Towards pervasive eye tracking using low-level image features

Yanxia Zhang*
Lancaster University

Andreas Bulling†
University of Cambridge
& Lancaster University

Hans Gellersen‡
Lancaster University

Abstract

We contribute a novel gaze estimation technique, which is adaptable for person-independent applications. In a study with 17 participants, using a standard webcam, we recorded the subjects' left eye images for different gaze locations. From these images, we extracted five types of basic visual features. We then sub-selected a set of features with minimum Redundancy Maximum Relevance (mRMR) for the input of a 2-layer regression neural network for estimating the subjects' gaze. We investigated the effect of different visual features on the accuracy of gaze estimation. Using machine learning techniques, by combing different features, we achieved average gaze estimation error of 3.44° horizontally and 1.37° vertically for person-dependent.

CR Categories: I.4.7 [Image processing and computer vision]: Feature Measurement— [I.5.4]: Pattern Recognition— Applications

Keywords: Gaze estimation, Pervasive eye tracking, Appearance-based, Low-level image features, Machine learning

1 Introduction

We are interested in eye gaze estimation with pervasive equipment, such as web cameras we might find mounted on public displays. Our motivation is to use equipment that we can expect to find in our environments for eye gaze interaction. Naturally, as cameras in our everyday environments are not optimized for gaze tracking, the challenge is to advance methods that are capable of estimating gaze using eye images captured under real-world constraints.

The method we propose is based on extraction of low-level features from images of the eye. Through image transformation into feature space, we are encoding information such as texture and edges in a feature vector that represents an image more compactly (i.e. with reduced dimensions) than a raw pixel image. The feature vector serves as input for a two-layer regression neural network that produces gaze estimates. The neural network requires a priori training to learn the relationship between image features and gaze direction.

In this paper we focus on an investigation into the robustness of our method with respect to a diverse user population. We report on a user study with 17 participants who were selected with a view of maximizing diversity, in terms of different gender, ethnicities, eye shapes, eye lashes and pupil colors. As the focus was on user diversity, other conditions were controlled by mounting a web camera in front of the participant's eye, using a stimulus at a fixed distance, and ensuring consistent illumination. Gaze data was collected for 13 predefined gaze locations.

The collected data was analyzed with five different features as well as their combination, to gain insight into their performance for gaze estimation. We first considered a person-dependent evaluation, in which training and testing were performed on data from the same user. In a pervasive computing world, we consider this approach reasonable for adopting a camera embedded in one's personal devices for eye-tracking. As the system is calibrated for a specific user, reasonably high gaze accuracy would be expected. We also conducted a person-independent evaluation where the system's robustness is tested using leaving-one-person-out cross-validation. As the system is trained by others than the user, it is naturally far less accurate. However, the case is compelling from a pervasive computing perspective, as person-independent tracking would permit eye-based interaction on an ad hoc basis, for instance at a public terminal.

2 Related Work

Work in eye tracking generally aims to achieve best possible accuracy (e.g., [Zhu and Ji 2005; Williams et al. 2006]) with hardware optimized for the task, and careful calibration to the individual user. While we do consider person-dependent tracking as well, we are basing our work on an approach that is generalizable to the person-independent case. In this study, we use a web camera mounted close to the user's eye for experimental control, but in principle we envision our approach to work with cameras operating at larger and varying range. This similar to the use of cameras for eye gaze attention detection (e.g., [Vertegaal et al. 2006]) but we aim to detect not only attention but gaze directional information.

Our approach for processing eye images is *appearance-based*, contrasting *model-based* approaches that use an explicit geometric model of the eye for gaze estimation but typically requiring a dedicated apparatus [Hansen and Ji 2010]. Appearance-based approaches work under normal illumination and directly infer gaze from video images. Previous works have shown the potential of this approach for estimating gaze from low resolution images, under laboratory conditions [Baluja and Pomerleau 1994; Xu et al. 1998; Williams et al. 2006; Sugano et al. 2008]. These works were based on raw pixel images - by representing images as input vectors with raw pixel values for machine learning. Williams *et al.* combined to use steerable filters on eye images and raw pixel data to achieve better accuracy for regression-based gaze estimation [2006]. Zhang *et al.* classified basic gaze directions using a small set of low-level features extracted from video images [2011].

3 Gaze estimation using image features

Typical appearance-based approaches use the entire eye image for gaze estimation. Among them, approaches that use raw pixel values can only represent general information (color/intensity) of the overall image. By transforming the image into another feature space

*e-mail: yazhang@lancaster.ac.uk
†e-mail: andreas.bulling@acm.org
‡e-mail: hwg@comp.lancs.ac.uk

Feature	Description
Color (C)	f_C extracts the red-green (RG) and blue-yellow (BY) color opponencies.
Intensities (I)	f_I extracts the grey scale intensities.
Orientation (O)	f_O is obtained by convolving the intensity image with a set of Gabor filters in four orientations $\{0°,45°,90°,135°\}$.
Haar (H)	f_H represents Haar features using two rectangular patterns which extracts local borders.
Spatiogram (S)	f_S encodes RGB color histogram and their spatial distribution.

Table 1: *The five types of low-level image features used in this work.*

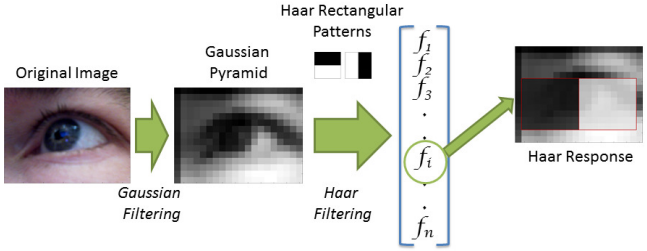


Figure 1: *Extraction of haar-like features. The image was first normalized and then sub-sampled into a Gaussian pyramid by convolution with a 3x3 Gaussian smoothing filter and decimation by a factor of two. Two rectangular patterns were used which detect where the border lies between a dark region and a light region, horizontally and vertically. Each value in the resulting Haar feature vector f_H is a response of the rectangular patterns at a certain position and scale in the image.*

allows us to encode information such as texture, edges, etc., and potentially reduce the data dimension.

3.1 Feature extraction and feature selection

In our approach, we adopted five different types of features, namely *color*, *intensities*, *orientation*, *haar-like* features as well as *spatiogram* (see Table 1). We chose these features because of their low computational complexity as well as their prevalent application in computer vision.

For calculating color f_C, intensities f_I, and orientation f_O features, we used the method from Walther and Koch [2006]. The input image I was processed for low-level features at multiple scales, and centre-surround differences were computed. Three individual feature vectors $f_{K\in\{C,I,O\}} = \{f^{\{i\}}\}_{i=1}^{1200}$ were extracted to represent I in color, intensities and orientation feature space respectively.

Haar-like features have a low computational cost and provide local edge information of an image [Viola and Jones 2001]. They consist of a set of simple rectangular features. Rectangles can be placed at any position and scale within the original image. The sum of the pixels which lie within the white rectangles is subtracted from the sum of pixels in the dark rectangles (see Figure 1). Each feature type can indicate the existence of edges or changes in texture.

While human look at different directions, the color distributions and shape configuration between the pupil and the sclera of the user's eye appeared in images from the camera change. To represent pixel values and spatial distribution of colors in an image simultaneously, we employed the spatio-histogram (spatiogram) [Birchfield and Rangarajan 2005]. It expresses local color patches over entire image. This allows us to encode object information about the texture and shape, as well as the spatial relationships between the pixels, such as the average locations of different color patches.

Given an input RGB image I, let $H^{\{k\}}$ represent the total number of pixels in the k_{th} bin of an ordinary color histogram. We define the mean of the x,y coordinates of all pixels in the k_{th} bin as $C_x^{\{k\}}$ and $C_y^{\{k\}}$. The spatiogram of the image is a vector $f_S = \{H^{\{k\}}, C_x^{\{k\}}, C_y^{\{k\}}\}_{k=(0,0,0)}^{(24,24,24)}$ where the quantization level was fixed to $M = 25$ for each color channel in this work.

Five feature vectors $f_{K\in\{C,I,O,H,S\}}$ were computed from each image I in the database. To yield a fast and efficient gaze estimation, a feature selection procedure is followed instead of directly using the high-dimensional raw vector as input to the neural network. By selectively choosing the essential elements in the feature vector, we can reduce the input dimension and minimize redundancy, while maximizing features' relevance. We employed mRMR (minimum Redundancy Maximum Relevance [Peng et al. 2005]) feature selection on the original feature vector extracted from I, thus reducing the high dimensional image data I into a low dimensional feature vector $z_{K\in\{C,I,O,H,S\}} = \{f^{\{i\}}\}_{i=1}^{m}$ where m = 50.

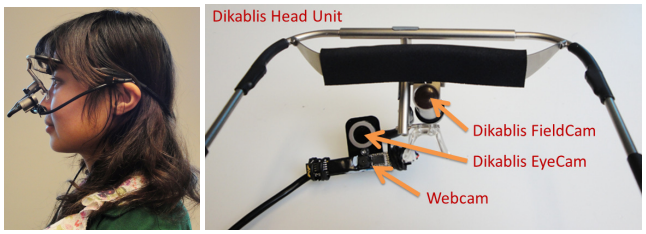
3.2 Gaze estimation using a 2-layer regression neural network (RNN)

Gaze estimation was performed by mapping extracted feature vector $z = \{f^{\{i\}}\}_{i=1}^{m}$ from image I to output gaze location $g = (x, y)$ of the user's. Using raw eye image pixels as the input to a 3 layer feed-forward Artificial Neural Network (ANN) for gaze estimation has been proposed in previous work [Baluja and Pomerleau 1994; Xu et al. 1998]. In this study, the dimension-reduced feature vector $z = \{f^{\{i\}}\}_{i=1}^{m}$ extracted from raw image was supplied as the input to the RNN. The input vector was first normalized to ensure all input features were in the same data range. We adopted a feed-forward neural network model using a 2-layer perceptron with linear output unit activation function to learn the gaze mapping function $g(z)$. This is a two-layer network where the first layer has $tanh()$ unit and the second layer is linear. The RNN was trained on a set of labelled eye image/gaze coordinates pairs by minimizing a sum-of-squares error function using the scaled conjugate gradient optimizer. The number of hidden units was decided by averaging the input and output units size.

4 Data collection and analysis

We conducted a user study to evaluate the performance and accuracy of our gaze estimation technique. 17 participants (five female, 12 male), aged between 18 and 40 years (mean=26.9±6.8) took part in the study. We took particular care to include participants of different ethnicities, with different eye lashes and pupil colors. Specifically, we had nine participants with dark (i.e. brown or black) and eight with bright eyes (i.e. green or blue)). None of the participants wore glasses, but two wore contact lenses during the experiment.

We used a standard webcam (Microdia Sonix USB 2.0) with a resolution of 640x480 pixels and a frame rate of 30Hz. In addition, we used a Dikablis eye tracker from Ergoneers GmbH for collecting



(a) (b)

Figure 2: *(a) Participant wearing the Dikablis eye tracker . (b) The additional webcam is mounted on the head unit to record close-up left eye images [Zhang et al. 2011].*

(a) (b) (c)

Figure 3: *(a) An image from the webcam, (b) a scene image from the Dikablis, and (c) a screenshot of the experimental stimulus. A red point is displayed in order at 13 different locations on the screen. The neighboring stimuli are spaced 10.75° in horizontal and 6.9° in vertical of visual angle from each other [Zhang et al. 2011].*

gaze data (see Figure 2(b)). The webcam was mounted to the eye tracker on a plastic frame attached to the head unit. The camera recorded images of the participant's left eye (see Figure 3(a)).

The experiment took place in a real office environment with fluorescent illumination. Participants were seated about 60cm away from a 23-inch LCD monitor (with visual angles of 43° horizontal and 27.6° vertical)(see Figure 3(b)). Free movements of the head and the upper body were allowed at any time but we encouraged the participants to move as little as possible during the experiment.

The visual stimulus consisted of a red dot with a radius of 20 pixels (i.e. 0.5° of visual angle) shown in front of a light grey background. The system guided each participant through a sequence of 13 different predefined gaze locations. Participants were asked to fixate on the red dot at each location for five seconds (see Figure 3(c)). After five seconds, the stimulus was shown at the next location. Thereafter, the participant was asked to follow a moving stimulus along several predefined paths. For each path, the stimulus moved horizontally, vertically and diagonally at constant speed. The entire procedure was performed three times. While the data was recorded, the system labeled the recorded images according to the stimulus's gaze point on the screen (see Figure 3(b)).

4.1 Analysis

We define the gaze direction by two rotation angles: Θ_h and Θ_v, for horizontal and vertical directions, respectively. The origin, $(\Theta_h, \Theta_v) = (0, 0)$, is the eye position when the gaze direction is perpendicular to the screen surface. Each direction of gaze (Θ_h, Θ_v) corresponds to only one gaze point $g = (x, y)$ on the screen. d denotes the distance of the users from the monitor. Given an estimation $g' = (x', y')$, the angular error was calculated by:

$$(\Delta\Theta_h, \Delta\Theta_v) = (|tan^{-1}(\frac{x - x'}{d})|, |tan^{-1}(\frac{y - y'}{d})|) \quad (1)$$

	Feature	Horizontal [°]	Vertical [°]
Individual	Color (C)	4.21 ± 0.56	2.41 ± 0.69
	Intensities (I)	4.03 ± 0.65	1.89 ± 0.62
	Orientation (O)	4.02 ± 0.65	2.04 ± 0.57
	Haar (H)	3.73 ± 0.56	1.52 ± 0.49
	Spatiogram (S)	5.48 ± 0.51	1.97 ± 0.53
Combined	All	3.44 ± 0.48	1.37 ± 0.40

Table 2: ***Person-dependent*** *mean and standard deviation of the gaze estimation angular error in horizontal and vertical direction averaged over the 17 participants for different feature types.*

	Feature	Horizontal [°]	Vertical [°]
Individual	Color (C)	11.29 ± 3.10	9.11 ± 1.59
	Intensities (I)	10.59 ± 2.71	8.26 ± 1.44
	Orientation (O)	10.59 ± 2.47	8.95 ± 1.03
	Haar (H)	15.37 ± 3.86	11.21 ± 1.77
	Spatiogram (S)	15.66 ± 1.82	9.17 ± 2.39
Combined	All	13.89 ± 3.84	8.63 ± 2.58

Table 3: ***Person-independent*** *mean and standard deviation of the gaze estimation angular error in horizontal and vertical direction averaged over the 17 participants for different feature types.*

5 Experimental results

5.1 Person-dependent evaluation

Around 1900 images/gaze coordinates pairs were collected for each participant. We first evaluated our system using a person-dependent evaluation scheme. For each participant 70% of the images were randomly selected for training (the "training set"); the remaining 30% (i.e. the test set) were used for gaze estimation on the same participant. The random splits of training and testing data were conducted 5 times for each participant.

Table 2 shows the average results for the different image features (cf. Table 1). The errors were calculated under the assumption that the participant looked at the centre of the stimulus on the screen. Using person-dependent evaluation the system achieved an average gaze estimation angular error of 3.44° horizontally and 1.37° vertically (1° corresponds to about $1.1cm$ on the screen plane). Figure 4 illustrates the angular errors for each participant using the five types of image features individually and all combined.

5.2 Person-independent evaluation

To test the algorithm's robustness across different people we further performed a leave-one-person-out cross-validation. In this scheme, gaze data of 16 participants was used for training the regression neural network and the data of the remaining person was used for testing. This was performed repeatedly over all 17 participants. The resulting gaze estimation performance averaged across all iterations and participants are summarized in Table 3.

6 Discussion

Our data set includes outliers (e.g. image blurring and blinks), noises (e.g. reflection from bright objects, such as the monitor), as well as human errors (e.g. a participant failed to follow the stimulus). Despite these challenges our results show that estimating gaze with a small set of low-level image features from webcam images

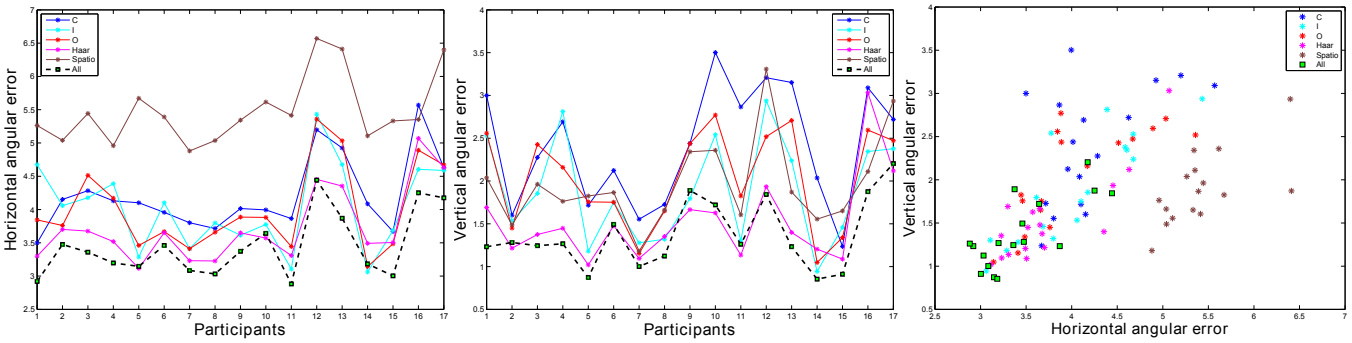


Figure 4: *Gaze estimation angular errors of 17 participants in horizontal (4(a)) and vertical (4(b)) directions. The sold lines in (a) and (b) represent angular errors using the five types of features individually, while the dashed lines illustrate the errors when using all the features combined. The error distribution of each instance in both directions is illustrated in (c).*

is feasible. A survey by Hansen and Ji identified that the accuracy of existing systems using webcamera without any additional illumination is 2-4 ° [2010]. Our system achieved similar accuracy with a user-dependent setup. Furthermore, the angular error was influenced by squinting (occlusions by the eyelid) and frequent blinking. In our post-study analysis, we observed that participant 12 blinked frequently and participant 16 squinted his eyes (see Figure 4(a) and 4(b)). A blinking detection method can be developed to increase the robustness of the system. The stimulus spreads less in vertical direction (10.75 ° horizontally, 6.9 ° vertically) which result in less error vertically. This suggests that our system can be improved by using more spatially closed training points.

Our dataset covers a large variance of eye appearance from people with different ages, gender and races. Consequently, person-independent gaze estimation shows higher angular errors than person-dependent. Although eye appearance differs across people, by using machine learning, our method learned common features which are effective in estimating gaze. Without re-calibration our method is able to provide sufficient accuracy to distinguish different areas of the monitor.

Figure 4 shows that although several individual features achieve better accuracy, overall the best performance is achieved by combining all features. Among the individual features, color, intensities, and orientation perform similarly, while spatiogram performs worst in horizontal direction. Haar features achieve better performance in person-dependent than in person-independent evaluation. This suggests Haar features are sensitive to eye appearance variance. Overall, these results show that it is difficult to select a single best feature set for different users. We plan to investigate other features that are potentially robust to different image scales (varying distance from eye to camera) and lighting variance, as well as methods to optimally select features for different applications. In addition, further improvements include adopting advanced gaze regression methods.

7 Conclusion

In this paper we presented a novel appearance-based technique that uses a small set of low-level image features and machine learning for gaze estimation. Our technique has the benefits of no calibration, non-intrusiveness and adaptability to new users. Results from a 17-participant user study show that the technique is robust across users with diverse characteristics and achieve decent performance for discrete gaze estimation. These initial results are promising and open up interesting applications in pervasive eye tracking.

References

BALUJA, S., AND POMERLEAU, D. 1994. Non-intrusive gaze tracking using artificial neural networks. Technical report cmu-cs-94-102, Carnegie Mellon University.

BIRCHFIELD, S. T., AND RANGARAJAN, S. 2005. Spatiograms versus histograms for region-based tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1158–1163.

HANSEN, D., AND JI, Q. 2010. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 3, 478–500.

PENG, H., LONG, F., AND DING, C. 2005. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1226–1238.

SUGANO, Y., MATSUSHITA, Y., SATO, Y., AND KOIKE, H. 2008. An incremental learning method for unconstrained gaze estimation. In *European Conference on Computer Vision*, 656–667.

VERTEGAAL, R., SHELL, J. S., CHEN, D., AND MAMUJI, A. 2006. Designing for augmented attention: Towards a framework for attentive user interfaces. *Computers in Human Behavior 22*, 4, 771 – 789.

VIOLA, P., AND JONES, M. 2001. Rapid object detection using a boosted cascade of simple features. In *Proceedings of CVPR*, IEEE, 511–518.

WALTHER, D., AND KOCH, C. 2006. Modeling attention to salient proto-objects. *Neural Networks 19*, 9, 1395–1407.

WILLIAMS, O., BLAKE, A., AND CIPOLLA, R. 2006. Sparse and semi-supervised visual mapping with the sˆ 3gp. In *Proceedings of CVPR*, IEEE, 230–237.

XU, L., MACHIN, D., AND SHEPPARD, P. 1998. A novel approach to real-time non-intrusive gaze finding. In *British Machine Vision Conference*, 428–437.

ZHANG, Y., BULLING, A., AND GELLERSEN, H. 2011. Discrimination of gaze directions using low-level eye image features. In *Proceedings of PETMEI*, ACM, 9–14.

ZHU, Z., AND JI, Q. 2005. Eye gaze tracking under natural head movements. In *Proceedings of CVPR*, IEEE, 918–923.