [8]. In [9], the gaze direction was estimated using the shape of the iris or the pupil in the image. Since circles are projected on the image plane as ellipses, the idea was to estimate the pose of the iris from the distortion of its projection (i.e., ellipse) in the image. Then, the gaze is estimated based on the recovered pose. The main challenge with this approach is that it requires a close view of the eye.

Depending on the application at hand, the accuracy of the eye-gaze estimation system can vary. In the area of human-computer interfaces, for example, eye-gaze is used as an input device to control the screen cursor or for menu selections. In this case, eye-gaze estimates must be very accurate. This is also the case when eye-gaze is used in medical applications. In the context of our application, however, we believe that a rough estimate of the driver's eye-gaze, computed fast and reliably, will probably be enough. It should be mentioned that eye-gaze is not the only characteristic taken into consideration when monitoring a driver. Other characteristics include head inclination, degree of eye openness, sluggish in facial expression, and sagging posture [10].

In this paper, we propose an eigenspace approach for eye-gaze estimation. Eigenspace methods have attracted a lot of interest lately mainly because of their simplicity, speed, and relative success in numerous application areas, especially in face recognition (*eigenface approach*) [11]-[14]. The main idea is very simple: Principal Components Analysis (PCA) [11][12] is used to linearly project images of objects to a low dimensional subspace (eigenspace). This subspace is defined by the principal components (eigenvectors) of the distribution of images (i.e., the most important eigenvectors of the covariance matrix). Using this approach each object is represented as a linear combination of eigenvectors. Objects under novel appearances are classified by computing their distance from the eigenspace. Here, our intention is to classify eye-gaze into five directions: **(1)** looking straight, **(2)** looking up, **(3)** looking down, **(4)** looking left, and **(5)** looking right. Each gaze direction, is represented by a separate eigenspace. New gaze directions are classified to one of the five predefined directions by computing the smallest distance among the distances from all the eigenspaces. The proposed approach is very simple, fast, and does not require any special features to be extracted (e.g., glint).

The rest of the paper is organized as follows: In Section 2, we present an overview of the proposed approach. Section 3 contains a review of the eigenspace approach while Section 4 presents our methodology in more detail. In Section 5, we present our preliminary results. Finally, Section 6 contains our conclusions.

2. Method Overview

In the proposed approach, eye-gaze is estimated by quantizing it into five directions: **(1)** looking straight, **(2)** looking up, **(3)** looking down, **(4)** looking left, and **(5)** looking right. A finer quantization is possible if better accuracy is required. Each eye-gaze direction is represented by an eigenspace which is built using face images, containing mostly the eyes, of people looking into the corresponding direction. Given a novel image (a person looking at some direction), we extract the subimage containing the eyes and compute the distance (*dffs*, see next section) from each one of the eigenspaces. The direction associated with the eigenspace having the smallest distance determines the gaze direction of the person in the novel image. Figure 1 illustrates this idea. In the next section, we review the theory of eigenspace-based recognition.

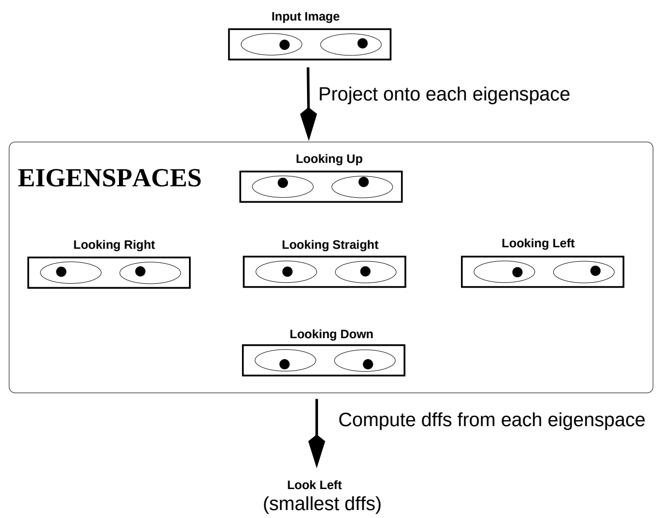

Figure 1. The proposed approach for coarse eye-gaze estimation.

3. Review of the eigenspace approach

The original formulation of the eigenspace method is based on Principal Component Analysis (PCA) [11][12], a standard statistical technique for reducing the dimensionality of data while attempting to preserve as much of information as possible in terms of variance. The key idea is to represent each data in a low dimensional space defined by the most important eigenvectors of the covariance matrix of the data distribution. A complete description of the eigenspace approach, applied on face recognition, can be found in [11][12].

In this approach, each image $I(x, y)$ is represented as a $N \ x \ N$ vector Γ_i. First, the average image Ψ is computed: $\Psi = \frac{1}{R}\sum_{i=1}^{R}\Gamma_i$, where R is the number of images in the training set. Next, the difference Φ of each image from the average image is computed $\Phi_i = \Gamma_i - \Psi$, and the covariance matrix is estimated:

$$C = \frac{1}{R}\sum_{i=1}^{R}\Phi_i\Phi_i^T = AA^T$$

where, $A = \left[\Phi_1\ \Phi_2\ \ldots\ \Phi_R\right]$. The eigenspace can then be defined by computing the eigenvectors u_i of C. Since C is very large ($N^2\ x\ N^2$), computing its eigenvectors will be very expensive. Instead, we can compute v_i, the eigenvectors of $A^T A$, an $R\ x\ R$ matrix. Then, u_i can be computed from v_i as follows (the details are given in [11]):

$$u_i = \sum_{j=1}^{R} v_{ij}\Phi_j, \qquad j = 1, \ldots, R$$

Usually, we only need to keep a smaller number of eigenvectors R', corresponding to the largest eigenvalues. Given a new image Γ, we subtract the mean ($\Phi = \Gamma - \Psi$) and we compute its projection: $\hat{\Phi} = \sum_{i=1}^{R'} w_i u_i$, where $w_i = u_i^T \Phi$ are the coefficients of projection.

Let us assume that our training set contains face images. A new image is considered to be a face if the mean square error (called the *distance from face (dffs)*) between its representation using the most important eigenvectors and its normalized counterpart (e.g., the difference of the input image and the mean image), is small.

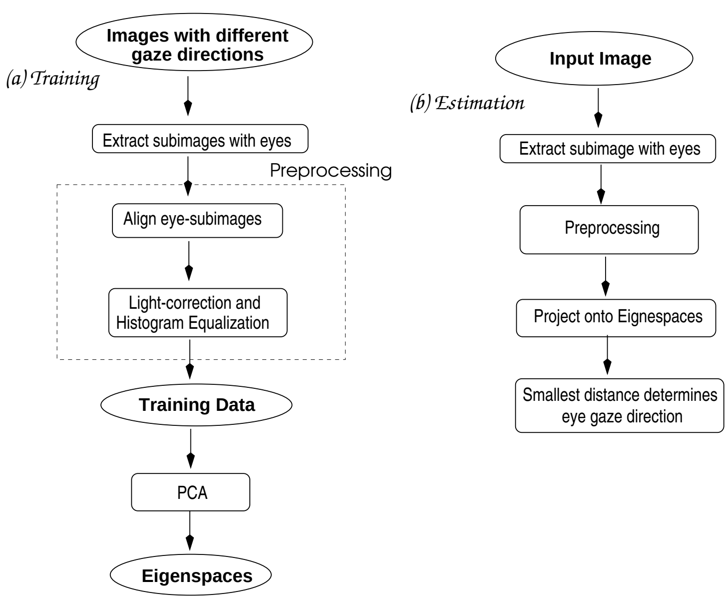

Figure 2. The main steps during training and estimation phases.

4. Methodology

In this section, we discuss the steps of the proposed methodology (see Figure 2). To build the eigenspaces (*training phase*), we use images (i.e., subimages containing mostly the eyes) of people looking at all five directions. During eye-gaze estimation (*estimation phase*), the image of a subject is presented to the system. Then, the subject's eye-gaze is estimated by computing the distance of the subimage containing the eyes from each eigenspace. The smallest distance determines the classification.

4.1. Preprocessing

To compute the eigenspaces corresponding to different gaze directions, first we extract the subimages corresponding to the eye-region and then we align them together. The procedure used to align the subimages is similar to that used in [6]. Specifically, the center of the eyes and the tip of nose (picked manually) are used to normalize each eye-region to same scale, orientation and position. The steps are described below:

Step1: Let $\bar{F}$ be a vector which contains the average positions of each labeled feature over all subimages. Initialize $\bar{F}$ with the feature locations in the first subimage F_1.

Step2: The feature coordinates in $\bar{F}$ are transformed so that the average locations of the eyes (P_1 and P_2) and tip of the nose (P_3) appear at predetermined locations (P_1^f,P_2^f,P_3^f respectively) in a $N\ x\ M$ window (see Figure 3). A 40 x 50 window was used in our experiments. An affine transformation is used to register the images:

$$P_1^f = AP_1 + b$$

$$P_2^f = AP_2 + b$$

$$P_3^f = AP_3 + b$$

The above equations can be rewritten as

$$Pc_1 = p_x$$

$$Pc_2 = p_y$$

where

$$P = \begin{bmatrix} X_1 & Y_1 & 1 \\ X_2 & Y_2 & 1 \\ X_3 & Y_3 & 1 \end{bmatrix}$$

$$p_x = \begin{bmatrix} X_1^f \\ X_2^f \\ X_3^f \end{bmatrix},\ p_y = \begin{bmatrix} Y_1^f \\ Y_2^f \\ Y_3^f \end{bmatrix},\ c_1 = \begin{bmatrix} a_{11} \\ a_{12} \\ b_1 \end{bmatrix},\ c_2 = \begin{bmatrix} a_{21} \\ a_{22} \\ b_2 \end{bmatrix}$$

(c_1 and c_2 are the parameters of the affine transformation).

Step3: For every subimage i in the training set, we compute the best affine transformation to align the features (eyes, tip of the nose) F_i with the average feature locations $\bar{F}$. Let's call the aligned feature locations $F_i^{'}$.

Step4: Update $\bar{F}$ by averaging the aligned feature locations $F_i^{'}$ for each subimage i.

Step5: If the error between $\bar{F}$ and $\bar{F}$, calculated in the previous iteration, is less than some threshold, then stop; otherwise go to step 2.

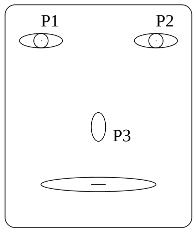

Figure 3. A typical face showing the features of interest.

The alignment algorithm converges usually within seven iterations. For each subimage, it yields an affine transformation that maps that subimage to the 40 *x* 50 window. To avoid gaps in the normalized subimage, each point in the desired subimage was actually determined through the inverse affine transformation. Figure 4 shows some examples of images before and after normalization.

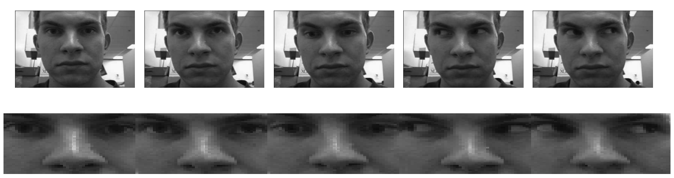

Figure 4. Examples before and after normalization.

After the alignment of all subimages, each subimage is processed to account for different lighting conditions [6]. First, we fit a linear model to the intensities of the image, having the following form:

$$f(x, y) = ax + by + cxy + d$$

where $f(x, y)$ denotes the image and a,b,c,d are the coefficients to be determined. To solve for the coefficients, we use a least squares approach. Then, the linear fit image is subtracted from the original image to account for lighting differences. Then, histogram equalization is performed to improve contrast. The results of these preprocessing steps are illustrated in Figure 5.

Figure 5. Examples showing images after light correction and histogram equalization.

4.2. Eigenspace representation

All the images in the training set were preprocessed as explained in the previous section. Then, these images were subjected to PCA to generate the eigenspaces. Figure 6 shows the most important eigenimages of the "looking straight" set.

Figure 6. (top) Some of the images in the dataset "look straight", (bottom) the mean image and the top four eigenimages of this dataset.

4.3. Eye-gaze estimation

During eye-gaze estimation, the image of the subject is captured and his face is detected using a skin-color algorithm [3]. Then, the eyes are detected and a subimage around the eyes is cropped out and mapped to a 40 *x* 50 window, followed by light-correction and histogram equalization. No alignment is performed at this step. Also, the eyes are detected using an iterative thresholding approach [3]. To estimate the gaze direction associated with the input subimage, we project it onto each eigenspace. Then, for each eigenspace, we reconstruct the input subimage using the most important eigenvectors of this eigenspace and we compare it against the original subimage. The comparison yields an error (dffs) which is considered to be the distance of the subimage from the eigenspace under consideration. The smaller the error the closer the image is to that eigenspace. The gaze direction associated with the eigenspace having the smallest distance determines the gaze direction of the subject.

5. Experimental Results

In this section, we report a number of preliminary experimental results to illustrate the performance of the proposed approach. In these experiments, we captured 65 images (5 images from 13 individuals, one image per gaze direction). For training, we used 8 of the individuals (40 images) while the rest 5 (25 images) were kept for testing. Figure 7 shows the images used to construct the "looking straight" eigenspace. It should be mentioned that the lighting conditions were not changed significantly in our experiments. Although we apply several normalization steps to account for lighting changes (see section 4), the original eigenspace approach -used here- is not very robust to different lighting conditions. Improved versions of the the eigenspace approach, however, are more successful in tolerating changes in lighting [14].

Figure 7. The training data used to build the eigenspaces ("look-straight").

Figures 8-12 show the test cases considered. Since our focus is on estimating the "local" gaze direction only, we have asked all the subjects to move their eyes only, trying to keep their head at a fixed location. Besides the subjects' effort to keep their head still, there are differences in the head orientation as we can see below (most of them, however, were very well tolerated by the proposed approach). Tables 1-5 show the distance of the input images (i.e., eye-regions only) from each eigenspace (the smallest distance is shown in boldface).

In test case 1, all five gaze directions were estimated correctly (the smallest distance corresponds to the eigenspace associated with the correct gaze direction). What is interesting to note about this subject is that he has not kept his head still while moving his eyes. Despite the changes in head orientation, the proposed approach was able to estimate the gaze direction correctly.

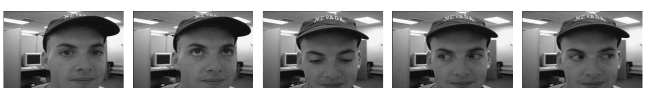

Figure 8. The first test case.

Table 1. The smallest computed distances for test1 (Fig 8).

Smallest Distances					
Input	Front	Left	Right	Up	Down
Front	**1821.5**	2044.8	2195.3	2012.9	2062.2
Left	2134.0	**1772.7**	2459.3	2264.6	2348.7
Right	2109.1	2444.3	**1908.4**	2304.9	2437.0
Up	2366.2	2425.9	2450.6	**2262.5**	2498.7
Down	2101.3	2227.5	2246.9	2232.4	**1711.8**

Test case 2 is a rather good test case since the subject has kept his head as still as possible. All five directions have been estimated correctly.

Figure 9. The second test case.

Table 2. The smallest computed distances for test2 (Fig. 9).

Smallest Distances					
Input	Front	Left	Right	Up	Down
Front	**1560.8**	1930.1	1821.7	1679.3	1952.1
Left	1936.1	**1624.2**	1993.9	1827.9	1919.3
Right	1739.0	1943.5	**1698.4**	1804.5	1714.3
Up	1845.1	2166.2	2076.2	**1705.9**	2327.5
Down	1686.6	1785.3	1758.3	1686.4	**1658.3**

In test case 3, all the gaze directions were estimated correctly without any problems. The subject has kept his head still and the different gazes are clearly distinguishable.

Figure 10. The third test case.

Table 3. The smallest computed distances for test3 (Fig 10).

Smallest Distances					
Input	Front	Left	Right	Up	Down
Front	**1609.1**	1933.4	1870.1	1650.2	1734.9
Left	1618.7	**1612.5**	1848.2	1626.7	1722.1
Right	1739.0	1943.5	**1698.4**	1804.5	1714.3
Up	1570.0	1888.5	1729.8	**1567.5**	1717.9
Down	1580.0	1818.6	1746.2	1656.6	**1450.4**

In test case 4, the "look-left" gaze direction was misclassified as "look-up". However, the difference in the distance computed for the "look-left" direction is very small.

Figure 11. The fourth test case.

Table 4. The smallest computed distances for test4 (Fig 11).

Smallest Distances					
Input	Front	Left	Right	Up	Down
Front	**1514.2**	1920.4	1657.9	1521.1	1611.5
Left	1618.5	1579.7	1866.3	**1530.8**	1783.9
Right	1605.8	1780.0	**1521.5**	1566.0	1802.5
Up	1666.4	1853.7	1750.4	**1470.2**	1734.6
Down	1582.4	1799.7	1786.7	1579.8	**1465.9**

The subject of test case 5 was asked to sit not very close to the camera. This is to test the robustness of the method to scale changes. Among the five input directions, three were classified correctly and the other two incorrectly. The "look-up" direction was classified as "look-straight". In fact, looking carefully at the test image, it is not clear whether the subject is looking up or straight. Also, the "look-down" test case was misclassified.

Figure 12. The fifth test case.

Table 5. The smallest computed distances for test5 (Fig. 12).

Smallest Distances					
Input	Front	Left	Right	Up	Down
Front	**1151.4**	1329.7	1333.6	1323.5	1238.1
Left	1345.3	**1246.7**	1525.2	1377.8	1501.1
Right	1163.2	1455.0	**1152.1**	1201.2	1352.7
Up	**1124.8**	1343.3	1279.6	1164.4	1280.5
Down	**1137.5**	1296.9	1320.6	1221.3	1238.1

6. Conclusions

In this paper, we presented a number of preliminary results on the problem of eye-gaze estimation using an eigenspace approach. From additional experiments we have performed, it seems that the most often confused cases are the "look-straight", "look-up" and "look-down". This is not unreasonable since the differences among these cases are not that great. We believe, however, that one of the reasons that has contributed to this problem is the fact that the camera was set up at the same level with that of the subjects' face. We believe that moving the camera lower will allow us to capture images where the differences among the confused directions are better emphasized.

In addition, our training set was rather small containing only 8 subjects. Increasing the size of the training set will improve the performance of the method. Another problem with the subimages used in our experiments is that they contain not only the eyes but also the eyebrows and the nose of the subject. This information, however, might confuse the eigenspace approach. Finally, we believe that improved versions of the eigenspace approach (i.e., using Linear Discriminant Analysis [14]) will further increase classification accuracy.

Acknowledgment: This work was supported by a grant from HONDA R&D Americas, Inc.

References

[1] C. Colombo and A. Del Bimbo, "Interacting through eyes", *Robotics and Autonomous Systems*, vol. 19, pp. 359-368, 1997.

[2] T. Hutchinson et al., "Human-computer interaction using eye-gaze input", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 6, pp. 1527-1534, 1989.

[3] R. Stiefelhagen, J. Yang, and A. Waibel, "A model-based gaze tracking system", *IEEE International Joint Symposia on Intelligence and Systems*, pp. 304-310, 1996.

[4] R. Jacob, "Eye tracking in advanced interface design", in *Virtual Environments and Advanced Interface Design* W. Barfield and T. Furness III (eds.), 1995.

[5] Y. Ebisawa, "Unconstrained pupil detection technique using two light sources and the image difference method", *Visualization and Intelligent Design in Engineering*, pp. 79-89, 1989.

[6] H. Rowley, S. Baluja and T. Kanade, "Neural Network-Based Face Detection", *IEEE Transactions on PAMI*, Vol. 20, No. 1, January 1998.

[7] J. Crowley and Y. Demazeau, "Principles and techniques for sensor data fusion", *Signal Processing*, vol. 23, pp. 5-27, 1993.

[8] S. Baluja and D. Pomerleau "Non-intrusive gaze tracking using artificial neural networks", *Advances in Neural Information Processing Systems (NIPS)*, vol. 6, 1994.

[9] M. Perrone, "Estimating human gaze direction", Computational Vision and Active Perception Laboratory, Royal Institute of Technology, Sweden.

[10] Q. Ji and G. Bebis, "Visual Cues Extraction for Monitoring Driver's Vigilance", *1st HONDA Initiation Grant Forum: The HIG Symposium*, pp. 48-55, 1999.

[11] M. Turk and A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, Vol. 3, pp. 71-86, 1991.

[12] M. Kirby and L. Sirovich, "Application of the Karhunen-Loe‘ve Procedure for the Characterization of Human Faces",*IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 1, pp. 103-108, 1990.

[13] G. Bebis, S. Uthiram and M. Georgiopoulos, "Genetic Search for Face Detection and Verification", IEEE International Conference on Intelligence, Information and Systems (ICIIS'99), pp. 360-368, Maryland, 1999.

[14] D. Swets and J. Weng, "Using Discriminant Eigenfeatures for Image Retrieval", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 831-836, 1996.