Person-independent eye gaze prediction from eye images using patch-based features

Feng Lu [a,b], Xiaowu Chen [c,*]

[a] School of Computer Science and Engineering, Beihang University, Beijing 100191, China
[b] International Research Institute for Multidisciplinary Science, Beihang University, Beijing 100191, China
[c] State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China

Keywords:
Gaze estimation
Eye image
Sparse auto-encoder
Gaze direction classification

A B S T R A C T

This paper delivers a preliminary attempt towards person-independent appearance-based gaze estimation. Conventional methods need to assume training and test data collected from the same person, otherwise eye shape difference due to individuality will affect the estimation severely. To solve this problem, the key idea in this paper is to extract from eye images more advanced eye features, which helps learn a person-independent relationship between eye gaze change and eye appearance variation. To this end, we propose employing the advantages of recent sparse auto-encoding techniques. We partition any eye image into small patches which can overlap with each other. With patches from many images, we learn a codebook comprising a set of bases, which can reconstruct any eye image patch with sparse coefficients. By examining these coefficients, we can analyze the eye shape more effectively. Finally, we produce the eye features by pooling the coefficients at different scales, and then combine these subfeatures from different codebooks. Experimental results show that the proposed method achieves good accuracy on a public dataset and it also outperforms conventional methods by a large margin.

1. Introduction

Human gaze tracking has been long considered as the next generation solution for human computer interaction (HCI) [1]. It is supposed to be much more powerful than any other conventional interfaces. The reason is that more than half of human sensory information is received by the eyes and processed by the human visual systems in the brain. Therefore, eye gaze movements can be used as an essential cue to directly reflect human attention, feeling, intention and other internal status [2,3], which are important for the design of an effective interaction interface for applications including virtual reality (VR), HCI and many other promising systems [4–6].

However, in the field of computer vision, existing gaze estimation techniques still face various difficulties and limitations. The so-called model-based methods, which occupy the majority of existing solutions, require to extract small eye features in the eye image. To this end, they usually need special devices such as infrared lights and multiple cameras. On the other hand, another type of methods, namely appearance-based methods, can use the entire eye image as input to learn a simple regression. They can work with low resolution images without active illuminations. However, training is necessary every time before use. In fact, the appearance-based methods only accept training data and test data from the same person. Otherwise the eye shape change due to individuality will greatly affect the accuracy.

In this paper, we propose a method that belongs to the category of appearance-based methods. Our focus is to develop a technique that can use training data from other persons when doing gaze estimation for the current person. In this sense, training can be done in advance without the participation for the current user. In other words, a user can use a pre-trained system directly without performing troublesome training every time. This makes a huge difference compared to conventional appearance-based methods.

To solve this problem, our idea is to propose a better eye feature that is less person-dependent during training and test. In particular, inspired by the recent developments on deep network and various autoencoders for image feature learning [7,8], we partition eye images into patches and use a sparse auto-encoder to learn a set of bases from randomly collected patches. When we use the learnt bases to reconstruct any original image patch, the resulting coefficients only have sparsely distributed non-zero values. In other words, every non-zero coefficient plays an important role in carrying essential structural information of the

* Corresponding author.
E-mail addresses: lufeng@ut-vision.org (F. Lu), chen@buaa.edu.cn (X. Chen).

original image patch. Therefore, the obtained coefficients have a good potential to be used as eye features in our problem. Furthermore, we propose to obtain coefficients by using multiple codebooks, and use spatial pyramid pooling to extract final features from the coefficients at different spatial layers. The resulting feature vector is able to handle factors such as scaling, translation and even shape deformation well, and thus it greatly improves the gaze estimation accuracy without training by the same person.

Overall, this paper makes the following contributions: (1) computation of patch-based sparse representation of the eye appearance (Section 3); (2) final eye feature by using multiple codebooks and spatial pyramid pooling (Section 4.1); (3) modeling and solving the problem by using the proposed eye feature (Section 4.2). Experimental evaluations in Section 5 demonstrate the advantage of the proposed method.

2. Related works

There have been many computer vision-based gaze estimation techniques proposed recently. According to recent surveys [9,10], most existing methods belong to either of the two major categories, namely the model-based methods and appearance-based methods.

The model-based methods assume some kinds of eyeball models, based on which eye gaze directions can be computed geometrically. Such models can be either 2D or 3D [11,12], or can be other models that involve environment configurations, e.g., cross ratio models [13,14]. In order to obtain the key parameters of the model, small features on the eyeball surface are extracted from eye images. The most commonly used features include near infrared (NIR) corneal reflections, pupil reflections [13,11,15] and iris contours [16,17]. To produce the NIR reflection points on the eyeball surface, active NIR illumination is usually assumed, together with infrared cameras with long focal lengths and multiview cameras for eyeball position tracking. Besides, positions of lights and cameras are always calibrated beforehand [18–20].

Recently, several methods have been proposed leveraging the depth information from depth sensors, e.g., Kinect [21–23]. The depth information is very useful in tracking user's head pose and locating the eye region. However, a depth sensor is also considered to be additional hardware with active illuminations and capturing systems. Similar to conventional model-based methods, they are also less practical in some common scenarios where only a mobile phone or tablet is being used.

Most appearance-based methods do not extract small eye features. Instead, they use all pixel values from an entire eye image as a high dimensional feature vector, and learn a mapping between eye features and real gaze positions through training. Such a mapping can be learnt via different techniques. For instance, early systems used neural networks to learn the mapping [24,25]. However, due to the large number of unknown weights in the network, thousands of training samples were needed to refine the neurons' connections. Later, linear regression became widely used. Tan et al. [26] proposed a simple method that interpolated the unknown gaze sample by using its nearest neighbors. This method exploits the local similarity of data in the eye appearance manifold and reduces the number of training samples to several hundreds. To further reduce the training cost, Williams et al. [27] proposed a semi-supervised method that could use both labeled and unlabeled samples to train a Gaussian Process Regressor. Lu et al. [28,29] followed the idea of linear regression and introduced an adaptive regression method to use much sparsely collected training samples. Their method can also handle problems such as alignment and eye blink in gaze estimation. Sugano et al. [30] proposed a calibration-free method by assuming that a user is watching a video. Visual saliency information from the video can then guide the gaze position prediction. However, when free head motion is considered, most existing methods need to increase their training samples' number [31–34] again.

Another common limitation shared by existing appearance-based methods is that they all assume that training data and test data are collected during the same session, i.e., from the same user. Therefore, a training stage becomes a must before a user can use the system. In contrast, if a system can be trained by someone else beforehand, while it can still work well for a new user, its applicability will be dramatically improved. Delivering techniques towards such a goal is the main purpose of this work.

3. Patch-based eye feature extraction using sparse auto-encoder

In this section, we propose to obtain patch-based features from eye images using sparse auto-encoder. The features are coefficients from sparse representations of images, and they carry important structural information of the eye shape more effectively than original pixel values.

3.1. Patch-based eye image codebook

Assume we are given a set of eye images $\boldsymbol{X}=[\mathbf{x}_1,\ldots,\mathbf{x}_n,\ldots,\mathbf{x}_N]$, where each $\mathbf{x}_n$ is a column vector stacking all pixel values from one image. We aim at finding a codebook comprising bases $\boldsymbol{B}=[\mathbf{b}_1,\ldots,\mathbf{b}_m,\ldots,\mathbf{b}_M]$ that are able to represent each image by

$$\mathbf{x}_n=\sum_{m=1}^{M} c_{m,n}\cdot\mathbf{b}_m. \tag{1}$$

We can write this in a matrix form by

$$\boldsymbol{X}=\boldsymbol{B}\boldsymbol{C}. \tag{2}$$

Given images $\boldsymbol{X}$ as observations, we can optimize the codebook $\boldsymbol{B}$ by solving

$$\boldsymbol{B}=\underset{\boldsymbol{B}}{\operatorname{argmin}}\|\boldsymbol{B}\boldsymbol{C}-\boldsymbol{X}\|^2, \tag{3}$$

where $\boldsymbol{C}=[\mathbf{c}_1,\ldots,\mathbf{c}_n,\ldots,\mathbf{c}_N]$ and $\mathbf{c}_n=[\ldots,c_{m,n},\ldots]^{\mathrm{T}}$.

Because Eq. (3) applies a matrix decomposition without any regular term, the solution will not be unique. A randomly obtained codebook $\boldsymbol{B}$ cannot be optimal in terms of representing $\boldsymbol{X}$ efficiently. On the other hand, the sparse auto-encoder theory [37] claims that one can assume the sparsity in the coefficients $\boldsymbol{C}$, which leads to the following problem:

$$\begin{aligned}&\boldsymbol{B}=\arg\min_{\boldsymbol{B}}\|\boldsymbol{B}\boldsymbol{C}-\boldsymbol{X}\|^2+\lambda\sum_{n=1}^{N}\|\mathbf{c}_n\|_1,\\ &\text{s.t.}\quad \|\mathbf{b}_m\|<T,\quad \forall m=1,\ldots,M.\end{aligned} \tag{4}$$

Note that in Eq. (4), we use an L_1 penalty in the second term. This is because for reconstructing each image, we expect as few as possible bases are activated, i.e., with non-zero coefficients. In this sense, each of the learnt basis has a maximum ability to represent an input image. Also note that we assume any basis in $\boldsymbol{B}$ has a norm that is no larger than a threshold T. This is because if the elements in $\boldsymbol{B}$ can grow freely, $\boldsymbol{C}$ will in return become very small while Eq. (4) can still hold true. This will make the L_1 penalty less influential and thus the sparsity can no longer be ensured.

The above method can learn a codebook for eye images. However, in practice, it is more effective to learn the image patches rather than the entire image. Therefore, we randomly crop image patches with a certain size from many eye images. These patches constitute the matrix $\boldsymbol{X}$ and from which we can learn a set

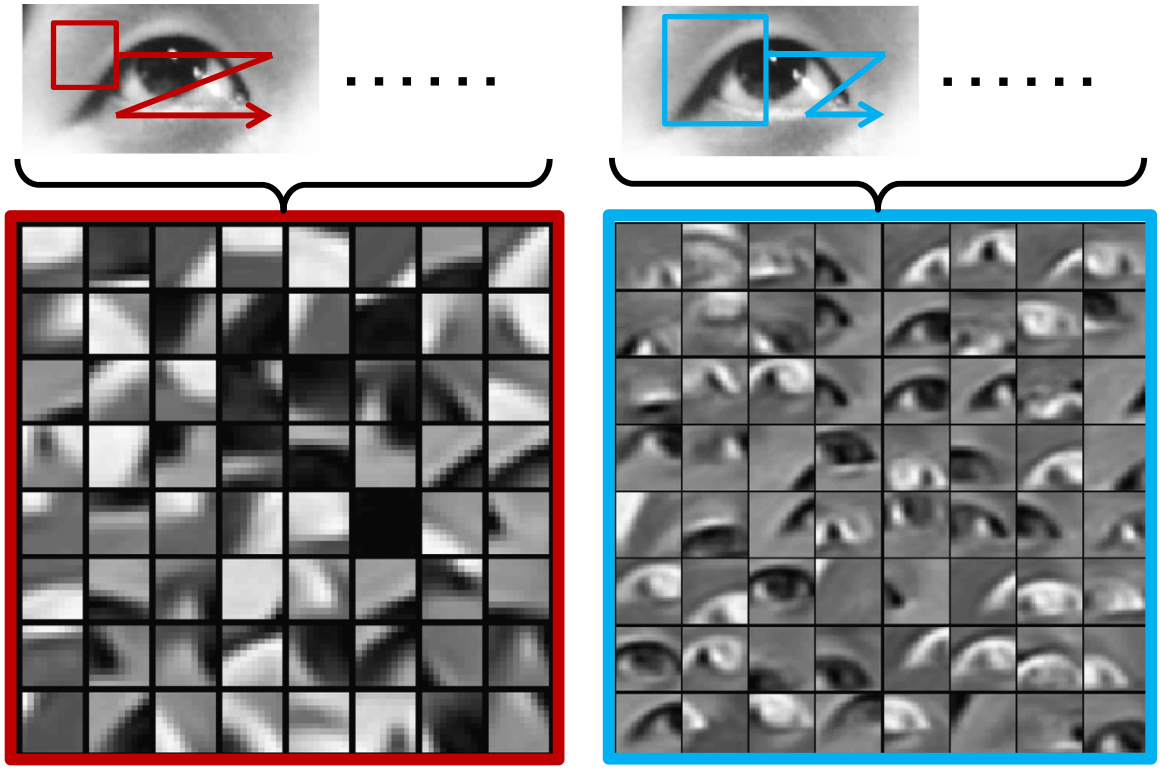

Fig. 1. Eye image codebooks are optimized by using image patches extracted from different locations of a set of eye images. With small and large patch sizes, the sparse auto-encoder learns two types of codebooks.

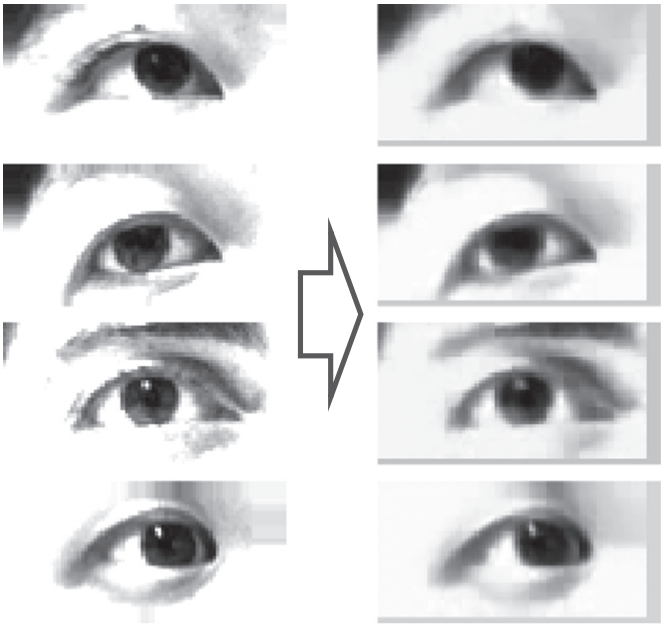

Fig. 2. Patch-based eye image reconstruction by using the same codebook. The reconstructions (right) approximate the real images (left).

of bases by solving Eq. (4). Here we use a fast sparse coding solver provided by Lee et al. [38] to solve Eq. (4) effectively.

Fig. 1 shows two typical codebooks learnt from eye image patches with two different patch sizes. When learnt by using small patches, the bases reveal local structure of the eye image; while larger bases show more global eye structure. These learnt base patches have strong ability to reconstruct any eye image from different persons with different gaze directions.

3.2. Eye image feature encoding

In practice, we first partition an entire eye image into patches $\{\mathbf{x}_n\}$, whose center positions in the image are $\{\mathbf{p}_n\}$ and the patch size is $d \times d$. Note that these patches are supposed to overlap with each other and thus any pixel may belong to more than one patch.

Then, given the learnt codebook $\boldsymbol{B}$, for an arbitrary eye image patch $\mathbf{x}_n$, we can compute a coefficient vector $\mathbf{c}_n$ that reconstructs $\mathbf{x}_n$. In particular, we solve the following problem:

$$\mathbf{c}_n = \arg\min_{\mathbf{c}_n} \|\boldsymbol{B}\mathbf{c}_n - \mathbf{x}_n\|^2 + \lambda \|\mathbf{c}_n\|_1, \tag{5}$$

where $\mathbf{c}_n$ can be used to reconstruct $\mathbf{x}_n$ by $\boldsymbol{B}\mathbf{c}_n$. Eq. (5) can be solved by using the feature-sign search algorithm proposed by Lee et al. [38].

After computing the coefficients for every patch, we can reconstruct all the patches and assemble them to reproduce the original eye image. Because different patches may overlap with each other, we keep a record at any pixel all pixel values from different patches, and then take their median as the final pixel value. For instance, we compute any pixel value v_i at position i as follows:

$$v_i = \text{median}(\{v^{(n)}\}),$$
$$\text{s.t.} \quad v^{(n)} \in \mathbf{x}_n, \quad i \in \text{Neighbor}_{d\times d}(\mathbf{p}_n), \tag{6}$$

where $\text{Neighor}_{d\times d}(\mathbf{p}_n)$ defines a $d \times d$ neighboring region centered by $\mathbf{p}_n$.

Examples of reconstructed eye images using the same patch-based codebook are shown in Fig. 2. From these results, we can make two observations. First, by using the same codebook, we are able to reconstruct eye images for different persons and different gaze directions. This demonstrates the effectiveness of eye image representation by using a single codebook. Second, the reconstructed eye images accurately recover the structure of original images, while they are more smooth and statistically noise free. In other words, the coding coefficients $\{\mathbf{c}_n\}$ are suitable to use as eye image features that carry the most essential information of the original images.

3.3. Eye image pre-processing

When capturing eye images in real scenes, factors such as ambient illuminations and camera parameters also affect the pixel values. However, since our patch-based image coding technique aims at capturing the intrinsic eye property, effects of these additional factors should be suppressed. Therefore, we propose to apply eye image pre-processing beforehand to rectify the originally captured images.

In particular, given an arbitrary eye image with pixel values $\{I_i\}$, we first edit its brightness to remove the detail in the bright regions and make the pupil region more distinguishable. This is because the most important information about gaze direction is carried by the pupil region, which is dark. This procedure is done by

$$I'_i = \begin{cases} I_i, & \text{if } I_i < \text{median}(\{I_i\}), \\ \text{median}(\{I_i\}), & \text{otherwise}, \end{cases} \tag{7}$$

and then we remap all pixel values to have zero mean and standard deviation of one by

$$I''_i = I'_i / \text{std}(\{I_i\}) - \text{mean}(\{I'_i / \text{std}(\{I_i\})\}), \tag{8}$$

where I''_i is the pixel value at pixel i after processing.

One example of the proposed eye image processing is illustrated in Fig. 3. The processed eye images are then used to produce patches, learn codebooks or compute coefficients as described in previous sections.

4. Eye gaze prediction

In the previous section, we describe how to represent any eye image by a set of coefficients $\{\mathbf{c}_n\}$ given a codebook $\boldsymbol{B}$ learnt from image-patches. In this section, we propose a method to extract more advanced eye image features from the coefficients, and use them for person-independent gaze direction prediction.

4.1. Final feature production

Although one can put all coefficients $\{\mathbf{c}_n\}$ from an eye image into a single feature vector, the resulting high dimensionality is usually unexpected. Moreover, we would prefer a more advanced and effective feature of the image, which keeps the most important information while also tolerates effects such as scaling, translation and even shape deformation due to individuality to some degree. This is essential for our method to work for different persons with different eye shapes.

Recent development on spatial pyramid pooling [35, 39] suggests a very promising solution. The key idea is to partition the image into divisions in a coarse to fine manner. In each layer, local features, i.e., coefficients in our case, are aggregated inside each division. All outputs in different layers are then concatenated to produce the final feature. In this manner, features are extracted at variant scales and thus both local and global information can be obtained.

We propose our feature production following the same idea. The general framework is shown in Fig. 4, where we use a similar spatial pyramid pooling strategy; furthermore, we propose to extract features by using two codebooks, which correspond to small and large eye image patches, and combine them together to form a final feature vector. This further improves our feature's ability to handle multi-scale image information.

Without loss of generality, consider the codebook $\boldsymbol{B}^S$, where 'S' stands for 'small patch', assume we have obtained coefficients $\{\mathbf{c}_n\}$ from a set of patches $\mathbf{x}_n$. Then, we produce a feature vector by using Algorithm1.

Algorithm 1. Feature vector from spatial pyramid pooling.

Input: Coefficients $\{\mathbf{c}_n\}$ for patches $\mathbf{x}_n$.
for d-th spatial layer **do**
  • Divide the image into W_d uniform regions
  for w-th region **do**
    • $\mathbf{f}_{d,w} =$ take elementary-wise max of all $\{\mathbf{c}_n\}$ inside this region
  end for
end for
• Concatenate all $\mathbf{f}_{d,w}$ to form $\mathbf{f}^S$
• **Output:** feature vector $\mathbf{f}^S$

In practice, the number of regions W_d is computed by

$$W_d = 4^{d-1}, \quad d = 1, 2, \ldots, D, \tag{9}$$

where D is the number of layers. Any configurations other than Eq. (9) are also valid. By using Algorithm 1, we can produce a feature vector $\mathbf{f}^S$ which corresponds to the codebook $\boldsymbol{B}^S$ with a small patch size; in a similar manner, $\mathbf{f}^L$ can be obtained with codebook $\boldsymbol{B}^L$. The final feature vector $\mathbf{f}$ of the eye image is then produced by $\mathbf{f} = [\mathbf{f}^S; \mathbf{f}^L]$.

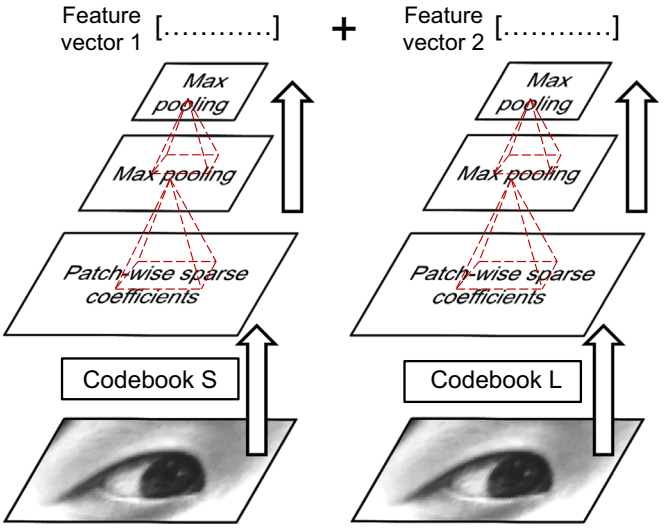

Fig. 4. Eye image feature extraction. Sparse coefficients are first solved given one or more codebooks. They are then pooled over different spatial scales as shown in the spatial pyramid structure. The final values are concatenated to form the output feature.

Fig. 3. Eye image preprocessing. Image brightness is first modified and then all pixel values are remapped to rectify their mean and standard deviation.

4.2. Gaze prediction by classification

We model the gaze prediction problem as a classification problem. In particular, we define a set of labels $\{1,\ldots,K\}$, which correspond to a set of anchor gaze directions $\{\mathbf{g}_k \in \mathbb{R}^3\}$ in the 3D space. These $\{\mathbf{g}_k\}$ can be chosen by using various strategies, and in most cases a uniform sampling is favorable.

As a result, the label y for any given gaze direction $\mathbf{g}$ can be computed by

$$y = k = \arg\min_{k \in \{1,\ldots,K\}} \|\mathbf{g}_k - \mathbf{g}\|^2. \tag{10}$$

In this sense, we choose to predict the class label y instead of the exact gaze direction $\mathbf{g}$ when given an eye image feature $\mathbf{f}$ as input. For this purpose, we implement a multi-class linear SVMs. In particular, given a set of training data $\{(\mathbf{f}_j, y_j)\}$ collected from different persons, we learn K linear transformations $\{\mathbf{w}_k^{\mathrm{T}}\mathbf{f}\}$, which predict the class label y from feature $\mathbf{f}$ by

$$y = \max_{k \in \{1,\ldots,K\}} \mathbf{w}_k^{\mathrm{T}}\mathbf{f}. \tag{11}$$

Each of the $\mathbf{w}_k^{\mathrm{T}}$ is trained separately by using the training data. The training is done by solving the following problem:

$$\mathbf{w}_k = \arg\min_{\mathbf{w}_k} \left(\|\mathbf{w}_k\|^2 + \mu \sum_j h(\mathbf{w}_k, \mathrm{equal}(y_j, k), \mathbf{f}_j) \right), \tag{12}$$

where functions $\mathrm{equal}(\alpha,\beta)$ and $h(\alpha,\beta,\gamma)$ are defined by

$$\mathrm{equal}(\alpha,\beta) = \begin{cases} +1, & \text{if } \alpha = \beta, \\ -1, & \text{otherwise}, \end{cases} \tag{13}$$

and

$$h(\alpha,\beta,\gamma) = \max(0, \alpha^{\mathrm{T}} \cdot \gamma \cdot \beta - 1)^2. \tag{14}$$

Finally, the learnt linear functions $\{\mathbf{w}_k^{\mathrm{T}}\mathbf{f}\}$ can be used to predict the label y from $\mathbf{f}$ by using Eq. (11), and the resulting y indicates which gaze direction in $\{\mathbf{g}_k\}$ is most similar to the unknown gaze direction corresponding to $\mathbf{f}$. This result is used as the gaze prediction output.

5. Experimental evaluations

5.1. Experimental setup

For experiments, we use the gaze dataset proposed in [36], from which we obtain eye images and 3D gaze directions for 15 different subjects. These gaze data are captured in a common user-screen scenario, and we use the eye images captured by a near frontal camera which is a common setting in practice. In total, there are 15×160 (gaze directions)$=2400$ samples in our dataset.

As described in Section 4.2, we use gaze direction labels instead of the exact 3D gaze directions in our experiments. In particular, as shown in Fig. 5, we observe that the gaze direction ranges in the dataset are 30° in the vertical direction and 50° in the horizontal direction. Therefore, we partition the gaze direction space into $5 \times 8 = 40$ uniform subregions, each of which is of an approximately $6^\circ \times 6^\circ$ size. Note that all gaze directions in the dataset are almost uniformly distributed in the 40 classes.

The 15 different subjects have different eye shapes, some of which are very unique. Therefore, these eye images with different shapes can be used to test our method for its ability in person-independent gaze computing.

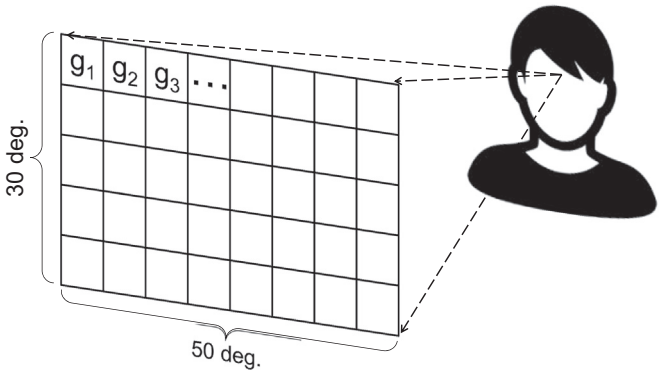

Fig. 5. All 3D gaze directions in the dataset are categorized into $8 \times 5 = 40$ classes, which are evenly distributed in the space.

5.2. Codebook learning and eye image reconstruction

We learn codebooks from eye image patches under different configurations. In particular, we set image patch size to be 10×10 and also 30×30, and set the sampling interval to be 3 and 6 pixels, respectively, given that a typical eye image size is 80×40.[1] We also learn different codebooks with different basis numbers, i.e., 25, 64, 100, 150 and 256.

Examples of learnt codebooks with different number of bases are shown in Fig. 6, where we use 'S25' to indicate 25 bases learnt from 'Small' image patches and use 'L64' to indicate 64 bases learnt from 'Large' image patches, etc. Note that we do not show cases of 150 and 256 bases in Fig. 6 due to their large numbers. In fact, they appear quite similar to the '100' cases.

For codebook S25, it contains very basic components in the image, like bright/dark regions located in two parts of the patch. By increasing the number to 64 and 100, the visual style still remains, while the difference is that more spatial patterns are included. However, for codebook L25, since it works with a larger image patch size, the learnt structures are more complex, and the bases are kind of blurred. By increasing the number to 64 and 100, more eye shape structures are kept with different gaze directions and individual eye shapes.

Overall, the codebooks can learn different but essential image components in different locations and scale levels. When using them for eye image reconstruction, the ratio of non-zero coefficients solved by Eq. (5) varies from 5% (of totally 256 bases) to 20% (of totally 25 bases). More intuitively, we examine the PSNR (peak signal-noise-ratio) of the reconstructed eye image by using different codebooks. The PSNR is computed by

$$PSNR = 10 \times \log_{10} \frac{\max(image)}{\mathrm{MSE}(image1, image2)}, \tag{15}$$

where MSE means the mean square error and a larger PSNR indicates a more accurate reconstruction. The results are shown in Table 1, where we can see that the reconstruction accuracy increases with more bases in the codebook. Moreover, reconstructions using codebooks with a large patch size show a little bit better performance.

5.3. Gaze estimation by using different codebooks

We use the learnt codebooks to predict gaze directions (in our case, labels) from eye image features. The feature extraction and gaze estimation methods are described in Section 4.1 and 4.2. For experiments, we use a leave-one-out test strategy. In particular, for each subject, we use his/her gaze data as test data, and use data

[1] We have down-sampled the original eye image in the dataset.

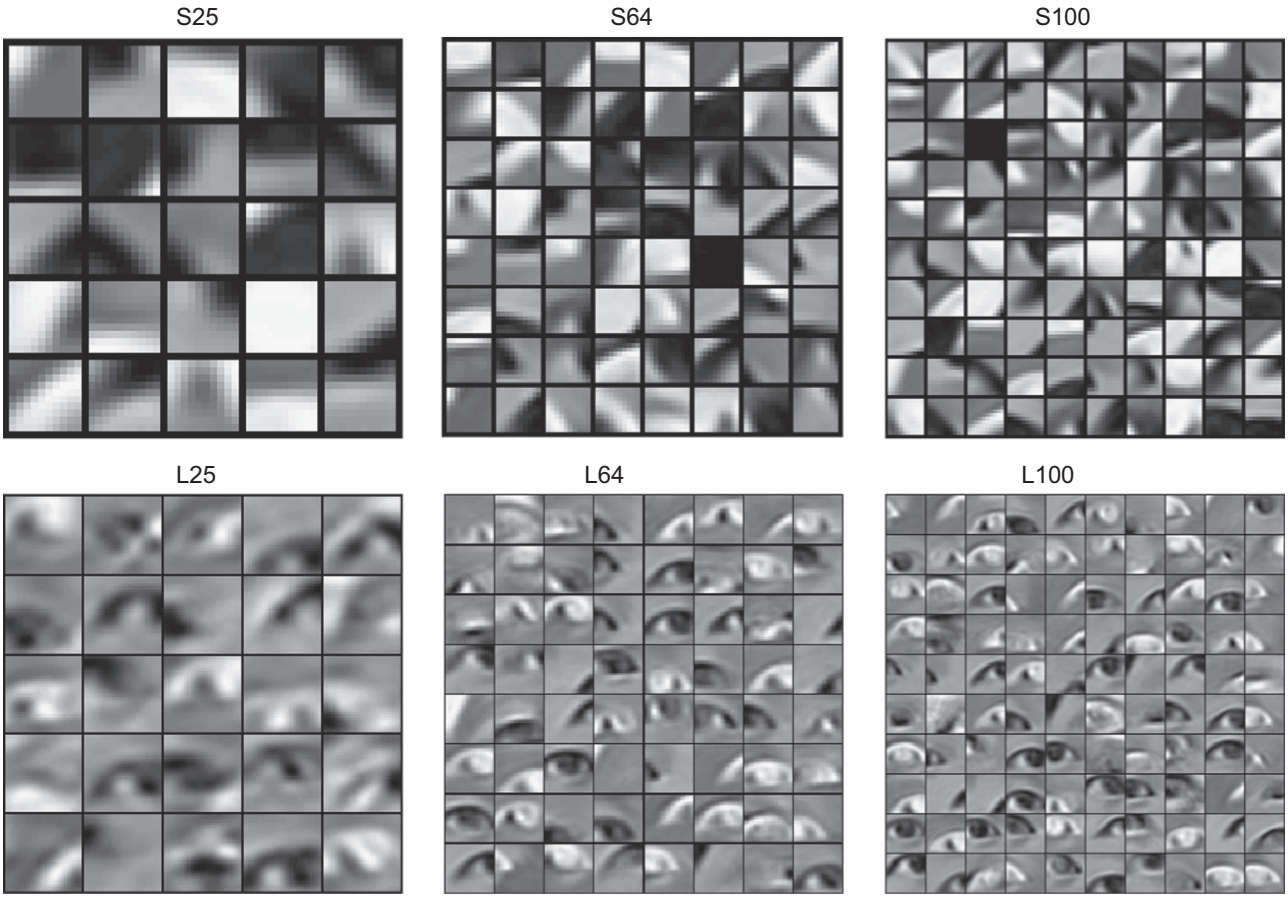

Fig. 6. Learnt eye image codebooks by using small patches (first row) and large patches (second row). They are learnt by setting the bases' number to be either 25, 64 or 100. Results with more bases (150 and 256) are not shown here because those numbers are too large for visualization.

Table 1
Average PSNRs of eye image reconstruction using differently learnt codebooks.

Codebook	S25	S64	S100	S150	S256
PSNR	22.32	23.40	23.73	24.16	24.50
Codebook	L25	L64	L100	L150	L256
PSNR	21.98	23.48	24.04	24.59	25.33

from all other subjects as training data. In this sense, the gaze estimation does not use the training data from the same subject, while the training data also contains sufficient samples.

We conduct experiments by using different codebooks learnt before. In particular, for small and large patch sizes, we have five codebooks for each of them containing different numbers of bases. Besides, we also test our method under S150+L64 and S256+L64 cases, where codebooks for both small and large patch sizes are used to produce eye features jointly as described in Section 4.1.

Detailed experimental results for every subject using every codebook are shown in Table 2 for classification accuracy and in Table 3 for angular gaze direction error. Based on all these results, we can make the following conclusions. First, when using a small patch size of 10×10, increasing the basis number of the codebook can improve the gaze estimation accuracy. Second, when using a large patch size of 30×30, a basis number of 64 or 100 already achieves the best estimation performance, while further increasing the basis number will lead to worse results. Third, by using the joint feature obtained from both small and large codebooks, we achieve the best estimation accuracy among all. All these results suggest that we should choose proper basis numbers for both small and large patch sizes and use the joint feature for gaze estimation. A more intuitive visualization of the average results is given in Fig. 7.

5.4. Comparison with other methods

We compare the proposed method with some conventional appearance-based methods on the same dataset. In particular, we implement a 'Pixel+SVM' method that differs from our method in that it uses raw pixel values as input features; we also implement a very commonly employed appearance-based method [26,34] by using linear interpolations of 15 most similar samples to the test sample. For these different methods, all experimental conditions are kept the same.

Comparisons on gaze classification accuracy and angular error are shown in Table 4 and Fig. 8, where we find that the proposed method outperforms the other two using any of the tested codebooks. In general, since we have 40 gaze direction classes, a random guess should result in a 2.5% average classification accuracy. Therefore, the two other methods achieve much better results than a random guess, while our method in addition nearly double their accuracy especially with the 'S50+L64' codebook. Here, a 20% accuracy of our method may still look not high; however, note that this does not account for other 'near accurate' estimates. In fact, our overall gaze angular error of 7.5° is very promising considering that it is obtained in a person-independent manner and it also outperforms other methods. This demonstrates the advantage of the proposed method in person-independent gaze estimation.

Table 2
Gaze classification accuracy using features produced by different codebooks.

Subject	S25 (%)	S64 (%)	S100 (%)	S150 (%)	S256 (%)	L25 (%)	L64 (%)	L100 (%)	L150 (%)	L256 (%)	S150+L64 (%)	S256+L64 (%)
Subj.1	18.1	14.4	24.4	28.7	20.0	20.0	24.4	28.1	19.4	21.9	37.5	33.8
Subj.2	3.1	4.4	3.8	3.1	3.1	3.8	3.8	1.9	3.1	6.3	5.0	3.1
Subj.3	5.0	4.4	6.9	6.3	8.1	7.5	5.6	6.9	6.9	6.9	8.1	10.0
Subj.4	18.8	11.3	15.6	19.4	14.4	10.6	9.4	11.3	10.6	11.9	15.0	13.8
Subj.5	11.3	18.1	24.4	23.8	20.6	16.3	21.9	20.0	22.5	20.6	25.6	21.9
Subj.6	7.5	8.1	11.3	11.3	10.0	18.1	17.5	21.9	20.0	20.6	17.5	13.1
Subj.7	12.5	15.6	15.6	20.0	13.8	16.3	22.5	18.1	23.1	16.9	22.5	18.1
Subj.8	21.3	31.3	20.0	20.6	25.0	32.5	25.6	27.5	26.3	29.4	30.0	27.5
Subj.9	13.8	18.8	15.0	17.5	13.8	21.3	26.3	32.5	36.9	25.6	24.4	21.3
Subj.10	17.5	27.5	25.6	28.1	33.1	23.1	29.4	28.1	28.1	32.5	35.6	38.1
Subj.11	18.8	21.3	16.3	15.6	19.4	21.9	25.6	26.9	23.8	20.6	21.3	23.8
Subj.12	12.5	18.1	17.5	15.0	18.8	7.5	11.9	8.8	12.5	15.6	18.1	18.1
Subj.13	12.5	12.5	12.5	12.5	17.5	11.3	8.1	6.9	7.5	4.4	6.9	11.9
Subj.14	15.6	19.4	25.6	18.1	24.4	11.3	10.0	10.0	10.6	13.8	15.6	20.0
Subj.15	15.6	18.1	16.3	16.9	21.9	21.9	24.4	28.1	25.0	15.6	23.8	27.5
Avg.	13.6	16.2	16.7	17.1	**17.6**	16.2	17.8	**18.5**	18.4	17.5	**20.5**	20.1

Table 3
Gaze direction error using features produced by different codebooks.

Subject	S25	S64	S100	S150	S256	L25	L64	L100	L150	L256	S150+L64	S256+L64
Subj.1	7.6°	8.8°	7.7°	6.1°	6.5°	6.9°	5.8°	5.4°	7.0°	6.0°	4.9°	5.1°
Subj.2	13.0°	12.5°	14.2°	13.1°	13.4°	14.8°	13.3°	17.2°	15.0°	14.4°	13.7°	14.1°
Subj.3	13.2°	13.9°	12.6°	11.9°	11.3°	10.1°	9.5°	9.4°	9.5°	9.7°	8.8°	9.2°
Subj.4	8.3°	9.9°	9.2°	8.2°	9.0°	9.6°	10.6°	10.1°	10.7°	10.1°	9.6°	9.7°
Subj.5	9.7°	7.2°	6.4°	6.4°	6.6°	7.9°	6.7°	6.7°	7.3°	7.1°	6.0°	6.1°
Subj.6	11.3°	11.5°	10.6°	9.4°	9.4°	6.6°	6.2°	6.0°	5.9°	6.8°	7.7°	7.9°
Subj.7	8.2°	7.5°	7.0°	6.9°	7.1°	7.6°	6.3°	6.5°	5.9°	7.5°	6.5°	6.7°
Subj.8	7.4°	5.9°	6.6°	6.9°	6.1°	5.1°	5.6°	5.8°	6.1°	6.1°	5.6°	5.7°
Subj.9	8.3°	6.9°	7.3°	8.1°	7.4°	6.2°	5.6°	5.3°	4.6°	5.4°	5.8°	6.2°
Subj.10	6.8°	5.9°	5.7°	5.6°	5.3°	6.6°	5.8°	5.8°	5.7°	5.5°	4.7°	4.6°
Subj.11	7.4°	6.4°	7.4°	6.7°	6.9°	6.9°	5.6°	5.3°	6.0°	6.5°	5.9°	6.1°
Subj.12	8.2°	6.6°	6.5°	6.6°	5.8°	9.4°	8.3°	10.8°	10.7°	8.8°	7.3°	6.8°
Subj.13	10.0°	9.8°	8.5°	7.9°	7.9°	13.0°	11.9°	12.2°	11.2°	12.8°	11.5°	10.6°
Subj.14	13.0°	7.5°	7.0°	6.5°	7.0°	13.3°	10.9°	8.2°	8.8°	9.2°	7.2°	7.2°
Subj.15	8.2°	7.8°	8.4°	7.8°	7.1°	7.1°	6.5°	6.5°	6.6°	6.5°	6.7°	6.1°
Avg.	9.4°	8.5°	8.3°	7.9°	**7.8°**	8.7°	**7.9°**	8.1°	8.1°	8.2°	**7.5°**	**7.5°**

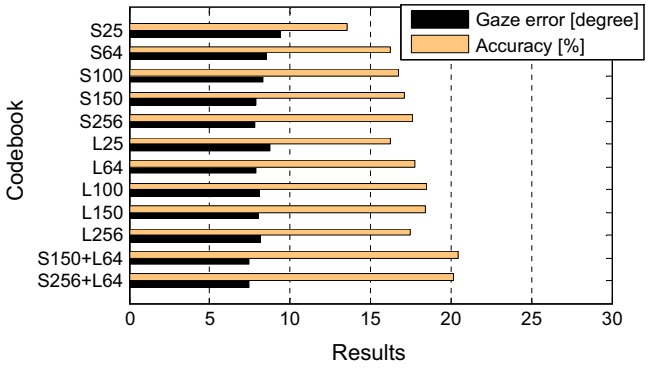

Fig. 7. Gaze estimation results (classification accuracy and gaze angular error) by using different codebooks.

6. Conclusions and future work

This paper aims at person-independent appearance-based gaze estimation. Conventional methods face difficulty when training and test samples are collected from different persons. Our idea to solve this problem is to extract from eye images more advanced eye features, which can work for different persons more effectively. To this end, we employ a sparse auto-encoder to learn a set of bases from eye image patches. By reconstructing any eye image

Table 4
Comparison of average gaze estimation results of all 15 subjects using different methods.

Criterion	S256	L64	S150+L64	Pixel+SVM	App-based [26,34]
Accuracy (%)	17.6	17.8	**20.5**	12.8	11.1
Gaze error	7.8°	8.1°	**7.5°**	10.1°	9.7°

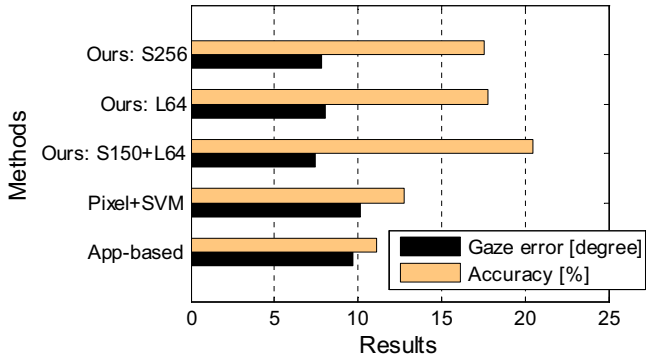

Fig. 8. Gaze estimation results (classification accuracy and gaze angular error) by using different methods. The proposed method achieves higher classification accuracy and smaller gaze error.

with these bases, the resulting sparse coefficients provide essential structural information of the eye image. We use spatial pyramid pooling to produce the final eye features, and we also combine features from codebooks with small and large patch sizes. Experimental results show that our method outperforms previous methods on a public dataset.

For future work, we suggest to further improve the design of the eye feature to overcome remaining limitations. For instance, there are some bases in the codebook that are more closely related to the important regions, e.g., iris contour and eyelids. It will be interesting to see how coefficients from such bases can be used more effectively. Besides, head motion is not explicitly considered in this paper while we plan to extend our work to support it.

Acknowledgements

This work was partially supported by the Joint Funds of NSFC-CARFC (U1533129), NSFC (61325011,61421003), SRFDP (20131102130002), and Lenovo Outstanding Young Scientists Program.

References

[1] F. Velasco-Álvarez, R. Ron-Angevin, L. da Silva-Sauer, S. Sancha-Ros, Audio-cued motor imagery-based brain–computer interface: navigation through virtual and real environments, Neurocomputing 121 (2013) 89–98.
[2] Y.-M. Jang, R. Mallipeddi, S. Lee, H.-W. Kwak, M. Lee, Human intention recognition based on eyeball movement pattern and pupil size variation, Neurocomputing 128 (2014) 421–432.
[3] A. Olmedo-Payá, A. Martínez-Álvarez, S. Cuenca-Asensi, J.M. Ferrández, E. Fernández, Modeling the role of fixational eye movements in real-world scenes, Neurocomputing 151 (2015) 78–84.
[4] Q. Cheng, D. Agrafiotis, A. Achim, D. Bull, Gaze location prediction for broadcast football video, IEEE Trans. Image Process. 22 (12) (2013) 4918–4929.
[5] Y. Yang, X. Wang, T. Guan, J. Shen, L. Yu, A multi-dimensional image quality prediction model for user-generated images in social networks, Inf. Sci. 281 (2014) 601–610.
[6] Y. Yang, X. Wang, Q. Liu, M. Xu, W. Wu, User models of subjective image quality assessment on virtual viewpoint in free-viewpoint video system, Multimed. Tools Appl. (2014) 1–21.
[7] A. Coates, A.Y. Ng, H. Lee, An analysis of single-layer networks in unsupervised feature learning, In: International Conference on Artificial Intelligence and Statistics, 2011, pp. 215–223.
[8] J. Masci, U. Meier, D. Cireşan, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, in: Artificial Neural Networks and Machine Learning—ICANN 2011, Springer, Espoo, Finland, 2011, pp. 52–59.
[9] D. Hansen, Q. Ji, In the eye of the beholder: a survey of models for eyes and gaze, IEEE Trans. Pattern Anal. Mach. Intell. 32 (3) (2010) 478–500.
[10] C. Morimoto, M. Mimica, Eye gaze tracking techniques for interactive applications, Comput. Vis. Image Underst. 98 (1) (2005) 4–24.
[11] E. Guestrin, M. Eizenman, General theory of remote gaze estimation using the pupil center and corneal reflections, IEEE Trans. Biomed. Eng. 53 (6) (2006) 1124–1133.
[12] J. Chen, Q. Ji, A probabilistic approach to online eye gaze tracking without explicit personal calibration, IEEE Trans. Image Process. 24 (3) (2015) 1076–1086.
[13] D.H. Yoo, M.J. Chung, A novel non-intrusive eye gaze estimation using cross-ratio under large head motion, Comput. Vis. Image Underst. 98 (1) (2005) 25–51.
[14] F.L. Coutinho, C.H. Morimoto, Improving head movement tolerance of cross-ratio based eye trackers, Int. J. Comput. Vis. (2012) 1–23.
[15] A. Nakazawa, C. Nitschke, Point of gaze estimation through corneal surface reflection in an active illumination environment, in: ECCV, 2012, pp. 159–172.
[16] J. Wang, E. Sung, R. Venkateswarlu, Eye gaze estimation from a single image of one eye, In: ICCV, 2003, pp. 136–143.
[17] M.J. Reale, S. Canavan, L. Yin, K. Hu, T. Hung, A multi-gesture interaction system using a 3-d iris disk model for gaze estimation and an active appearance model for 3-d hand pointing, IEEE Trans. Multimed. 13 (3) (2011) 474–486.
[18] D. Beymer, M. Flickner, Eye gaze tracking using an active stereo head, In: CVPR, 2003, pp. 451–458.
[19] X. Brolly, J. Mulligan, Implicit calibration of a remote gaze tracker, In: CVPRW, 2004, p. 134.
[20] J. Chen, Q. Ji, Probabilistic gaze estimation without active personal calibration, in: CVPR, 2011, pp. 609–616.
[21] J. Li, S. Li, Eye-model-based gaze estimation by rgb-d camera, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2014, pp. 606–610.
[22] K. Funes Mora, J.-M. Odobez, Geometric generative gaze estimation (g3e) for remote rgb-d cameras, In: CVPR, 2014, pp. 1773–1780.
[23] X. Xiong, Q. Cai, Z. Liu, Z. Zhang, Eye gaze tracking using an rgbd camera: a comparison with an rgb solution, In: The 4th International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction (PETMEI 2014), 2014.
[24] S. Baluja, D. Pomerleau, Non-intrusive gaze tracking using artificial neural networks, In: NIPS, 1994, pp. 753–760.
[25] L.Q. Xu, D. Machin, P. Sheppard, A novel approach to real-time non-intrusive gaze finding, In: BMVC, 1998, pp. 428–437.
[26] K. Tan, D. Kriegman, N. Ahuja, Appearance-based eye gaze estimation, In: WACV, 2002, pp. 191–195.
[27] O. Williams, A. Blake, R. Cipolla, Sparse and semisupervised visual mapping with the S^3GP, In: CVPR, 2006, pp. 230–237.
[28] F. Lu, Y. Sugano, T. Okabe, Y. Sato, Inferring human gaze from appearance via adaptive linear regression, in: ICCV, 2011.
[29] F. Lu, Y. Sugano, T. Okabe, Y. Sato, Adaptive linear regression for appearance-based gaze estimation, IEEE Trans. Pattern Anal. Mach. Intell. 36 (10) (2014) 2033–2046.
[30] Y. Sugano, Y. Matsushita, Y. Sato, Appearance-based gaze estimation using visual saliency, IEEE Trans. Pattern Anal. Mach. Intell. 99 (10) (2012) 1.
[31] Y. Sugano, Y. Matsushita, Y. Sato, H. Koike, An incremental learning method for unconstrained gaze estimation, In: ECCV, 2008, pp. 656–667.
[32] F. Lu, T. Okabe, Y. Sugano, Y. Sato, A head pose-free approach for appearance-based gaze estimation, In: BMVC, 2011.
[33] F. Lu, Y. Sugano, T. Okabe, Y. Sato, Head pose-free appearance-based gaze sensing via eye image synthesis, In: ICPR, 2012.
[34] F. Lu, T. Okabe, Y. Sugano, Y. Sato, Learning gaze biases with head motion for head pose-free gaze estimation, Image Vis. Comput. 32 (3) (2014) 169–179.
[35] K. Grauman, T. Darrell, The pyramid match kernel: discriminative classification with sets of image features, In: Tenth IEEE International Conference on Computer Vision, 2005, ICCV 2005, vol. 2, IEEE, Beijing, China, 2005, pp. 1458–1465.
[36] Y. Sugano, Y. Matsushita, Y. Sato, Learning-by-synthesis for appearance-based 3d gaze estimation, In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[37] B.A. Olshausen, et al., Emergence of simple-cell receptive eld properties by learning a sparse code for natural images, Nature 381 (6583) (1996) 607–609.
[38] H. Lee, A. Battle, R. Raina, A. Y. Ng, Ecient sparse coding algorithms, in: Advances in neural information processing systems, 2006, pp. 801–808.
[39] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 2169–2178.

Feng Lu received the B.S. and M.S. degrees in automation from Tsinghua University, in 2007 and 2010, respectively, and the Ph.D. degree in information science and technology from The University of Tokyo, in 2013. After working with the Institute of Industrial Science, the University of Tokyo, he joined the Beihang University, China, in 2015. His research interests include human gaze analysis, shape recovery, and reflectance analysis.

Xiaowu Chen received the PhD degree in computer science from Beihang University, Beijing, China, in 2001. He is currently a professor with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University. His current research interests include virtual reality, computer graphics, and computer vision.