Appearance-based Gaze Estimation using Visual Saliency

Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato

Abstract—We propose a gaze sensing method using visual saliency maps that does not need explicit personal calibration. Our goal is to create a gaze estimator using only the eye images captured from a person watching a video clip. Our method treats the saliency maps of the video frames as the probability distributions of the gaze points. We aggregate the saliency maps based on the similarity in eye images to efficiently identify the gaze points from the saliency maps. We establish a mapping between the eye images to the gaze points by using Gaussian process regression. In addition, we use a feedback loop from the gaze estimator to refine the gaze probability maps to improve the accuracy of the gaze estimation. The experimental results show that the proposed method works well with different people and video clips and achieves a 3.5-degree accuracy, which is sufficient for estimating a user's attention on a display.

Index Terms—Gaze estimation, Visual attention, Face and gesture recognition.

1 Introduction

Gaze estimation is important for predicting human attention, and therefore can be used to better understand human activities as well as interactive systems. There is a wide range of applications for gaze estimation including market analysis of online content and digital signage, gaze-driven interactive displays, and many other human-machine interfaces.

In general, gaze estimation is achieved by analyzing the appearance of a person's eyes. There are two categories of camera-based remote sensing methods: Model-based and appearance-based. Model-based methods use a geometric eye model and its associated features. Using specialized hardware such as multiple synchronized cameras and infrared light sources, they extract the geometric features of an eye to determine the gaze direction. Appearance-based methods, on the other hand, use the natural appearances of eyes observed from a commodity camera without requiring any dedicated hardware. Various implementations of camera-based gaze estimators have been proposed including commercial products (see [1] for a recent survey).

One of the key challenges in previous gaze estimators is the need for explicit *personal* calibration to adapt to individual users. The users in these existing systems are always required to actively participate in calibration tasks by fixating their eyes on explicit reference points. Another problem that most estimation methods suffer from is calibration drift, and their calibration accuracy highly depends on the users and installation settings. An interactive local calibration scheme with, *e.g.*, user feedback [2], is sometimes required in practical application systems to correct personal calibration errors. In many scenarios, such active personal calibration is too restrictive as it interrupts natural interactions and makes unnoticeable gaze estimation impossible. Although the number of reference points for personal calibration can be reduced using specialized hardware such as multiple light sources [3], [4], [5] and stereo cameras [6], it still requires a user to actively participate in the calibration task.

It is also well known in the class of model-based approaches that the gaze direction can be approximately estimated as the direction of the optical axis without requiring personal calibration [7]. However, its offset with the visual axis, which corresponds to the actual gaze direction, can be as large as 5 degrees [1], [4], and the accuracy varies significantly based on the individual. More importantly, such hardware-based attempts add a strong constraint to the application setting, and this naturally limits the user scenarios.

There are previous studies that aim at completely removing the need for explicit personal calibration processes. Yamazoe *et al*. use a simple eyeball model for gaze estimation and perform automatic calibration by fitting the model to the appearance of a user's eye while the user is moving his/her eyes [8]. In Sugano *et al*.'s method, in a similar spirit to [2], a user's natural mouse inputs are used for the incremental personal calibration of the appearance-based gaze estimation without any calibration instructions [9]. Both methods use only a monocular camera, however, these approaches still have some limitations. Yamazoe *et al*.'s approach suffers from inaccuracy due to the simplified eyeball model, and Sugano *et al*.'s approach can only be applied to interactive environments with user inputs.

Apart from these gaze estimation studies, computational models of visual saliency have been studied to estimate the visual attention on an image, which is computed in a bottom-up manner. In contrast to gaze estimation approaches that aim to determine where peoples' eyes actually

- *Y. Sugano and Y. Sato are with the Institute of Industrial Science, the University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505, Japan. E-mail: {sugano,ysato}@iis.u-tokyo.ac.jp*
- *Y. Matsushita is with Microsoft Research Asia, 13F, Building 2, No. 5 Dan Ling Street, Haidian district, Beijing, 100080, China. E-mail: yasumat@microsoft.com*

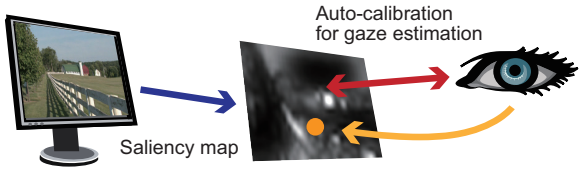

Fig. 1. Illustration of our method. Our method uses saliency maps computed from video frames in bottom-up manner to automatically construct a gaze estimator.

look, visual saliency computes the image region that is likely to attract human attention. Biologically, a human tends to gaze at an image region with high saliency, *i.e.*, a region containing unique and distinctive visual features compared to the surrounding regions. After Koch and Ullman's original concept [10] of visual saliency, various bottom-up computational models of visual saliency maps have been proposed in [11], [12], [13], [14], [15]. Experiments show that there is a correlation between bottom-up visual saliency and fixation locations [16]. However, the visual attention mechanism is not yet fully understood. It is already known that fixation prediction becomes much more difficult under natural dynamic scenes, in which a high-level task and knowledge have a stronger influence on the gaze control [17].

Gaze estimation (top-down) and visual saliency (bottom-up) models are closely related. Nonetheless, not many studies exist that bridge these two subjects. Kienzle *et al*. [18], [19] propose a method for learning the computational models of bottom-up visual saliency by using the gaze estimation data. A visual saliency map is modeled in their work as a linear combination of the Gaussian radial basis functions, and their coefficients are learned using a support vector machine (SVM). Judd *et al*. [20] and Zhao and Koch [21] also use this approach with different features and a larger database. The linear weights of low-level image features (*e.g*., color and intensity) and high-level features (*e.g*., face detector) are learned via the SVM in [20]. In [21], the optimal feature weights are learned by solving a non-negative least squares problem using an active set method. These approaches learn accurate saliency models using gaze points. In contrast to these methods, our goal is to create a gaze estimator from the collection of visual saliency maps. To our knowledge, this is the first work using visual saliency as prior information for gaze estimation.

In this paper, we propose a novel gaze sensing method that uses computational visual saliency, as illustrated in Fig. 1. Our approach is based on the assumption that bottom-up visual saliency is correlated with actual gaze points. By computing the visual saliency maps from a video and relating them with the associated eye images of a user, our method automatically learns the mapping from the eye images to the gaze points. We aggregate the saliency maps based on the similarity of the eye images to produce reliable maps, which we call *gaze probability maps* in this paper, to handle low prediction accuracy of raw saliency maps. Once the gaze probability maps are obtained, our method learns the relationship between the gaze probability maps and the eye images. In addition, a feedback scheme optimizes the feature weights used to compute the visual saliency maps. The feedback loop enables us to further strengthen the correlation between the gaze probability maps and the eye images. From one point of view, our method closes the bottom-up visual saliency and top-down gaze estimates loop; the visual saliency determines the likely location of the gaze points, and the gaze points in return refine the computation of the visual saliency. We demonstrate our approach through extensive user testing and verify the effectiveness of the use of visual saliency for gaze estimation.

Our method takes a set of eye images recorded in synchronization with any video clip as the input. From such an input, our method automatically determines the relationship between the eye images and gaze directions. In addition, our method does not distinguish between the test data and training data, *i.e*., one dataset can be used for both the calibration and estimation at the same time. Therefore, when only the gaze estimates for a particular video clip are needed, a user only needs to watch the video clip once. Once the relationship is learned, our gaze estimator can be used in other application scenarios as long as the configuration between the camera and user remains unchanged. In this manner, the proposed framework leads to a gaze estimation technique that exempts the users from the active personal calibration.

In general, a fundamental trade-off between the accuracy and a system's portability exists. Our system aims at minimizing the hardware and calibration constraints for developing a fully ambient gaze estimation technique, which is a key factor for opening up a new way of attentive user interface [22], [23]. For example, to collect the gaze data over a film clip on a public display, the film creator may only have to place a camera to capture the eye images of the audience. Similarly, movie players on PCs can naturally obtain gaze data for media understanding without the users' notice. In addition, the calibrated gaze estimators can be used for gaze-based interactions. Our method can further enhance the calibration accuracy during the gaze estimation process by using the eye images as input. By closing the loop of calibration and estimation in this manner, this work aims at enhancing the approach of calibrating gaze estimators through daily activities [9].

The preliminary version of this work appeared in [24]. A closely related work is recently introduced by Chen and Ji [25]. They use the idea of using visual saliency maps for model-based gaze estimation of a person looking at still pictures. While Chen and Ji's approach achieves a higher level of accuracy and allows for free head movement, their results rely on a model-based setup with a longer recording time on a single image. In contrast, our system uses an appearance-based estimation and is built using only a monocular camera. While it is often discussed that a gaze prediction from saliency maps is more reliable when using static photographs than when using video clips, our method avoids this problem via the aggregation of the saliency

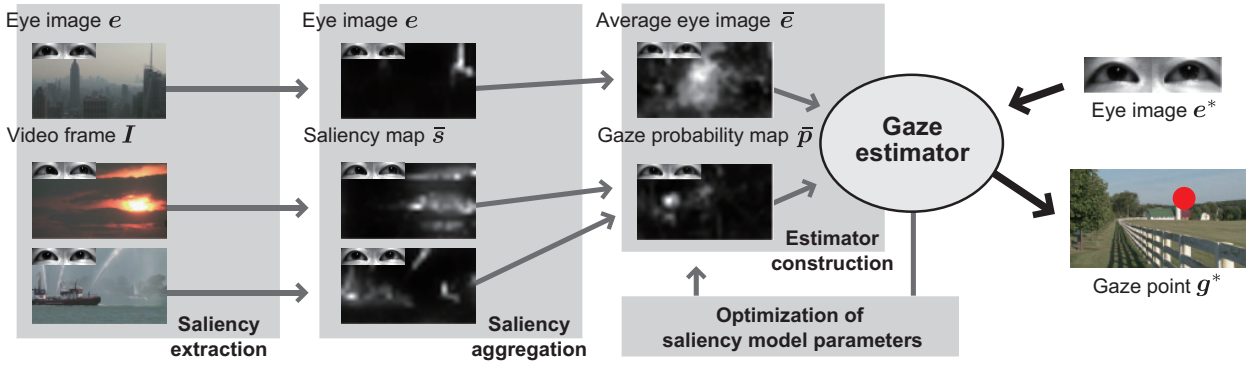

Fig. 2. Illustration of proposed approach. Our method consists of four steps. The saliency extraction step computes saliency maps from the input video. The saliency aggregation step combines saliency maps to produce gaze probability maps. Using the gaze probability maps and associated average eye images, the estimator construction step learns the mapping from an eye image to a gaze point. A feedback loop is used to optimize the feature weights to improve the accuracy by using cross validation.

maps, which results in statistically accurate and stable gaze probability maps.

This paper is organized as follows. In Section 2, we describe the proposed gaze estimation method that auto-calibrates from the bottom-up saliency maps. Section 3 describes the feedback loop from the estimated gaze point to the saliency weight computation. This feedback loop is intended to bridge the gap between the top-down gaze point and the bottom-up visual saliency, and improves gaze estimation accuracy. Finally, we validate the proposed method by conducting user tests in Section 4. Our results show that our method can achieve a 3.5-degree of accuracy without needing any specialized hardware or explicit personal calibration processes.

2 Gaze Estimation from Saliency Maps

Our goal is to construct a gaze estimator without an calibration stage. Our method assumes a fixed head pose and fixed relative positions among the user's head, camera, and display. The term calibration indicates obtaining the mapping function from an eye image to a point on the display coordinate. The relationship between the eye images (input) and the gaze points (output) is expressed as a single regression function in an appearance-based gaze estimation, and our goal is to estimate the parameters of the gaze estimation function without using explicit training data.

The inputs for our system are N video frames $\{\boldsymbol{I}_1, \ldots, \boldsymbol{I}_N\}$ and associated feature vectors $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_N\}$ extracted from the eye images of a person who is watching a video clip with a fixed head position. The implementation details of the feature vector $\boldsymbol{e}$ are described in Section 4.1; but our framework does not depend on specific image features. For presentation clarity, we denote $\boldsymbol{e}$ simply as *eye image* in this paper. In our setting, the eye images and video frames are synchronized. The i-th eye image $\boldsymbol{e}_i$ is captured at the same time as frame $\boldsymbol{I}_i$ is shown to the person. Using this dataset $\{(\boldsymbol{I}_1, \boldsymbol{e}_1), \ldots, (\boldsymbol{I}_N, \boldsymbol{e}_N)\}$, a gaze estimation function from an eye image $\boldsymbol{e}^*$ to an unknown gaze point $\boldsymbol{g}^*$ is built.

Our method consists of four steps: Saliency extraction, saliency aggregation, estimator construction, and feature weights optimization, as illustrated in Fig. 2. Once the saliency maps are computed in the saliency extraction step, the saliency aggregation step produces gaze probability maps that have a higher concentration of gaze point estimates than the saliency maps. The average eye images are computed by clustering the eye images, and all the saliency maps are aggregated according to the eye image similarities to compute the gaze probability maps. Using the gaze probability maps and associated average eye images, the estimator construction step learns the mapping from an eye image to a gaze point by using a variant of the Gaussian process regression. Our method further optimizes the feature weights that are used for the saliency computation by using the feedback loop. By optimizing the weights in a cross-validation manner, this fourth step improves the accuracy of the gaze estimator. The resulting gaze estimator outputs the gaze points for any eye image of the user. In the following subsections, we describe the details of the saliency extraction, aggregation, and estimator construction steps, and in Section 3 the feature weight optimization.

2.1 Saliency Extraction

This step extracts the visual saliency maps from the input video frames $\{\boldsymbol{I}_1, \ldots, \boldsymbol{I}_N\}$. As shown in Fig. 3, our method adopts six features to compute the saliency maps: five low-level features and one high-level feature.

Each frame $\boldsymbol{I}$ is first decomposed into multiple feature maps $\boldsymbol{F}$. We use commonly-used feature channels, *i.e.*, color, intensity, and orientations as the static features, and flicker and motion are used as dynamic features in our method. The intensity channel indicates the grayscale luminance, two color channels are red/green and blue/yellow differences, and four orientation channels are the responses from the 2D Gabor filters with orientation at $0°$, $45°$, $90°$,

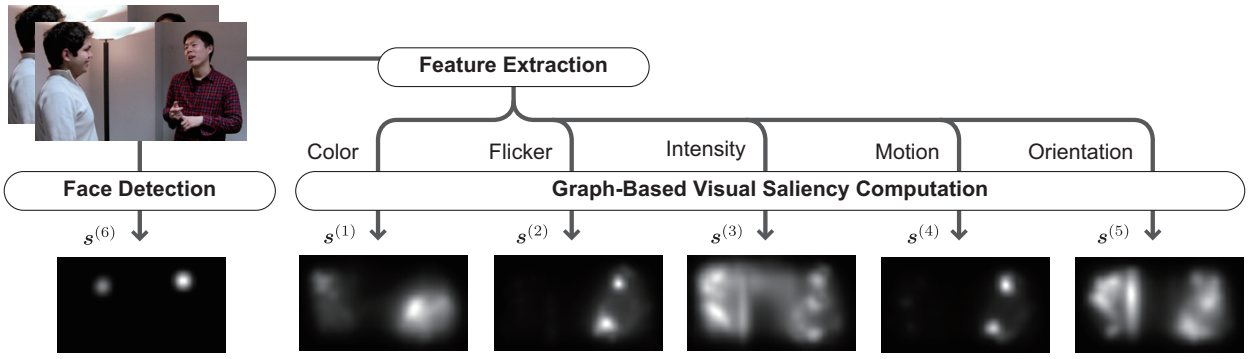

Fig. 3. Computation of visual saliency maps. Our method uses six features to compute the saliency maps: Face detection-based high level saliency, and five low-level features obtained by using a graph-based saliency computation; color, intensity, orientation, flicker, and motion. $\boldsymbol{s}^{(1)} \sim \boldsymbol{s}^{(6)}$ show examples of computed saliency maps.

and $135°$, respectively. The flicker channel indicates an absolute intensity difference from the previous frame, and four motion channels use the spatially-shifted differences between the Gabor responses. The feature maps are computed at three levels of the image pyramid which are $1/2$, $1/4$, and $1/8$ of the original image resolution. As a result, 36 (3 levels $\times$ (1 intensity + 2 color + 4 orientation + 1 flicker + 4 motion)) feature maps $\boldsymbol{F}$ are computed.

The saliency maps are then computed from the feature maps $\boldsymbol{F}$ using a Graph-based Visual Saliency (GBVS) [14]. Computation in the GBVS algorithm is conducted in two stages: Activation and normalization. *Activation maps* $\boldsymbol{A}$ are first computed from the feature maps $\boldsymbol{F}$ to locate the regions with prominent image features. Greater values are assigned to the pixels in activation maps $\boldsymbol{A}$ where they have distinct values compared with their surrounding regions in the feature maps. In the GBVS algorithm, this computation is performed in a form of a steady-state analysis of a Markov chain G_A. Each node of G_A corresponds to a pixel position in feature maps $\boldsymbol{F}$, and a transition probability Ω_a between nodes (i, j) and (p, q) is defined based on a dissimilarity between the two corresponding pixels in $\boldsymbol{F}$ as

$$\Omega_a((i,j),(p,q)) \triangleq \Omega_d|\boldsymbol{F}(i,j) - \boldsymbol{F}(p,q)|, \tag{1}$$

where Ω_d indicates the Gaussian weight evaluating the Euclidean distance between (i, j) and (p, q). In this way, the nodes (= pixels) with a higher dissimilarity to their surroundings have higher transition probabilities. Therefore, by iteratively computing the equilibrium distribution $\boldsymbol{d}_a$ (a raster-scanned vector form of $\boldsymbol{A}$) of G_A that satisfies

$$\boldsymbol{\Omega}_a \boldsymbol{d}_a = \boldsymbol{d}_a, \tag{2}$$

where $\boldsymbol{\Omega}_a$ is the transition probability matrix consisting of Ω_a, the salient pixels in $\boldsymbol{F}$ have larger values in $\boldsymbol{A}$.

Since the resulting activation maps often have many insignificant peaks, the GBVS algorithm further normalizes them to suppress the local maxima. Using the computed activation maps $\boldsymbol{A}$, a Markov chain G_N is defined in a similar way with a transition probability Ω_n as:

$$\Omega_n((i,j),(p,q)) \triangleq \Omega_d|\boldsymbol{A}(p,q)|. \tag{3}$$

By computing the equilibrium distribution of G_N as described above, the resulting maps are concentrated so that they have fewer important peaks. These normalized activation maps are averaged within each channel, and as a result, five low-level saliency maps $\boldsymbol{s}^{(1)}, \ldots, \boldsymbol{s}^{(5)}$ are computed.

It is well known that humans tend to fixate on faces, especially on the eyes, which are highly salient for humans. With this observation, Cerf *et al*. [26] proposed a face channel-based saliency model using a face detector. We follow this approach to produce reliable saliency maps using the facial features. We use a facial feature detector OKAO Vision library developed by OMRON Corporation [27] to obtain facial features. This sixth saliency map $\boldsymbol{s}^{(6)}$ is modeled as 2-D Gaussian circles with a fixed variance at the center of two detected eye positions. When the detector only detects a face but not the eyes, *e.g*., due to a limited resolution, the facial saliency is defined at the center of the facial region.

Finally, our method computes the temporal average of each saliency map $\bar{\boldsymbol{s}}_i^{(1)}, \ldots, \bar{\boldsymbol{s}}_i^{(6)}$ within a temporal window n_s as

$$\bar{\boldsymbol{s}}_i^{(f)} = \frac{1}{n_s + 1} \sum_{j=i-n_s}^{i} \boldsymbol{s}_j^{(f)}, \tag{4}$$

where $\boldsymbol{s}_j^{(f)}$ is the raw saliency map of the f-th feature computed from the j-th frame, and n_s is the number of frames used for temporal averaging. This is because humans cannot instantly follow rapid scene changes, and only the past frames are used for the smoothing to account for the latency. As a result, synchronized pairs of saliency maps and eye images $\mathcal{D}_s = \{(\bar{\boldsymbol{s}}_1^{(1)}, \ldots, \bar{\boldsymbol{s}}_1^{(6)}, \boldsymbol{e}_1), \ldots, (\bar{\boldsymbol{s}}_N^{(1)}, \ldots, \bar{\boldsymbol{s}}_N^{(6)}, \boldsymbol{e}_N)\}$ are produced.

2.2 Saliency Aggregation

Although it is assumed that the saliency maps can predict gaze points, their accuracy is insufficient for determining the exact gaze point locations as discussed in previous studies [17]. In this section, we describe our method to

compute the probability distribution of the gaze point by aggregating the computed saliency maps.

The computed saliency maps $\{\bar{\boldsymbol{s}}^{(f)}\}$ encode the distinctive visual features of video frames. While the saliency map does not provide the exact gaze point, highly salient regions in the saliency map are likely to coincide with the actual gaze point. Suppose we have a set of saliency maps that statistically have high saliency scores around the actual gaze point and random saliency scores at other regions. Since we assume a fixed head position, there is a one-to-one correspondence between the gaze points and the eye images; actual gaze points would be almost the same between visually similar eye images. Therefore, by aggregating the saliency maps based on the similarity of the associated eye images, we can assume that the image region around the actual gaze point has a vivid peak of saliency compared with other regions. The aggregated map can be used as the *gaze probability map*, *i.e.*, the probability distribution of the gaze point.

The similarity score w_s of a pair of eye images $\boldsymbol{e}_i$ and $\boldsymbol{e}_j$ is defined as

$$w_s(\boldsymbol{e}_i, \boldsymbol{e}_j) = \exp(-\kappa_s^2 ||\boldsymbol{e}_i - \boldsymbol{e}_j||^2), \tag{5}$$

where the factor κ_s controls the similarity score. The similarity score w_s is higher when the appearances of two eye images are similar, *i.e.*, the gaze points of the eye images are close. Since the appearance variation of the eye images is quite large for different people, the optimal value of κ_s in Eq. (5) is highly person-dependent. Therefore, in this work, the factor κ_s is indirectly defined via the range of the values taken by w_s. More specifically, κ_s is optimized by minimizing an error defined as

$$\kappa_s = \underset{\kappa_s}{\operatorname{argmin}} ||T_s - \det(\boldsymbol{W}_s)||^2, \tag{6}$$

where $\boldsymbol{W}_s \in \mathbb{R}^{N_s \times N_s}$ is a similarity weight matrix computed using randomly selected N_s eye images in $\mathcal{D}_s$. T_s is the target value of the determinant that is empirically defined, *e.g.*, by quantitatively checking a sample dataset. The factor κ_s is determined to adapt to the person-dependency by minimizing Eq. (6) via the gradient descent.

We eliminate the eye images that are not useful for gaze estimation from the dataset, *e.g.*, eye images of blinking, prior to the computation of the gaze probability maps. On the other hand, the eye images recorded during fixation are useful as training data. To automatically identify such eye images, we use a fixation measure of an eye image $\boldsymbol{e}$ defined as

$$w_e(\boldsymbol{e}_i) = \exp(-\alpha_e \kappa_s^2 \mathrm{Var}(\boldsymbol{e}_i)), \tag{7}$$

where α_e is a weighting factor, and $\mathrm{Var}(\boldsymbol{e}_i)$ denotes the variance in the eye images $\{\boldsymbol{e}_{i-n_f}, \ldots, \boldsymbol{e}_{i+n_f}\}$ over a temporal window $2n_f + 1$ centered at i,

$$\mathrm{Var}(\boldsymbol{e}_i) = \sum_{j=i-n_f}^{i+n_f} ||\boldsymbol{e}_j - \mu_{\boldsymbol{e}_i}||^2, \tag{8}$$

$$\mu_{\boldsymbol{e}_i} = \frac{1}{2n_f + 1} \sum_{k=i-n_f}^{i+n_f} \boldsymbol{e}_k. \tag{9}$$

Eq. (7) evaluates the stability of the eye regions, and it is assumed that there are no significant changes in the lighting conditions during the temporal window. Since the appearances of the eye images rapidly change during the fast movement of the eyes, $w_e(\boldsymbol{e}_i)$ becomes small when $\boldsymbol{e}_i$ is captured during eye movement or blinking. A subset $\mathcal{D}_{s'} = \{(\bar{\boldsymbol{s}}_1^{(1)}, \ldots, \bar{\boldsymbol{s}}_1^{(6)}, \boldsymbol{e}_1), \ldots, (\bar{\boldsymbol{s}}_{N'}^{(1)}, \ldots, \bar{\boldsymbol{s}}_{N'}^{(6)}, \boldsymbol{e}_{N'})\}$ is created from $\mathcal{D}_s$ by removing the eye images where the w_e scores are lower than a predefined threshold τ_f.

Since variation in the gaze points is limited in $\mathcal{D}_{s'}$ and there can be many samples that share almost the same gaze point, the eye images are clustered according to similarity w_s to reduce redundancy and computational cost. Using the similarity score (Eq. (5)), each eye image $\boldsymbol{e}_i$ is sequentially added to the cluster whose average eye image $\bar{\boldsymbol{e}}$ is the most similar to $\boldsymbol{e}_i$. A new cluster is adaptively created if the highest similarity among all existing clusters is lower than a threshold τ_e. M clusters and their average eye images $\{\bar{\boldsymbol{e}}_1, \ldots, \bar{\boldsymbol{e}}_M\}$ are computed from these computations.

After these steps, a gaze probability map $\bar{\boldsymbol{p}}_i^{(f)}$ of each feature f is computed as

$$\bar{\boldsymbol{p}}_i^{(f)} = \frac{\sum_{j=1}^{N'} w_s(\bar{\boldsymbol{e}}_i, \boldsymbol{e}_j)(\bar{\boldsymbol{s}}_j^{(f)} - \bar{\boldsymbol{s}}_{\mathrm{all}}^{(f)})}{\sum_{j=1}^{N'} w_s(\bar{\boldsymbol{e}}_i, \boldsymbol{e}_j)}, \tag{10}$$

where $\bar{\boldsymbol{s}}_{\mathrm{all}}^{(f)}$ is the average of all the maps $\bar{\boldsymbol{s}}_1^{(f)}, \ldots, \bar{\boldsymbol{s}}_{N'}^{(f)}$ in $\mathcal{D}_{s'}$. It is known that a center bias of visual saliency [20], [21] exists because man-made pictures usually have a higher saliency at the center of the image. The average saliency map $\bar{\boldsymbol{s}}_{\mathrm{all}}^{(f)}$ is used to eliminate this center bias in a gaze probability map. Without this, the gaze probability map tends to have a higher value at the center regardless of the eye image $\bar{\boldsymbol{e}}_i$. The gaze probability map $\bar{\boldsymbol{p}}_i^{(f)}$ can also have negative values. In our case, only the relative differences matter, and therefore, we used the computed results by normalizing the values to a fixed range. We again use the graph-based normalization scheme (Eq. (3)) to the gaze probability maps $\bar{\boldsymbol{p}}_i^{(f)}$ in order to enhance the peaks in the gaze probability maps.

The final gaze probability map $\bar{\boldsymbol{p}}_i$ is computed as a weighted sum of all the feature-dependent maps $\bar{\boldsymbol{p}}^{(f)}$ as

$$\bar{\boldsymbol{p}}_i = \sum_{f=1}^{6} \omega_f \bar{\boldsymbol{p}}_i^{(f)}, \tag{11}$$

where ω_f is a weight for f-th feature. $\bar{\boldsymbol{p}}_i$ is then normalized to a fixed range, and we obtain a dataset $\mathcal{D}_p = \{(\bar{\boldsymbol{p}}_1, \bar{\boldsymbol{e}}_1), \ldots, (\bar{\boldsymbol{p}}_M, \bar{\boldsymbol{e}}_M)\}$. We followed many existing visual saliency map models and used equal weights at this step to aggregate feature maps. However, it is often

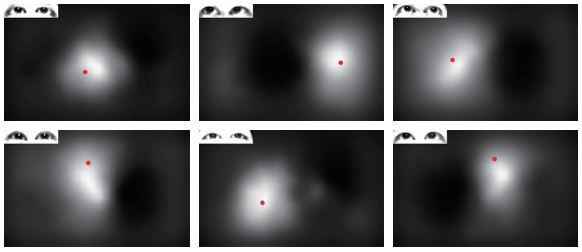

Fig. 4. Examples of gaze probability maps $\bar{\boldsymbol{p}}$ and corresponding average eye images $\bar{\boldsymbol{e}}$. The overlaid dots depict the actual gaze points of $\bar{\boldsymbol{e}}$ to illustrate the correspondence between the gaze points and the gaze probability. The true gaze points are obtained using a calibration-based gaze estimator.

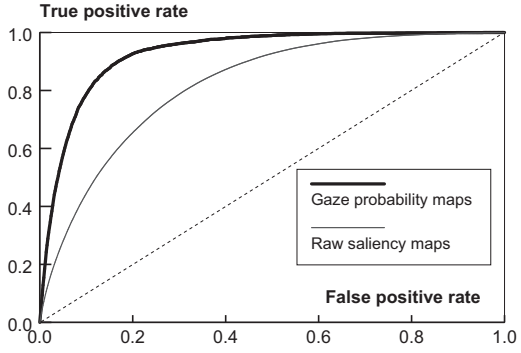

Fig. 5. ROC curves of raw saliency maps and gaze probability maps. The horizontal axis indicates the false positive rate, *i.e.*, the rate of the pixels above a threshold. The vertical axis indicates the true positive rate, *i.e.*, the rate of the frames that have a higher saliency value than the threshold at the true gaze point. The thin line (AUC = 0.82) indicates the performance of the raw saliency maps obtained by the process described in Section 2.1. The bold line (AUC = 0.93) indicates the performance of the gaze probability maps described in Section 2.2.

pointed out that the contribution of each feature is not uniform, and there is a certain degree of data dependency. We use a feedback scheme to optimally adjust the weight parameters to address these issues. The feedback scheme is discussed in Section 3.

Fig. 4 shows examples of the obtained gaze probability maps $\bar{\boldsymbol{p}}$ for six people. The eye images shown at the top-left of each sub-figure indicate the corresponding average eye images $\bar{\boldsymbol{e}}$, and the overlaid dots indicate the actual gaze points of $\bar{\boldsymbol{e}}$. Note that $\bar{\boldsymbol{e}}$ is a prototype of the eye images synthesized through the above process, and the actual gaze points are unknown. Therefore, we used the estimates from the appearance-based gaze estimator using explicit calibration, which is described in Section 4, to obtain the true gaze points as a reference. Although the gaze probability maps $\bar{\boldsymbol{p}}_i$ are generated without knowing the actual gaze points, they have a significant correlation with the actual gaze points.

We compare the gaze probability maps with the original raw saliency maps to assess the correlation improvement with the actual gaze points. Fig. 5 shows the correlation improvement using a receiver operating characteristic (ROC) curve. We sweep the threshold value for the gaze probability maps and raw saliency maps to obtain the plots, and assess all the ground truth gaze points that we obtain through the experiment. The horizontal axis represents the false positive rate, *i.e.*, the rate of the pixels in a map above a threshold value. The vertical axis is the true positive rate, which indicates the rate of frames whose saliency value at the gaze point is greater than the threshold. The area under the curve (AUC) of the gaze probability maps is 0.93, and that of the raw saliency maps is 0.82. This result shows that the correlation is significantly enhanced by the aggregation process.

2.3 Estimator Construction

In the previous step, M average eye images $\{\bar{\boldsymbol{e}}_1, \ldots, \bar{\boldsymbol{e}}_M\}$ and corresponding gaze probability maps $\{\bar{\boldsymbol{p}}_1, \ldots, \bar{\boldsymbol{p}}_M\}$ are obtained. This section describes our method for creating a gaze estimator using them as the training dataset. Our goal is to establish a mapping from the eye image to gaze points. We develop a method based on Gaussian process regression that has been successfully applied to both appearance-based [28] and model-based [29] gaze estimation to efficiently achieve this task.

With the standard Gaussian process regression framework, an estimator is built to output the probability distribution $P(\boldsymbol{g}^*|\boldsymbol{e}^*, \mathcal{D}_g)$ of an unknown gaze point $\boldsymbol{g}^*$ from an eye image $\boldsymbol{e}^*$, given the labeled training data $\mathcal{D}_g = \{(\boldsymbol{g}_1, \bar{\boldsymbol{e}}_1), \ldots, (\boldsymbol{g}_M, \bar{\boldsymbol{e}}_M)\}$, which consists of the eye images and corresponding gaze points. In our case, however, we only know $\mathcal{D}_p = \{(\bar{\boldsymbol{p}}_1, \bar{\boldsymbol{e}}_1), \ldots, (\bar{\boldsymbol{p}}_M, \bar{\boldsymbol{e}}_M)\}$ where the gaze probability map $\bar{\boldsymbol{p}}_i$ only provides the probability distribution of the gaze point $\boldsymbol{g}_i$. Therefore, instead of directly applying the standard Gaussian process regression, we work on a marginalized probability where explicit training labels are not required.

After normalizing the gaze probability maps, we define the gaze probability distribution $P(g|\bar{\boldsymbol{p}})$ as

$$P(g|\bar{\boldsymbol{p}}) = \frac{\bar{\boldsymbol{p}}(g)}{\sum_x \sum_y \bar{\boldsymbol{p}}}, \tag{12}$$

where $\bar{\boldsymbol{p}}(g)$ indicates the value of $\bar{\boldsymbol{p}}$ at the gaze point g, and $\sum_x \sum_y \bar{\boldsymbol{p}}$ is the overall summation of $\bar{\boldsymbol{p}}$. In the above equation, we describe the estimation of a one-dimensional scalar g to simplify the notation, but two regressors are independently built for each X- and Y-direction in the actual implementation. Using Eq. (12), the target distribution $P(g^*|e^*, \mathcal{D}_p)$ can be obtained by marginalizing over all the possible combinations of M gaze points $\hat{\mathcal{D}}_g = \{\hat{g}_1, \ldots, \hat{g}_M\}$ as

$$P(g^*|e^*, \mathcal{D}_p) = \sum_{\hat{\mathcal{D}}_g} P(g^*|e^*, \hat{\mathcal{D}}_g) P(\hat{\mathcal{D}}_g|\mathcal{D}_p), \tag{13}$$

where

$$P(\hat{\mathcal{D}}_g|\mathcal{D}_p) = \prod_i^M P(\hat{g}_i|\bar{\boldsymbol{p}}_i). \tag{14}$$

In Eq. (13), g^* indicates an unknown gaze point associated with the eye image $\boldsymbol{e}^*$, and $\hat{g}_i$ is a candidate of the gaze point that corresponds to $\bar{\boldsymbol{e}}_i$.

However, the summations of the Eq. (13) are computationally expensive. Various approximation techniques for the Gaussian process regression [30] can reduce the computational cost of computing $P(g^*|\boldsymbol{e}^*, \hat{\mathcal{D}}_g)$, however they cannot directly approximate Eq. (13) itself. For these reasons, we solve Eq. (13) using a Monte Carlo approximation. We randomly produce n_g sets of samples $\mathcal{D}_g^{(l)} = \{(g_1^{(l)}, \bar{\boldsymbol{e}}_1), \ldots, (g_M^{(l)}, \bar{\boldsymbol{e}}_M)\}_{l=1}^{n_g}$ according to the probability distribution defined by Eq. (12). In particular, $g_i^{(l)}$ in the l-th set is generated according to the distribution $P(g_i|\bar{\boldsymbol{p}}_i)$ defined by the i-th probability map. It has been experimentally shown in Fig. 5 that gaze probability maps have a high correlation with the actual gaze points. From this observation, we discard the low saliency values from the gaze probability maps to reduce the number of samples. We use a threshold τ_s to set the probability to zero if $\bar{\boldsymbol{p}}(x, y)$ is lower than the threshold. Using these sets, Eq. (13) can be approximated as

$$P(g^*|\boldsymbol{e}^*, \mathcal{D}_p) = \frac{1}{n_g}\sum_{l=1}^{n_g} P(g^*|\boldsymbol{e}^*, \mathcal{D}_g^{(l)}). \tag{15}$$

Finally, $P(g^*|\boldsymbol{e}^*, \mathcal{D}_g^{(l)})$ for each l is computed based on the Gaussian process regression as follows [30].

Gaussian Process Regression.

We assume a noisy observation model for a gaze point $g_i = f(\boldsymbol{e}_i) + \epsilon_i$, *i.e.*, a gaze point g_i is given as a function of the eye image $\boldsymbol{e}_i$ with a noise term $\epsilon_i = \mathcal{N}(0, \gamma^2\varsigma_i^2)$. The data-dependent noise variance $\gamma^2\varsigma_i^2$ is defined as being proportional to ς_i^2, which is an actual variance of the generated samples $\{g_i^{(1)}, \ldots, g_i^{(n_g)}\}$. It explicitly assigns a higher noise variance for samples from ambiguous saliency maps that have several peaks. The function $f(\boldsymbol{e}_i)$ is assumed to be a zero-mean Gaussian with a covariance function k:

$$k(\boldsymbol{e}_i, \boldsymbol{e}_j) = \alpha^2 \exp(-\beta^2||\boldsymbol{e}_i - \boldsymbol{e}_j||^2), \tag{16}$$

with hyperparameters α and β. With this assumption, $P(g^*|\boldsymbol{e}^*, \mathcal{D}_g^{(l)})$ is derived as a Gaussian distribution $\mathcal{N}(\mu_l, \sigma_l^2)$ that has a mean μ_l and variance σ_l^2:

$$\mu_l = \boldsymbol{K}^*\boldsymbol{K}^{-1}\boldsymbol{G}^{(l)}, \tag{17}$$

and

$$\sigma_l^2 = k(\boldsymbol{e}^*, \boldsymbol{e}^*) - \boldsymbol{K}^*\boldsymbol{K}^{-1t}\boldsymbol{K}^*, \tag{18}$$

where the matrix $\boldsymbol{K} \in \mathbb{R}^{M\times M}$ is a covariance matrix where its (i, j)-th element is defined as $\boldsymbol{K}_{ij} = k(\bar{\boldsymbol{e}}_i, \bar{\boldsymbol{e}}_j) + \gamma^2\varsigma_i^2\delta_{ij}$. The vector $\boldsymbol{K}^* \in \mathbb{R}^{1\times M}$ represents a covariance vector of the input eye image and average eye images, whose i-th element is $\boldsymbol{K}_i^* = k(\bar{\boldsymbol{e}}_i, \boldsymbol{e}^*)$, and $\boldsymbol{G}^l \in \mathbb{R}^{1\times M}$ is a vector of the gaze points, where its i-th element is $\boldsymbol{G}_i^{(l)} = g_i^{(l)}$. As a result, the distribution $P(g^*|\boldsymbol{e}^*, \mathcal{D}_p)$ can be estimated as a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ with

$$\mu = \frac{1}{n_g}\sum_{l=1}^{n_g}\mu_l, \quad \sigma^2 = \frac{1}{n_g}\sum_{l=1}^{n_g}\sigma_l^2 = \sigma_1^2. \tag{19}$$

The variance σ^2 is simply the same as σ_1^2, because σ_l^2 of Eq. (18) is actually independent of the index l.

Tuning Hyperparameters.

There are three hyperparameters, α, β, and γ, in the above formulation, which need to be optimized for each dataset. We use a cross-validation approach [31] for the optimization in our method. The optimal parameters can be estimated by maximizing a leave-one-out log predictive probability L defined as

$$L(\{\mathcal{D}_g^{(l)}\}, \boldsymbol{\theta}) = \sum_{l=1}^{n_g}\sum_{i=1}^{M} \log p(g_i^{(l)}|\{\mathcal{D}_g^{(l)}\}_{-i}, \boldsymbol{\theta}), \tag{20}$$

where $\boldsymbol{\theta}$ is a set of hyperparameters $\boldsymbol{\theta} = \{\alpha, \beta, \gamma\}$, and $\{\mathcal{D}_g^{(l)}\}_{-i}$ is a set of generated samples that excludes the samples with the i-th eye image. The predictive probability $p(g_i^{(l)}|\{\mathcal{D}_g^{(l)}\}_{-i}, \boldsymbol{\theta})$ is defined as a Gaussian function as

$$\log p(g_i^{(l)}|\{\mathcal{D}_g^{(l)}\}_{-i}, \boldsymbol{\theta}) = \\ -\frac{1}{2}\sigma_{-i}^2 - \frac{1}{2}\frac{(g_i^{(l)} - \mu_{-i})^2}{2\sigma_{-i}^2} - \frac{1}{2}\log 2\pi, \tag{21}$$

where μ_{-i} and σ_{-i}^2 are the estimated mean and variance using the sample set $\{\mathcal{D}_g^{(l)}\}_{-i}$.

In our case, however, the actual gaze points of the eye images in the samples $\{\mathcal{D}_g^{(l)}\}$ also have a center bias for the same reason discussed in Section 2.2. This results in fewer samples in the peripheral region, and the errors in this region become overly small. When Eq. (20) is directly minimized, the optimization result also tends to be biased to the center. Therefore, we modify Eq. (20) to remove the center bias by normalizing the predictive probability as

$$L = \sum_{l=1}^{n_g}\sum_{i=1}^{M} \frac{1}{n_{(i,l)}} \log p(g_i^{(l)}|\{\mathcal{D}_g^{(l)}\}_{-i}, \boldsymbol{\theta}), \tag{22}$$

where $n_{(i,l)}$ is the total number of samples that have the same gaze points as $g_i^{(l)}$. By evaluating the average errors of each gaze point, Eq. (22) can evaluate the estimation errors in an unbiased manner on the display coordinates. Using partial derivatives with respect to the hyperparameters, Eq. (22) is maximized via a conjugate gradient method. The readers are referred to [30] for a detailed derivation of the partial derivatives.

2.4 Gaze Estimation

Once we have matrices $\boldsymbol{K}$, $\boldsymbol{S}$, and $\{\boldsymbol{G}^{(1)}, \ldots, \boldsymbol{G}^{(n_g)}\}$ in Eqs. (17) and (18), a gaze point can be estimated by taking a new eye image $\boldsymbol{e}$ as an input. The estimated distributions

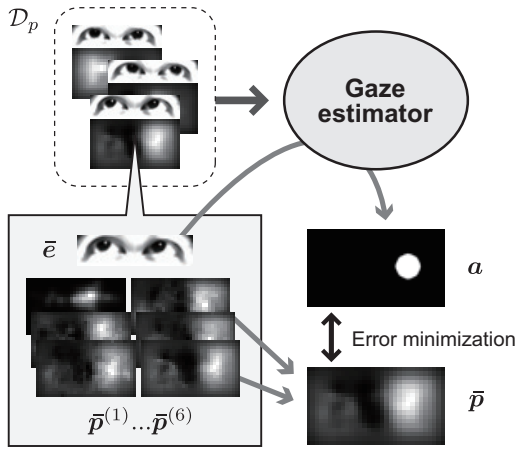

Fig. 6. Illustration of feature weight optimization. Target attention maps $\{a\}$ are generated based on leave-one-out estimates, and feature weights are optimized by minimizing a sum of squared residuals between target maps and sum maps $\{\bar{p}\}$.

for each X- and Y-direction, $\mathcal{N}(\mu_x, \sigma_x^2)$ and $\mathcal{N}(\mu_y, \sigma_y^2)$, are converted to the display coordinates $\mathcal{N}(\hat{\mu}_x, \hat{\sigma}_x^2)$ and $\mathcal{N}(\hat{\mu}_y, \hat{\sigma}_y^2)$ by

$$\hat{\mu}_x = x_o + \frac{W_I}{W_s}\mu_x, \quad \hat{\mu}_y = y_o + \frac{H_I}{H_s}\mu_y, \tag{23}$$

and

$$\hat{\sigma}_x^2 = \frac{W_I}{W_s}\sigma_x^2, \quad \hat{\sigma}_y^2 = \frac{H_I}{H_s}\sigma_y^2, \tag{24}$$

where W_s, H_s indicates the width and height of the saliency maps, W_I, H_I indicates the actual width and height of the displayed images $\{\boldsymbol{I}_1, \ldots, \boldsymbol{I}_N\}$, and (x_o, y_o) indicates the display origin of the images. The average $(\hat{\mu}_x, \hat{\mu}_y)$ corresponds to the estimated gaze point $\boldsymbol{g}$. The estimated variances are not directly used in our current system, however the estimation accuracy will be improved by incorporating techniques such as the Kalman filter as in [28].

3 Feature Weight Optimization

The previous section describes the complete pipeline of the training the gaze estimator and testing method. In the saliency aggregation step (Section 2.2), all six saliency features $\{\bar{\boldsymbol{p}}^{(f)}\}$ are independently aggregated, and the aggregated maps are linearly combined by Eq. (11) to produce the summed map $\bar{\boldsymbol{p}}$.

Although most existing methods use equal weights for simplicity [11], [14], [26], how to optimally tune the weights to achieve an even higher correlation between the visual saliency and gaze points remains unclear. Some studies use a data-driven learning approach to optimally tune the weight parameters using known gaze points [18], [19], [20], [21]. In our scenario, we do not have access to ground truth gaze points. Thus, the feature weights are refined in our method by using the estimated gaze points using the data-driven learning approach. The correlation between the combined saliency maps and gaze points is refined using this feedback loop.

Once the gaze estimator is built (Section 2.3), it can be used to estimate the gaze points from the associated average eye images $\bar{\boldsymbol{e}}$ in $\mathcal{D}_p$. Using this dataset, our method optimizes the feature weights so that the correlation between the peak of the gaze probability map $\bar{\boldsymbol{p}}$ and the gaze point estimate is higher.

Our approach is illustrated in Fig. 6. As described in Section 2.3, leave-one-out estimates $\{\mu_{-1}, \ldots, \mu_{-M}\}$ are obtained for each average eye image in both the X- and Y-directions. Using these estimated gaze coordinates, *target attention maps* $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M\}$, which represent the top-down gaze point distribution, are generated by drawing circles at the gaze points with a fixed radius. Since these leave-one-out estimates can include a large error, the radius is set to a relatively larger value (~ 4 degrees in our current implementation) than the central area of human vision (~ 1 degree). Using the target attention maps $\boldsymbol{a}$, the feature weights $\boldsymbol{\omega} = {}^t(\omega_1, \ldots, \omega_6)$ are optimized by minimizing the sum of the squared residuals as

$$\boldsymbol{\omega} = \arg\min_{\boldsymbol{\omega}} \sum_{i=1}^{M} ||\boldsymbol{a}_i - \sum_{f=1}^{6} \omega_f \bar{\boldsymbol{p}}_i^{(f)}||^2, \tag{25}$$

with a non-negativity constraint

$$\boldsymbol{\omega} \geq \boldsymbol{0}. \tag{26}$$

To reduce the number of equations in Eq. (25), n_a points are randomly sampled from both the positive and zero regions in all the attention maps $\boldsymbol{a}$. The matrix form of Eq. (25) can be written as

$$\boldsymbol{\omega} = \arg\min_{\boldsymbol{\omega}} \sum_{i=1}^{M} ||\boldsymbol{A} - \bar{\boldsymbol{P}}\boldsymbol{\omega}||^2, \tag{27}$$

where $\boldsymbol{A} \in \mathbb{R}^{2Mn_a \times 1}$ is a vector that consists of values at the selected points and $\bar{\boldsymbol{P}} \in \mathbb{R}^{2Mn_a \times 6}$ contains the corresponding feature values in each row. Eq. (27) is solved by using the non-negative least-squares algorithm of Lawson and Hanson [32] to obtain the optimal set of feature weights.

4 Experimental Results

In this section, we present our experimental results to evaluate our method. We use a set of 80 video sources in the experiments that are downloaded from the Vimeo website [33], which include various types of video clips, *e.g.*, music videos and short films. Some example frames are shown in Fig. 7. 30-second video sequences are randomly extracted from each video source without an audio signal, and resized to a fixed resolution of 960×540. These 80 short clips are divided into four datasets A, B, C, D, and seven novice test subjects $t_1, \ldots, t_7$ are asked to watch all of them. The video clips are shown at 25 fps; therefore, the number of frames is $N = 15000$ in each set, and the display resolution is set to $W_I = 1920$ and $H_I = 1080$. Although it highly depends on the algorithms and can be done prior

Fig. 7. Examples of video clips used in experiments. We use a set of 80 video clips downloaded from the Vimeo website [33]. All the pictures are licensed under a Creative Commons License.[2]

to the recording session of the eye images, the most time consuming part of the proposed framework is the saliency extraction step. For an efficient computation, the saliency maps are calculated at a smaller resolution, $W_s = 32$ and $H_s = 18$, in our experiments. One pixel corresponds to about 1.35×1.35 degrees in our current setting, which is close to the limit of the central area of human vision.

Throughout the experiment, the parameters are set at $n_s = 5$, $\tau_e = 0.45$, $T_s = -300$, $\alpha_e = 0.008$, $n_f = 3$, $\tau_f = 0.2$, $n_g = 1000$, $n_a = 40$, and τ_s is adaptively set to retain the top 15% of pixels and the remaining 85% are set to zero in each map. These parameter settings are empirically obtained from our experiment. In our current implementation, when $M \simeq 100$, it took about 1 minute for parameter optimization, and 1 millisecond per frame for estimation using a 3.33-GHz Core i7 CPU with a simple code parallelization using OpenMP [34].

4.1 Experiment Details

A chin rest is used during the experiments to fix the peoples' head positions, and a 23-inch Full HD (508.8×286.2 mm) display is placed about 630 mm away from the subject to show the video clips. A VGA-resolution camera is placed under the display to capture eye images.

We use the OMRON OKAO Vision library to detect the corners of each eye. We use template matching around the detected corners of each eye to ensure alignment accuracy. Small template images of the corners of each eye are registered during the initialization, and the locations with the highest normalized cross-correlation are used as the aligned corner positions. Based on the aligned positions as illustrated in Fig. 8 (a), the eye images are cropped to a fixed size of 70×35 pixels.

The eye images are histogram-equalized and pixels with intensity lower than the given threshold are truncated to zero so that images contain only eye regions (Fig. 8 (b)) to minimize effects caused by lighting changes. The threshold value is automatically decided using Otsu's method [35].

2. From top to bottom, left to right: "The Eyewriter" by Evan Roth (http://vimeo.com/6376466), "Balloons" by Javi Devitt (http://vimeo.com/10256420), "Tenniscoats - Baibaba Bimba — A Take Away Show" by La Blogotheque (http://vimeo.com/11046286), "Persona" by superhumanoids (http://vimeo.com/13848244), "MADRID LONGBOARD" by Juan Rayos (http://vimeo.com/12132621), and "MATATORO" by Matatoro Team (http://vimeo.com/13487624).

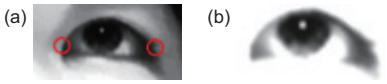

Fig. 8. Examples of eye images. (a) The eye images are cropped to a fixed size based on the detected positions. (b) The images are histogram-equalized and thresholded so that they contain only the iris and eye contours.

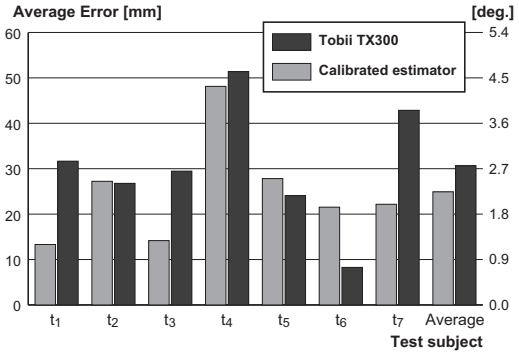

Fig. 9. Error comparison of commercial gaze estimator (Tobii TX300) and calibrated estimation method. A target point (the ground truth) is explicitly displayed to the test subjects, and the estimation accuracy is evaluated by assessing the deviation from the ground truth.

Finally, we apply a discrete Fourier transform to obtain the 4900-dimensional feature vectors $\boldsymbol{e}$, which consists of Fourier magnitudes of both the left and right eye images.

Ground truth.

We use a Tobii TX300 gaze tracker [36] to obtain the ground truth to quantitatively assess the effectiveness of our proposed method. The Tobii gaze tracker is placed within our setup and run in parallel with our method to obtain the ground truth gaze points. In addition, we also run a standard appearance-based gaze estimation method that uses an explicit calibration (in short, what we call a calibrated method hereafter) as a baseline method for further assessing our method. We show 16×9 reference points for each test subject at a regular interval on the display to train the estimator of the calibrated method. The eye images are recorded during the calibration to establish the mapping between the eye image and the gaze points. Once the pairs of reference gaze points and eye images are obtained, a learning process, which is the same manner as described in Section 2.3, is performed to obtain the mapping.

It is important to discuss the accuracy of the reference method that we use as the ground truth. The catalog specification of the accuracy of the Tobii TX300 is less than 1 degree. However, the error can be larger depending on the test subjects and installation conditions. We conduct a preliminary experiment using our setting to verify the actual accuracy of both the Tobii gaze tracker and the calibrated gaze estimator. A total of 120 target points are randomly

shown on the display to each of the seven test subjects. The subjects are asked to look at these target points, and we assess the gaze estimation accuracy using the target points as the ground truth gaze points. Fig. 9 shows the average distance errors between the ground truth target points and the estimated gaze points by the commercial and calibrated gaze estimators. As shown in the plot, the estimation accuracy depends on the subjects in our setting. The average errors are similar in these two approaches; it is about 30 mm ($\simeq$ 2.7 degrees) with the commercial estimator, and 25 mm ($\simeq$ 2.3 degrees) with the calibrated estimator. We observe that there is a fundamental limit in the accuracy evaluation of a gaze estimator from these experiments. As described above, we use the output of the commercial gaze estimator as the ground truth because of its availability to the readers and reproducibility in the experiments.

4.2 Gaze Estimation Result

We examine the performance of the proposed method by using the following procedure. First, we use the entire dataset for both training and testing to assess the upper-bound accuracy of the proposed method, *i.e.*, the training data performance. Second, we divide the dataset into two, one for training and the other for testing, to evaluate the generalizability of the proposed gaze estimator, *i.e.*, the test data performance.

Performance evaluation.

We first assess the performance, where the same dataset is used for both the training and testing. We perform this evaluation using four dataset A, B, C, D independently. The estimation results are summarized in Table 1. Each row corresponds to the result using dataset A, B, C, D, where all 20 video clips are used for both the training and testing. The first two columns indicate the AUCs of the average ROC curves of the raw saliency maps $\boldsymbol{s}$ and the gaze probability maps $\bar{\boldsymbol{p}}$. The remaining columns list the estimation errors of the proposed method and the calibrated appearance-based estimator for both the distance and angular errors. The errors are described by their (*average* $\pm$ *standard deviation*) form. The distance error is evaluated by using the Euclidean distance between the estimated and the ground truth gaze points, and the angular errors are computed using the distance between the eyes and the display.

Similarly, Table 2 lists the estimation error of each subject $t_1, \ldots, t_7$. Each row corresponds to the average of the results of the corresponding test subject using the four datasets A, B, C, D. The columns list the AUCs and estimation errors in the same manner as in Table 1. The overall average error is 39 mm ($\simeq$ 3.5 degrees). It can be seen from Table 1 and Table 2 that the performance does not have a strong dependency on the dataset and subjects. Although our method has a larger error than the calibrated estimator, our method can still achieve an accuracy of 3.5 degrees, which is sufficient for obtaining the regions of attention in images.

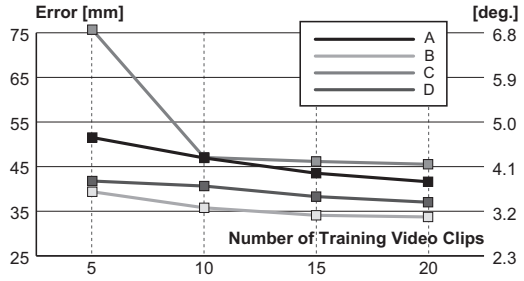

Fig. 10. Estimation errors w.r.t. different amounts of training video clips. Each point is plotted by choosing a limited number of learning video clips from the corresponding dataset, and the average error is computed from the results from the seven test subjects.

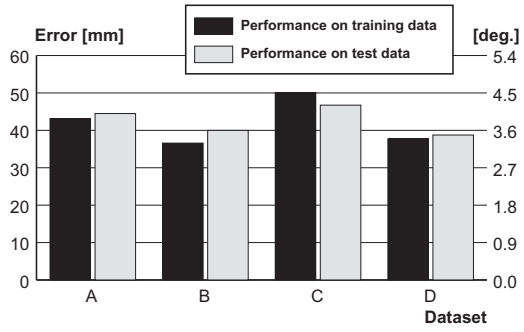

Fig. 11. Comparison of two estimation methods. All four datasets are divided into two subsets. One subset is used to train the gaze estimators, and the estimation accuracy is independently examined using each subset. The dark bars represent the training data performance, and the light bars represent the test data performance.

Performance variation w.r.t. the amount of training data.

We conduct a test using varying amounts of training video data to analyze the performance variations with respect to the amount of training data. Fig. 10 shows a comparison of the estimation errors for different amounts of training video clips. The X-axis represents the number of video clips used from each dataset. The Y-axis shows the average error that is computed from all seven test subjects. For this evaluation, five video clips are randomly selected from the training dataset and used as the test data. The result shows that the larger amount of training data results in a more accurate estimation result, though improvement slows after 10 training clips.

Performance on test data.

We use the test data sampled from the training dataset to verify the upper-bound on the accuracy in the above experiments. Our gaze estimator can also be applied to unseen video clips after training. We perform a test by separating the training and test datasets to evaluate the generalizability of the proposed method. Each of the four dataset $A, \ldots, D$ in this test are divided into two subsets

TABLE 1
Average error for each dataset. Two AUC columns show the AUCs of the average ROC curves of the raw saliency maps s and the gaze probability maps $\bar{p}$. The remaining columns are the distance and angular estimation errors (*average* $\pm$ *standard deviation*) when using the two estimation methods.

	s	$\bar{p}$	**Proposed method**		Calibrated method	
Dataset	AUC	AUC	error [mm]	error [deg.]	error [mm]	error [deg.]
A	0.80	0.92	41 $\pm$ 26	3.7 $\pm$ 2.3	33 $\pm$ 15	3.0 $\pm$ 1.4
B	0.83	0.95	36 $\pm$ 23	3.3 $\pm$ 2.1	24 $\pm$ 13	2.2 $\pm$ 1.2
C	0.81	0.94	41 $\pm$ 25	3.7 $\pm$ 2.3	27 $\pm$ 15	2.5 $\pm$ 1.4
D	0.83	0.91	36 $\pm$ 25	3.3 $\pm$ 2.3	34 $\pm$ 16	3.1 $\pm$ 1.5
Average	0.82	0.93	**39 $\pm$ 25**	**3.5 $\pm$ 2.3**	30 $\pm$ 15	2.7 $\pm$ 1.4

TABLE 2
Average error of each subject. The columns indicate the AUCs of the average ROC curves and estimation errors in the same manner as in Table 1.

	s	$\bar{p}$	**Proposed method**		Calibrated method	
Subject	AUC	AUC	error [mm]	error [deg.]	error [mm]	error [deg.]
t_1	0.80	0.93	41 $\pm$ 28	3.7 $\pm$ 2.6	29 $\pm$ 16	2.6 $\pm$ 1.4
t_2	0.79	0.92	41 $\pm$ 27	4.0 $\pm$ 2.4	30 $\pm$ 14	2.7 $\pm$ 1.3
t_3	0.83	0.93	33 $\pm$ 22	3.0 $\pm$ 2.0	34 $\pm$ 15	3.1 $\pm$ 1.4
t_4	0.81	0.94	42 $\pm$ 24	3.8 $\pm$ 2.2	30 $\pm$ 14	2.7 $\pm$ 1.3
t_5	0.84	0.92	35 $\pm$ 23	3.2 $\pm$ 2.1	27 $\pm$ 15	2.5 $\pm$ 1.4
t_6	0.83	0.94	36 $\pm$ 22	3.2 $\pm$ 2.0	24 $\pm$ 12	2.2 $\pm$ 1.1
t_7	0.82	0.90	39 $\pm$ 27	3.6 $\pm$ 2.5	33 $\pm$ 18	3.0 $\pm$ 1.7

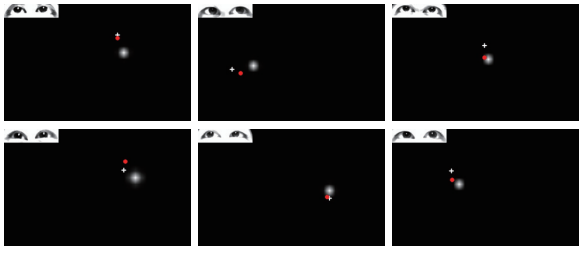

Fig. 12. Gaze estimation results. The estimation results of our method are rendered as 2-D Gaussian circles. The corresponding input eye images are shown at the top-left corner. The overlaid cross shapes represent the ground truth gaze points obtained by the commercial gaze tracker (Tobii TX300), and the solid circles indicate the gaze points obtained from the calibrated estimator.

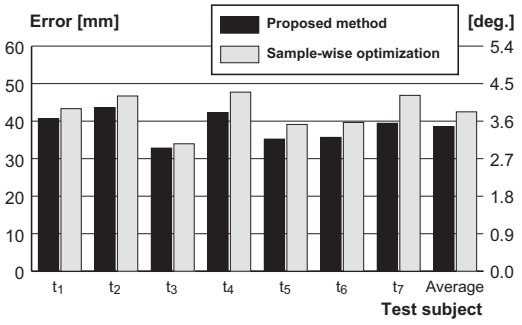

Fig. 13. Comparison of hyperparameters tuning method. The dark bars indicate the estimation errors after optimization using Eq. (22), and the light bars indicate the optimization results using Eq. (20). The proposed method without using Eq. (22) results in lower errors.

that consist of 10 video clips each. One subset is used to train our gaze estimator, and the other subset is used for testing. At the same time, we evaluate the performance using the original training data as the test data for a comparison. Fig. 11 shows the comparative results of these two estimation scenarios. The dark bars in the figure indicate the training data performance, and the light bars correspond to the test data performance. In most of the datasets, the training data performance showed the higher expected accuracy. However, the test data performance is comparable to it without any significant performance degradation.

Fig. 12 shows some examples of the gaze estimation results. The output of our method is rendered as a 2-D Gaussian circle centered at $(\hat{\mu}_x, \hat{\mu}_y)$ with a variance $(\hat{\sigma}_x^2, \hat{\sigma}_y^2)$ given by Eq. (19), and the mean coordinate $(\hat{\mu}_x, \hat{\mu}_y)$ is used as the estimated gaze point. The input eye images are shown at the top-left corner. The overlaid cross shape represents the ground truth gaze point obtained by the commercial gaze tracker, and the solid circle shows the gaze point estimated by the calibration-based estimator.

4.3 Effects of Parameter Optimization

In this section, we quantitatively evaluate the effectiveness of our parameter optimizations. We first assess the effectiveness of the hyperparameter tuning described in Section 2.3, and second, we evaluate the feature weight optimization described in Section 3.

Hyperparameter tuning results.

The effectiveness of the hyperparameter tuning of the Gaussian process regression is summarized in Fig. 13. The dark bars indicate the estimation errors after our

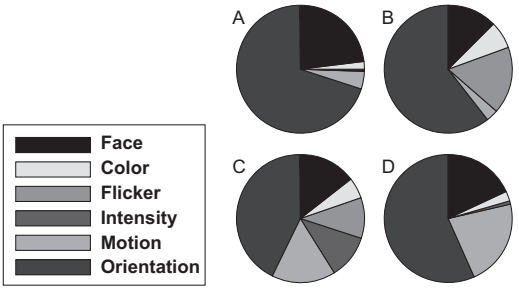

Fig. 14. Optimized weights of saliency features. The pie graph shows the ratio of the optimized feature weights averaged over all test subjects.

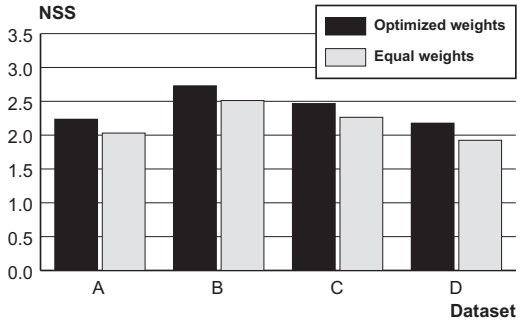

Fig. 15. Normalized Scanpath Saliency (NSS) scores of gaze probability maps. Dark bars indicate NSS values using optimized weights, and light bars indicate NSS values using initial equal weights. Optimized weights always outperform equal weights.

optimization using Eq. (22), and the light bars show the optimization results using Eq. (20), which corresponds to the standard formulation used to evaluate the sample-wise error. Our optimization method reduces the error by about 0.5 degrees.

Feature weight optimization results.

This experiment assesses the effectiveness of the feature weight optimization by our feedback loop, where the feature weight is updated to further enhance the accuracy of the gaze estimation. The original feature weights are all set to one. After feature weight optimization, the ratio of the feature weights is optimized, as shown in Fig. 14. The pie graph shows the ratio of the average weight computed from the data from all test subjects. After optimization, face and orientation features have greater weight compared with the others. This result is consistent with the report given by Zhao *et al*. [21], which optimized the feature weights using known gaze points.

We compute the Normalized Scanpath Saliency (NSS) [37] to evaluate the correlation between the gaze probability map and the estimated gaze point. The original definition of NSS is the normalized average of the saliency values at the fixation locations, *i.e*., the saliency maps are linearly normalized to have a zero mean and a unit standard deviation, and the NSS is computed as the average of the saliency values at the fixated positions. Therefore,

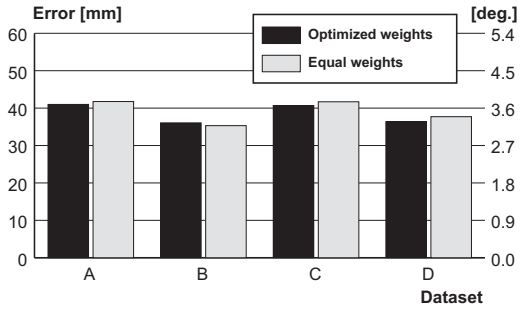

Fig. 16. Error comparison between before and after feature weight optimization. Optimized weights led to slightly reduced error for 3 of 4 datasets.

a higher NSS score indicates a greater correlation. In our case, instead of using the fixation locations, we use the true gaze points that are associated with the average eye images to compute the NSS. Given all the sets of average eye images and the gaze probability maps, we first compute the ground-truth gaze points. Since the average eye images are synthesized features as discussed in Section 2.2, we use the estimates from the calibrated method instead of the commercial gaze tracker. Using the ground-truth gaze points, we compute the NSS using the normalized gaze probability maps. Fig. 14 shows the summary of the results; dark bars indicate NSS values after optimization, and light bars indicate NSS values of initial equal weights. NSS scores are improved after optimization for all datasets.

Fig. 16 shows an error comparison between the before and after feature weight optimization. Dark bars indicate errors after feature weight optimization, and light bars represent errors when initial equal weights are used. While the improvement is rather small, the optimized weights yield a higher accuracy in 3 out of 4 datasets. One possible explanation for this small improvement would be that the aggregated feature maps $\bar{\boldsymbol{p}}^{(f)}$ are already sufficiently accurate for predicting gaze points, and the estimation accuracy is almost saturated in our setting. However, this improvement suggests that it is useful to incorporate the feedback loop for saliency optimization, especially when more complex saliency map models with many features are incorporated.

4.4 Spatial Bias in Error

The accuracy of our method depends on the distribution of the training samples. Fig. 17 shows the spatial distribution of the average estimation errors in the display coordinate. In the figure, each grid corresponds to the ground truth gaze location, and the magnitude and direction of the average estimation errors are rendered. In Fig. 17 (a), the higher intensity indicates a greater magnitude of estimation errors. In (b), the error directions are color coded, where each color corresponds to a certain direction that is illustrated in the reference circle, and a higher saturation indicates a greater error magnitude, like in Fig. 17 (a). There exists a

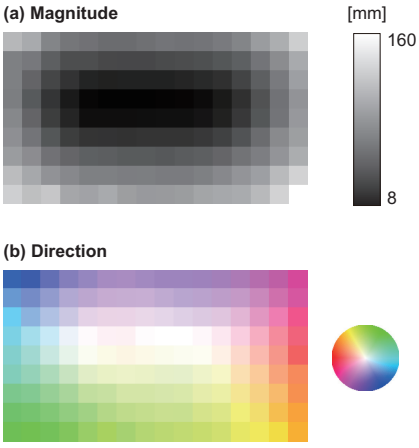

Fig. 17. Spatial distribution of estimation errors in display coordinate: (a) higher intensity corresponds to greater numbers of estimation errors, and (b) color wheel indicates error directions while saturation indicates error magnitudes.

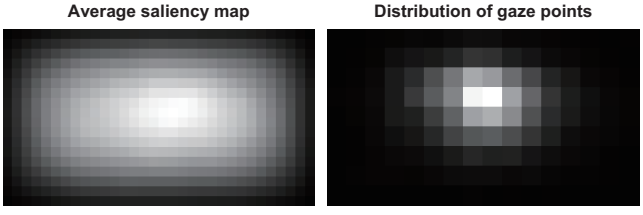

Fig. 18. Average saliency map and spatial histogram of gaze points. The image on the left shows the average of all the raw saliency maps extracted from the four video clips used in the experiment. The image on the right shows the spatial histogram of the ground truth gaze points from the experimental dataset. A higher intensity corresponds to larger counts of gaze points.

systematic bias, *i.e.*, large errors are observed around the peripheral areas, and the error directions are biased toward the center of the display.

Fig. 18 shows the average saliency map and spatial histogram of the gaze points. The image on the left shows the average of all raw saliency maps extracted from all datasets used in our experiment. The image on the right shows the spatial histogram of the ground truth gaze points obtained from the same datasets, in which the brighter intensity represents a higher frequency. Since salient objects are typically located near the centers of video frames, the average for the saliency maps is lower around the display boundary. The actual gaze points also tend to concentrate around the center of the display. As a result, the number of learning samples at the display edges is limited, and a bias in the estimation accuracy exists because of this.

5 Conclusion

We propose a novel gaze estimation framework in this paper that auto-calibrates by using saliency maps. Unlike the previous approaches that require an explicit calibration, our method automatically establishes the mapping from the eye image to the gaze point using video clips. Taking a synchronized set of eye images and video frames, our method trains the gaze estimator by regarding the saliency maps as the probabilistic distributions of the gaze points. In our experimental setting with fixed head positions, our method achieves an accuracy with about a 3.5-degree error.

We took an appearance-based gaze estimation approach. Appearance-based methods have a significant benefit in that they can be constructed using only a monocular camera without requiring a specialized hardware device. However, one of the biggest technical challenges common among existing appearance-based methods is the difficulty in handling the head pose movements. This is mainly due to the fact that allowing a head pose movement significantly expands the space of the training samples, and thus, the training becomes more difficult. There is some effort put forth to handle the head pose variations in an appearance-based setting [9], [38], and our future work includes adopting these techniques to allow head pose movement. In addition, as shown in [25], it is an alternative future direction to incorporate our approach in a model-based gaze estimation framework to further improve the estimation accuracy.

Naturally, the estimation accuracy of our method depends on the quality of the raw saliency maps extracted from the input video clips. The human gaze control mechanism is not yet completely understood, and there is a wide range of possibilities for more advanced saliency models to be used in our method to improve the gaze estimation accuracy.

Acknowledgments

This research was supported by CREST, JST.

References

[1] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 32, no. 3, pp. 478–500, 2010.

[2] R. J. K. Jacob, "Eye tracking in advanced interface design," in *Virtual environments and advanced interface design*, W. Barfield and T. A. Furness, Eds. Oxford University Press, 1995, pp. 258–288.

[3] T. Ohno, "One-point calibration gaze tracking method," in *Proceedings of the 2006 symposium on Eye tracking research & applications (ETRA '06)*, 2006, pp. 34–34.

[4] E. D. Guestrin and M. Eizenman, "Remote point-of-gaze estimation requiring a single-point calibration for applications with infants," in *Proceedings of the 2008 symposium on Eye tracking research & applications (ETRA '08)*, 2008, pp. 267–274.

[5] A. Villanueva and R. Cabeza, "A novel gaze estimation system with one calibration point," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 4, pp. 1123–1138, 2008.

[6] T. Nagamatsu, J. Kamahara, T. Iko, and N. Tanaka, "One-point calibration gaze tracking based on eyeball kinematics using stereo cameras," in *Proceedings of the 2008 symposium on Eye tracking research & applications (ETRA '08)*, 2008, pp. 95–98.

[7] E. D. Guestrin and M. Eizenman, "General theory of remote gaze estimation using the pupil center and corneal reflections," *Biomedical Engineering, IEEE Transactions on*, vol. 53, no. 6, pp. 1124 –1133, 2006.

[8] H. Yamazoe, A. Utsumi, T. Yonezawa, and S. Abe, "Remote gaze estimation with a single camera based on facial-feature tracking without special calibration actions," in *Proceedings of the 2008 symposium on Eye tracking research & applications (ETRA '08)*, 2008, pp. 245–250.

[9] Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike, "An incremental learning method for unconstrained gaze estimation," in *Proceedings of the 10th European Conference on Computer Vision (ECCV 2008)*, 2008, pp. 656–667.
[10] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," *Human neurobiology*, vol. 4, no. 4, pp. 219–227, 1985.
[11] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.
[12] C. Privitera and L. Stark, "Algorithms for defining visual regions-of-interest: Comparison with eye fixations," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 9, pp. 970–982, 2000.
[13] L. Itti and P. F. Baldi, "Bayesian surprise attracts human attention," in *Advances in Neural Information Processing Systems, Vol. 19 (NIPS 2005)*, 2006, pp. 547–554.
[14] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proceedings of Advances in neural information processing systems (NIPS 2007)*, vol. 19, 2007, pp. 545–552.
[15] N. D. B. Bruce and J. K. Tsotsos, "Saliency, attention, and visual search: An information theoretic approach," *Journal of Vision*, vol. 9, no. 3, pp. 1–24, 2009.
[16] D. Parkhurst, K. Law, and E. Niebur, "Modeling the role of salience in the allocation of overt visual attention," *Vision research*, vol. 42, no. 1, pp. 107–123, 2002.
[17] A. C. Schütz, D. I. Braun, and K. R. Gegenfurtner, "Eye movements and perception: A selective review," *Journal of Vision*, vol. 11, no. 5, 2011.
[18] W. Kienzle, F. A. Wichmann, B. Scholkopf, and M. O. Franz, "A nonparametric approach to bottom-up visual saliency," in *Proceedings of Advances in neural information processing systems (NIPS 2006)*, vol. 19, 2006, pp. 689–696.
[19] W. Kienzle, B. Scholkopf, F. A. Wichmann, and M. O. Franz, "How to find interesting locations in video: a spatiotemporal interest point detector learned from human eye movements," in *Proceedings of the 29th Annual Symposium of the German Association for Pattern Recognition (DAGM 2007)*, 2007, pp. 405–414.
[20] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proceedings of the Twelfth IEEE International Conference on Computer Vision (ICCV 2009)*, 2009.
[21] Q. Zhao and C. Koch, "Learning a saliency map using fixated locations in natural scenes," *Journal of Vision*, vol. 11, no. 3, 2011.
[22] E. Horvitz, C. Kadie, T. Paek, and D. Hovel, "Models of attention in computing and communication: from principles to applications," *Communications of the ACM*, vol. 46, no. 3, pp. 52–59, 2003.
[23] R. Vertegaal, J. Shell, D. Chen, and A. Mamuji, "Designing for augmented attention: Towards a framework for attentive user interfaces," *Computers in Human Behavior*, vol. 22, no. 4, pp. 771–789, 2006.
[24] Y. Sugano, Y. Matsushita, and Y. Sato, "Calibration-free gaze sensing using saliency maps," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2010)*, 2010, pp. 2667–2674.
[25] J. Chen and Q. Ji, "Probabilistic gaze estimation without active personal calibration," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2011)*, 2011.
[26] M. Cerf, J. Harel, W. Einhauser, and C. Koch, "Predicting human gaze using low-level saliency combined with face detection," in *Proceedings of Advances in neural information processing systems (NIPS 2008)*, vol. 20, 2008, pp. 241–248.
[27] OMRON OKAO Vsion library, https://www.omron.com/r_d/coretech/vision/okao.html.
[28] O. Williams, A. Blake, and R. Cipolla, "Sparse and semi-supervised visual mapping with the S^3GP," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2006)*, vol. 1, 2006, pp. 230–237.
[29] D. W. Hansen, J. S. Agustin, and A. Villanueva, "Homography normalization for robust gaze estimation in uncalibrated setups," in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications (ETRA '10)*, 2010, pp. 13–20.
[30] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. The MIT Press, 2006.
[31] S. Sundararajan and S. S. Keerthi, "Predictive approaches for choosing hyperparameters in gaussian processes," *Neural Computation*, vol. 13, no. 5, pp. 1103–1118, 2001.
[32] C. L. Lawson and R. J. Hanson, *Solving least squares problems*. Society for Industrial Mathematics, 1987.
[33] Vimeo, http://vimeo.com/.
[34] OpenMP, http://openmp.org/.
[35] N. Otsu, "A threshold selection method from gray-level histograms," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 9, no. 1, pp. 62 –66, 1979.
[36] Tobii Technology, http://www.tobii.com/.
[37] R. Peters, A. Iyer, L. Itti, and C. Koch, "Components of bottom-up gaze allocation in natural images," *Vision Research*, vol. 45, no. 18, pp. 2397–2416, 2005.
[38] F. Lu, Y. Sugano, O. Takahiro, and Y. Sato, "A head pose-free approach for appearance-based gaze estimation," in *Proceedings of the 22nd British Machine Vision Conference (BMVC 2011)*, 2011.

Yusuke Sugano received his B.S., M.S. and Ph.D. degrees in information science and technology from the University of Tokyo in 2005, 2007 and 2010 respectively. He is currently a Project Research Associate at Institute of Industrial Science, the University of Tokyo. His research interests include computer vision and human-computer interaction.

Yasuyuki Matsushita received his B.S., M.S. and Ph.D. degrees in EECS from the University of Tokyo in 1998, 2000, and 2003, respectively. He joined Microsoft Research Asia in April 2003. He is a Lead Researcher in Visual Computing Group of MSRA. His major areas of research are computer vision (photometric techniques, such as radiometric calibration, photometric stereo, shape-from-shading), computer graphics (image relighting, video analysis and synthesis). Dr. Matsushita is on the editorial board member of IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), International Journal of Computer Vision (IJCV), IPSJ Journal of Computer Vision and Applications (CVA), The Visual Computer Journal, and Encyclopedia of Computer Vision. He served/is serving as a Program Co-Chair of PSIVT 2010, 3DIMPVT 2011 and ACCV 2012. He is appointed as a Guest Associate Professor at Osaka University (April 2010-) and National Institute of Informatics, Japan (April 2011-). He is a senior member of IEEE.

Yoichi Sato is a professor at Institute of Industrial Science, the University of Tokyo. He received his B.S. degree from the University of Tokyo in 1990, and his M.S. and Ph.D. degrees in robotics from School of Computer Science, Carnegie Mellon University in 1993 and 1997 respectively. His research interests include physics-based vision, reflectance analysis, image-based modeling and rendering, and tracking and gesture analysis. He is currently on the Editorial Board of International Journal of Computer Vision, IPSJ Journal of Computer Vision and Applications, and IET Computer Vision. He also served as Associate Editor of IEEE Transactions on Pattern Analysis and Machine Intelligence. He is a Program Co-Chair of ECCV 2012.