# Learning-by-Synthesis for Appearance-based 3D Gaze Estimation

Yusuke Sugano
The University of Tokyo
Tokyo, Japan
sugano@iis.u-tokyo.ac.jp

Yasuyuki Matsushita
Microsoft Research Asia
Beijing, P. R. China
yasumat@microsoft.com

Yoichi Sato
The University of Tokyo
Tokyo, Japan
ysato@iis.u-tokyo.ac.jp

## Abstract

*Inferring human gaze from low-resolution eye images is still a challenging task despite its practical importance in many application scenarios. This paper presents a learning-by-synthesis approach to accurate image-based gaze estimation that is person- and head pose-independent. Unlike existing appearance-based methods that assume person-specific training data, we use a large amount of cross-subject training data to train a 3D gaze estimator. We collect the largest and fully calibrated multi-view gaze dataset and perform a 3D reconstruction in order to generate dense training data of eye images. By using the synthesized dataset to learn a random regression forest, we show that our method outperforms existing methods that use low-resolution eye images.*

## 1. Introduction

Gaze is an important cue to infer human attention and has been considered as a key factor to understanding internal states of humans. While gaze estimation techniques for wearable or short-distance ($\sim$ 60 cm) remote eye trackers are more or less mature technologies, it is still a challenging problem for mid- to far-distance scenarios due to unavailability of high-resolution eye images. Because in many practical scenarios, such as human-robot interaction, first-person vision, and digital signage systems, only low-resolution eye images are available, addressing the issues in low-resolution eye gaze estimation is essential. In this setting, *appearance-based* methods, which learn a mapping from eye images to gaze directions, have an advantage over *model-based* methods, which use geometrically derived eye features from high-resolution observations.

Appearance-based 3D gaze estimation is defined as a supervised regression task to predict a 3D gaze direction from an input feature, *i.e.*, a set of an eye image and a 3D head pose. The performance of appearance-based methods generally depend on the quality and diversity of training data and generalization ability of the regression algorithm. The current biggest limitation of many appearance-based methods is that person- and session-dependent training is always required. This is a significant disadvantage, and becomes a major factor of performance degradation especially when the head is moving. In fact, in most of previous studies, evaluation has been conducted using the test and training data of the same person, and it is unclear how they generalize to cross-subject training scenarios.

To address these problems, we propose a learning-by-synthesis approach to appearance-based gaze estimation using a large dataset that contains diverse people, head poses, and gaze directions. Images are recorded by a fully calibrated multi-camera system, and the synthesis of new appearances is performed via 3D reconstruction of eye regions and use of it for view warping. It enables to produce a large amount of training data as done in recent works on human body and head pose estimation [28, 8]. Using the synthesized dataset, the gaze estimator is trained by an extension of random forests [3], where a set of regression trees are learned with *redundant* subsets of the training data. The redundancy aims at fully utilizing the nature of the mixed-modal input and contributes to improve the estimation accuracy. With our method, appearance-based gaze estimation can become more effective and reliable than previous methods as we will see in the experiments.

Our contributions can be summarized as follows: 1) The largest multi-view gaze dataset with full 3D annotations, 2) The learning-by-synthesis approach to appearance-based gaze estimation, and 3) The best accuracy in person- and pose-independent, calibration-free gaze estimation from low-resolution images.

## 2. Related work

There are two categories of gaze estimation techniques, *i.e.*, model-based and appearance-based approaches [15]. There have been several methods proposed in each category that learn a direct 2D mapping from eye features to gaze positions on the target screen; however, they have a critical disadvantage that they implicitly assume pre-defined and static eye and head positions. In contrast, 3D gaze es-

timation methods infer gaze directions in the form of a 3D vector spanned from the eye center and thus are able to naturally handle head pose variations. Our method falls into the category of 3D gaze estimation, and in what follows, we briefly discuss the previous approaches in this domain.

**Model-based gaze estimation:** Model-based 3D gaze estimation methods use 3D eyeball models and estimate the gaze direction using geometric eye features [14, 6, 24]. They typically use infrared light sources together with a high-resolution camera to locate the 3D eyeball position and its line of sight via personal calibration. Although this approach can accurately estimate gaze directions, its requirement of specialized hardware limits its application. There are methods that relax this requirement and use only eye images for determining the line of sight from, *e.g.*, the iris contour [16, 5, 36]. These are effective in short distance scenarios where high-resolution observations are available; however, their effectiveness in mid-distance scenarios is unclear.

**Appearance-based gaze estimation:** Unlike model-based methods, appearance-based methods compute non-geometric image features from the input eye images and estimate gaze directions. This approach casts the gaze estimation problem to learning a mapping function from eye images to gaze directions. Such a mapping function can be learned using various regression techniques, including neural networks [2, 35], local interpolation [31, 19], or Gaussian process regression [34, 30].

For appearance-based 3D gaze estimation, the 3D position of the eye has to be determined in order to estimate the gaze target in the world coordinate system. With the recent advancement of monocular 3D head pose tracking [23] and the increasing availability of RGB-D cameras with head pose tracking [4], the means of capturing 3D head poses are becoming readily available. Indeed, recent appearance-based 3D gaze estimation methods use 3D head poses obtained as an additional input for gaze estimation [21, 20, 9]. Appearance variations of the eye images caused by head pose changes is another technical challenge, and in these methods they are handled by learning an additional compensation function [21], or by warping training images to new head poses [20, 9].

While most of these previous studies used person-dependent training dataset, Funes *et al.* presented a cross-subject training method for appearance-based gaze estimation [10]. Following their previous work [9], they used an RGB-D camera to warp training and test images to the frontal view and used the adaptive linear regression method [20] to estimate 3D gaze directions. Under the cross-subject training scenario (five subjects), they reported a mean error larger than 10 degrees. The goal of our method is similar to theirs, but we use a learning-by-synthesis approach using a significantly larger ($10\times$) dataset with a random forest-based learning algorithm that reduces the error by $50\%$ from their work.

| | Subjects | Gazes | Head Poses | Images |
|---|---|---|---|---|
| Weidenbacher *et al.* [32] | 20 | 2 – 9 | 19 | 2,220 |
| Ponz *et al.* [25] | 103 | 12 | 1 | 1,236 |
| McMurrough *et al.* [22] | 20 | 16 | 1 | (Videos) |
| Smith *et al.* [29] | 56 | 21 | 5 | 5,880 |
| Ours | 50 | 160 | 8 | 64,000 |

Table 1: Comparison of gaze dataset sizes. From left to right, the number of subjects, gaze targets, head poses, and the total number of observations are shown.

## 3. Multi-view gaze dataset

This section describes our multi-view gaze dataset[1]. For the purpose of learning a person- and pose-independent gaze regression function, the training dataset must contain a large number of subjects, head poses, and gaze directions.

As summarized in Tab. 1, recently published datasets have relatively large numbers of participants, *e.g.*, $103$ in [25] and $54$ in [29]. However, their sampling density of gaze directions is rather limited (at most $21$ directions per head pose [29]) because their purpose is for learning coarse gaze classifiers. Head pose variation is also limited, *e.g.*, at most $19$ different poses in [32]. Our dataset is designed to address these sampling issues and has $64,000$ images ($= 50$ subjects $\times 8$ views $\times 160$ gaze directions), while the largest dataset in the literature [29] is limited to $5,880$ images in total. This density allows us to recover 3D shape of eye regions for synthetically increasing the number of observations as we will see in the next section.

Another important aspect for a gaze dataset is that the ground-truth 3D gaze directions need to be provided in the 3D world coordinate system. In addition, precise 3D positions of both eyes and gaze targets in each camera coordinate system should be provided as an annotation, and the coordinate system must be consistent across subjects. In our dataset, eight cameras and the gaze target plane are fully calibrated, and all annotations are provided in the 3D world coordinate system.

### 3.1. Data collection

Figure 1 shows the setup of our data collection system. Eight $1.3$ megapixel color cameras, PointGrey Flea3 USB 3.0 with a $8$ mm fixed focal-length lens, are attached to the frame of a 22-inch WUXGA (473.8 mm $\times$ 296.1 mm)

[1]The dataset is available at `http://www.hci.iis.u-tokyo.ac.jp/datasets/`

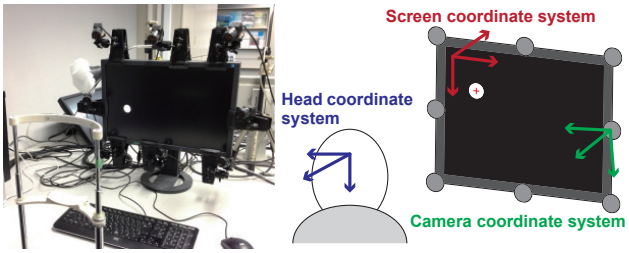



Figure 1: System configuration for data collection

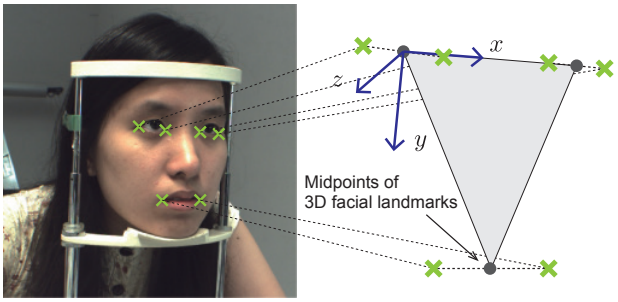



Figure 2: Definition of head pose. The head coordinate system is defined based on a triangle connecting three midpoints of the eyes and mouth.

LCD monitor, and these cameras capture images in a synchronized manner via a software trigger controlled by the host computer. Intrinsic and extrinsic camera parameters are calibrated beforehand, and the 3D position of the monitor plane in the camera coordinate system is also calibrated using mirrored calibration patterns displayed on the monitor [26].

A total of 50 (15 female and 35 male) people ranging in age approximately from 20 to 40 years old participated in the data collection. A chin rest was used to stabilize the head position located at 60 cm apart from the monitor. During recording sessions, participants were instructed to look at a visual target displayed on the monitor. As illustrated in Fig. 1, the target was a white circle with a red cross at its center on the black background. The screen was divided into a $16 \times 10$ regular grid, and the visual target moved to the center of each grid in a random order. The white circle shrank after the target stops at each position, and cameras were triggered at the time the circle disappeared. As a result, $G = 160$ (gaze directions) $\times 8$ (cameras) images were acquired from each participant at SXGA resolution, together with the 3D positions of the visual targets. The gaze directions spanned approximately $\pm 25$ degrees horizontally and $\pm 15$ degrees vertically, and this covered the range of natural gaze directions [1].

### 3.2. Facial landmark annotation

The captured images are further annotated with facial landmarks. The locations of six facial landmarks, corners of the eyes and mouth, are manually annotated using the first eight images for each subject, and their 3D positions are recovered. Since there is a slight possibility that participants moved their head during the recording session, these landmark positions are refined frame-by-frame via a simple multi-view template matching as followings. The annotated 3D positions are projected back to the next set of images, and a template-matching search is performed around the projected position to find correct 2D and 3D facial landmark locations. In this manner, the 3D positions of the six facial landmarks are estimated for each gaze direction.

These 3D landmark positions are used to define head poses of the subjects. As illustrated in Fig. 2, the head coordinate system is defined on the basis of a triangle connecting three midpoints of the eyes and mouth. The $x$-axis of the head coordinate system is defined as the direction from the right eye center to the left eye center, and the $z$-axis is defined as the perpendicular direction from the triangle plane towards the back of the subject.

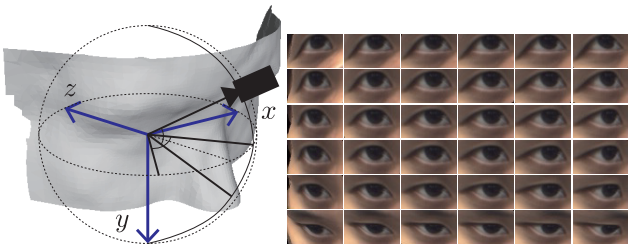
## 4. Learning-by-synthesis for gaze estimation

Using our multi-view dataset that consists of $G$ gaze directions of $N$ people, we take a learning-by-synthesis approach to person- and pose-independent gaze estimation. Our method consists of three steps: 1) reconstructing the 3D shape of eye regions from the multi-view gaze dataset, 2) synthesizing eye images from dense viewing angles, and 3) learning an appearance-based gaze estimator.

Given a set of training samples $\{((\boldsymbol{e}_i, \boldsymbol{p}_i), \boldsymbol{g}_i)\}$, where $\boldsymbol{g}_i$ is a gaze direction vector of the $i$-th sample and $(\boldsymbol{e}_i, \boldsymbol{p}_i)$ is its concatenated feature vector of an eye image $\boldsymbol{e}_i$ and head pose $\boldsymbol{p}_i$, our goal is to learn a regression function that predicts a 3D gaze direction $\boldsymbol{g}^*$ from the input feature $(\boldsymbol{e}^*, \boldsymbol{p}^*)$. In our method, eye images are converted into grayscale, histogram-equalized, and raster-scanned to form a feature vector $\boldsymbol{e}$. The gaze direction $\boldsymbol{g}$ is defined as a 2D polar angle vector in the camera coordinate system, and the head pose vector $\boldsymbol{p}$ consists of the 3D position of the eye midpoint and the 3D head rotation. To be exact, the initial point of the 3D gaze direction vector should be at the position of the eyeball center. However, since it is difficult to know the 3D eyeball position from low-resolution images without personal calibration, we approximate the initial point as the midpoint of the eye corners in this work.

Our method increases the number of training samples by synthesis, and there are two advantages to synthesizing training data. First, it makes the sampling in the head pose space denser, and thus gaze estimation from diverse head poses $\boldsymbol{p}^*$ is enabled. Second, the synthesized data can also

Figure 3: Examples of reconstructed 3D models. The left image shows a point cloud obtained with [11], and the right image shows examples of the rendered 3D eye regions.




(a) Virtual camera (b) Synthesized images

Figure 4: Training data synthesis: (a) Placement of the virtual cameras, (b) Examples of synthesized eye images.

increase the information beneficial for learning the relationship between $\boldsymbol{e}$ and $\boldsymbol{g}$, and it indeed improves the estimation accuracy as we will see in the experiment. This is because the gaze direction $\boldsymbol{g}$ is defined in the camera coordinate system and the appearance of the eye image $\boldsymbol{e}$ is directly correlated with $\boldsymbol{g}$ regardless of the head pose $\boldsymbol{p}$. For example, if two eye images have similar gaze directions in the camera coordinate system, the position of the iris contour should appear at similar positions in these images.

### 4.1. 3D reconstruction of eye regions

Our method first synthesizes dense multi-view eye images by recovering the 3D shape of eye regions. We use a patch-based multi-view stereo algorithm [11] to reconstruct the 3D shapes from 8 multi-view images. The reconstructed 3D point cloud is further processed by using statistical outlier removal [27] and moving least squares smoothing [18], and a cuboid region (16 cm $\times$ 8 cm $\times$ 6 cm of the eye region) predefined in the head coordinate system is cropped. We then use a Poisson reconstruction method [17] to reconstruct the 3D mesh of the eye region. The texture of the 3D mesh is finally computed using the mean of all source images with visibility checking. Figure 3 shows the examples of reconstructed 3D eye regions that convey faithful appearance.

### 4.2. Training data synthesis

The purpose of the data synthesis is to increase the variation coverage of the 6D head pose $\boldsymbol{p}$. However, without loss of generality, the required training space can also be reduced to 2D polar coordinates $\boldsymbol{r}$, *i.e.*, positions of the virtual camera on a viewing sphere around the eye position (Fig. 4a). In other words, it is only required to synthesize training samples $\{((\boldsymbol{e}_i, \boldsymbol{r}_i), \boldsymbol{g}_i)\}$ in the reduced 2D space and learn the regression function $\boldsymbol{g} = F_r(\boldsymbol{e}, \boldsymbol{r})$, because the head pose $\boldsymbol{p}$ and the input image can be explicitly converted into an equivalent set of a 2D polar coordinates $\boldsymbol{r}$ and an eye image $\boldsymbol{e}$ as follows.

Let $\boldsymbol{t}$ be the 3D position of the eye midpoint corresponding to the original head pose $\boldsymbol{p}$, and $\boldsymbol{R}$ be the head rotation matrix. We need to convert them to the polar coordinates $\boldsymbol{r}$ in a way that the equivalent conversion can be also done for the eye image $\boldsymbol{e}$. Given the radius $d_s$ of the viewing sphere, we can compute such a conversion matrix $\boldsymbol{M} = \boldsymbol{S}_c\boldsymbol{R}_c$ that maps the camera position onto the viewing sphere. $\boldsymbol{S}_c = \mathrm{diag}(1, 1, d_s/\|\boldsymbol{t}\|)$ is a $z$-direction scaling matrix, and $\boldsymbol{R}_c$ is the inverse of the rotation matrix that rotates the camera to look at $\boldsymbol{t}$ and make the $x$-axes of both the camera and head coordinate systems parallel. The converted rotation matrix $\hat{\boldsymbol{R}} = \boldsymbol{M}\boldsymbol{R}$ tells us the corresponding 2D polar coordinates $\boldsymbol{r}$. Then, given the projection matrix of the virtual camera $\boldsymbol{C}_s$, an equivalent image transformation matrix can be obtained as $\boldsymbol{W} = \boldsymbol{C}_s\boldsymbol{M}\boldsymbol{C}_r^{-1}$, where $\boldsymbol{C}_r$ is the projection matrix of the real camera.

Eye images are synthesized in the range of viewing angles around the eye position where the eye is observable. As shown in Fig. 4a, the range is 66 degrees horizontally (30 degrees upward and 36 degrees downward) and vertically (30 degrees inward and 36 degrees outward). The angle range is divided into 6-degree intervals, and eye images are synthesized at a total of $P$ (=144) view positions.

Figure 4b shows examples of the synthesized eye images. Each image is rendered with a predefined image size $W \times H$, and in total $G \times P$ eye images are synthesized for each eye of each subject.

## 5. Random regression forests with redundancy

We use a method based on random forests [3] to learn the regression function because it can handle a large-scale regression problem like ours with a low computational cost in the testing phase.

In our problem setting, the input feature consists of multiple modalities, the appearance and the pose of an eye, which are both closely correlated with the output variable, 3D gaze direction. A similar problem setting in a different context has been studied for facial feature detection with a varying head pose [7]. In their method, random forests are learned independently on subsets of training data. Training

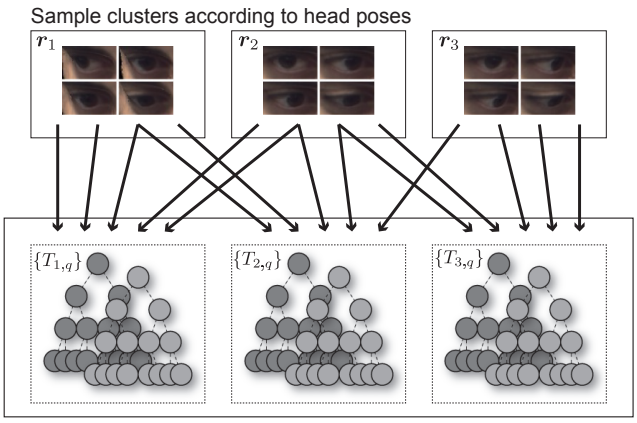



Figure 5: Structure of the proposed regression forests. Instead of splitting the regression problem into a set of pose-dependent estimation tasks, a set of regression trees with different but overlapping head pose ranges is learned.

samples are clustered based on their head poses, and an output is derived using a conditional probability given a head pose estimated by another set of decision trees.

In our case, however, it is not an optimal strategy to split this regression problem into a set of pose-dependent estimation tasks because the relationship between head pose and eye appearance is not totally conditional. As discussed in the previous section, appearance of the eye images is directly correlated with the gaze direction regardless of their corresponding head poses. At the same time, gaze directions are distributed within a limited range around the head direction, which indicates the correlation between gaze directions and head poses. Therefore, we take an approach of learning random forests with some redundancy of head poses.

Figure 5 illustrates the overall structure of our *redundant* random forests. Based on the synthesized $P$ head poses $\boldsymbol{r}$, we cluster training samples into $P$ pose clusters, where each cluster contains $2 \times N \times G$ samples of the same head pose. Instead of directly learning image-based regression functions $\boldsymbol{g} = F(\boldsymbol{e})$ for each cluster, we create redundant subsets of the training data to learn $P$ joint regression functions $\boldsymbol{g} = F_r(\boldsymbol{e}, \boldsymbol{r})$. Namely, to learn the $p$-th regression function $F_r$ corresponding to the $p$-th head pose $\boldsymbol{r}_p$, we randomly select $S \ll 2 \times N \times G$ training samples from each of the $R$-nearest sample clusters in the head pose space. The head pose $\boldsymbol{r}_p$ is only used to select training samples, and a random regression forest is built using the selected $R \times S$ random samples.

As a whole, the overall structure of the learned estimator is still a simple ensemble of regression trees whose associated head poses $\boldsymbol{r}_p$ are different. In the testing phase, the input feature is queried to $R$ regression forests corresponding to $R$ nearest head pose cluster centers for the input head pose $\boldsymbol{r}$, and their mean is taken as the output.

### 5.1. Training and testing

In the training phase, each regression function $\boldsymbol{g} = F_r(\boldsymbol{e}, \boldsymbol{r})$ is learned as a set of $Q$ binary regression trees $\{T_{p,q}\}_{q=1}^{Q}$ as in the original random forest algorithm [3], with modifications to handle the mixed-modal inputs. To build each regression tree $T_{p,q}$, a random subset of $R \times S$ training samples is first created, and the regression tree is grown so that each node splits the training samples so as to minimize the mean squared error among them. The binary split at each node is made by comparing the feature value $f$ to a threshold $\tau$. Namely, samples are divided into two child nodes $\mathcal{L} = \{((\boldsymbol{e}, \boldsymbol{r}), \boldsymbol{g}) | f < \tau\}$ and $\mathcal{R} = \{((\boldsymbol{e}, \boldsymbol{r}), \boldsymbol{g}) | f \geq \tau\}$. Hence, the goal of the training process becomes determining the optimal set of splitting features $f$ and their thresholds $\tau$ at the tree nodes.

Specifically, candidate features for splitting are randomly chosen from two functions, the intensity difference $f_e$ and pose value $f_r$:

$$f_e(\boldsymbol{e}; s, t) = e_s - e_t, \quad (1)$$

$$f_r(\boldsymbol{r}; i) = r_i, \quad (2)$$

where $e_s$ and $e_t$ indicate intensities at the $s$-th and $t$-th pixels of $\boldsymbol{e}$, and $r_i$ indicates the $i$-th element of the vector $\boldsymbol{r}$. At each node, the best threshold $\tau$ for splitting is examined for each of a random subset of candidate features. In our case, since the number of possible candidates becomes significantly larger for Eq. (1) ($= {}_{W \times H}C_2$) than Eq. (2) (= 2), the two candidates of Eq. (2) are always examined together with a random subset of Eq. (1), where the subset size is a squared root of the number of features [3]. From these candidates, the feature $f$ with its best threshold $\tau$ that maximizes the gain in terms of the mean squared error of the gaze direction $\boldsymbol{g}$ is selected. The growth of the tree is stopped when each node contains only one sample, and the leaf nodes store gaze direction labels $\boldsymbol{g}$.

Each regression forest $\{T_{p,q}\}$ is stored together with the head pose $\boldsymbol{r}_p$, and in total, there are $P \times Q$ regression trees built. The test input $(\boldsymbol{e}^*, \boldsymbol{r}^*)$ is queried to its $R$-nearest regression forests in terms of the distance between $\boldsymbol{r}^*$ and $\boldsymbol{r}_p$. Then, the output gaze direction $\boldsymbol{g}^*$ can be computed as a mean across all trees of the $R$ regression forests, *i.e.*, the value stored in a reached leaf node.

## 6. Experiments

This section evaluates the performance of the proposed method. We use synthesized images for training, and recorded images for testing. Namely, eye images and head poses for test data were extracted from each of $8 \times G$ images using the conversion process described in Sec. 4.2. Figure 6

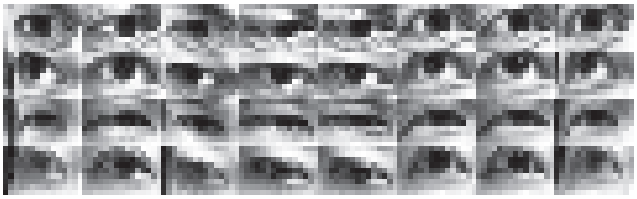


Figure 6: Example of test images

shows some examples of the actual test images. The image size $(W, H)$ was set to $(15, 9)$ for both training and testing. Moreover, the camera angles were distant from the frontal direction, and this made it more difficult to detect iris edges.

Because eyes are not exactly symmetrical, appearance-based gaze estimators are usually learned independently for each of left and right eyes. However, such a difference becomes trivial in a cross-subject training scenario where diverse appearances are learned at once, and there is no need to treat the left and right eyes separately. Hence, we swapped the eye image horizontally and mirrored the pose $\boldsymbol{r}$ and gaze $\boldsymbol{g}$ so that both eyes could be handled by a single regression function. In order to reduce the memory requirement and increase the efficiency of the feature selection in random forests, we also restricted the candidate pixel pairs for Eq. (1). Variable importances that were obtained through the training using all features were evaluated with respect to the distance between pixel pairs. Since most of the important pairs had distances shorter than a certain threshold (6 pixels in our case), the candidate pairs in Eq. (1) were restricted to a subset whose lengths were less than the threshold.

### 6.1. Comparison with baseline methods

We compared our method with two baseline methods. The first method is ALR (adaptive linear regression) [19] which was one of very few prior methods tested in the cross-subject training scenario [10]. $k$-nearest neighbor ($k = 10$) is selected as the second method because of its real-time estimation capability, which is crucial for various gaze applications.

Figure 7 shows the mean estimation errors of all $50$ participants for within-subject and cross-subject training. Within-subject errors are evaluated using the target subject's own synthesized training data, and cross-subject errors are evaluated using three-fold cross validation using synthesized training data of 33 different subjects. However, since ALR requires to solve an L1 optimization problem for each input data, it takes a prohibitively large amount of time for evaluation with this size of training dataset. Hence, we reduced the number of training subjects to five by taking a similar approach to [10] [2].

[2] ALR finds a sparse set of training data for interpolation, and hence interpolation weights are almost zero for most of the training subjects. The five training subjects with highest interpolation weights were selected through preliminary tests.

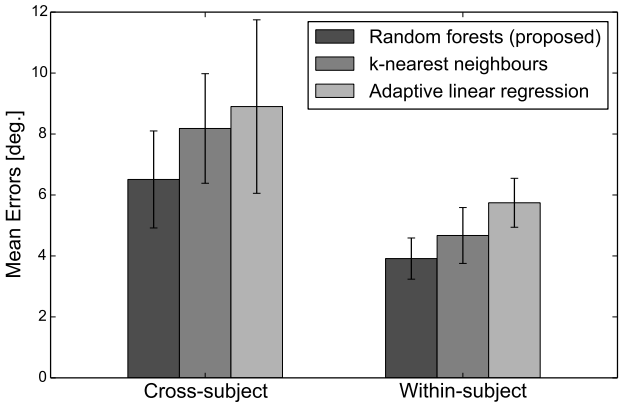


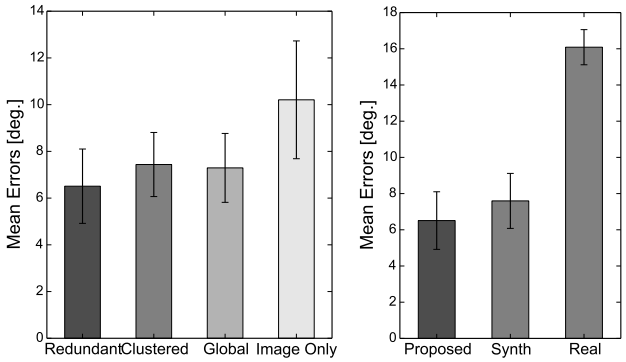

Figure 7: Comparison with baseline methods. Mean estimation errors of the proposed method and two baseline methods are shown for within-subject and three-fold cross-subject training. Error bars indicate standard deviations.

The number of neighbors $R$ in the redundant regression forests was set to five. The number of regression trees $Q$ in each forest was set to ten, with $S$ set to half of the total number of redundant samples. For the $k$-NN estimation, estimators were built for each of $P$ head poses independently, and the mean output of the same $R$-nearest estimators was taken as the output. A similar procedure was taken for ALR; however, the number of nearest estimators was set to three because increasing the number did not improve the accuracy and it took significantly longer for testing (approximately $0.5$ seconds per estimator with our MATLAB implementation using CVX [13, 12]). The proposed method was implemented in C++, and it took less than $1$ millisecond per input.

Although the accuracy of the ALR method was even higher than the value reported in [10], $k$-NN approach achieved greater accuracy in our problem setting because of densely synthesized training samples. The proposed method further improved the accuracy and achieved the lowest error with both within-subject and cross-subject training (paired Wilcoxon test [33], $p < 0.01$). Due to the approximated eyeball center position and the offset ($\sim 3$ degrees) between the optical and visual axes, the lower limit of the accuracy is much higher for cross-subject training. The mean error of our method with cross-subject training was $6.5 \pm 1.5$ degrees.

### 6.2. Comparison of random forests structures

Now let us assess the effectiveness of the proposed method, *i.e.*, redundant regression forests with synthesized




(a) Comparison of random forests (b) Effect of synthetic data

Figure 8: Evaluation of the effectiveness of our method under the three-fold cross-subject training: (a) comparison of four different types of random regression forests structures, and (b) comparison of regression forests learned with and without synthesized data.

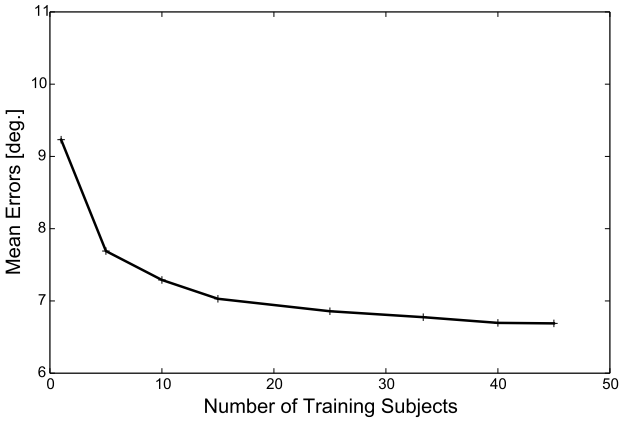



Figure 9: Estimation accuracy with respect to training data size. The horizontal axis indicates the number of training subjects, and the vertical axis indicates the mean estimation accuracy of the proposed method.

training data. Figure 8a shows mean accuracy of the same three-fold cross-subject tests for four different types of random regression forests. The leftmost bar corresponds to the proposed redundant forests, and the next bar corresponds to the case when regression forests are learned independently for each pose cluster. The third bar corresponds to the case when a single regression forest is learned using all training data. The last bar shows the mean error of the same global regression forest without using the head pose $\boldsymbol{r}$ as input. As can be seen by comparing the global and image-only cases, random forests can learn the mixed-modal regression function, and the global case shows comparative performance to the clustered case. The result shows that our redundant forests further improves performance (paired Wilcoxon test, $p < 0.01$).

Figure 8b shows an additional comparison between regression forests learned with and without synthesized data. The biggest advantage of using synthesized training data is that it enables the gaze estimator to handle head poses that are not contained in the training data. If the cross-subject test is performed using the real data, however, eye images captured from the same camera are always contained in the training dataset. Hence, we conduct experiments by excluding the nearest forest, which is trained using the data captured by the same camera as the test image, in each test.

The right bar in Figure 8b corresponds to the test with three-fold cross-subject training using the real eye images ($8 \times G$ per subject, equivalent to the test data), and the middle bar corresponds to the same test using the synthesized ($P \times G$ per subject) data. Since the training head poses are sparsely distributed in the real dataset, in these two cases the estimators are built independently for each pose cluster as in the second case shown in Fig. 8a. The leftmost bar corresponds to the proposed redundant forests. As expected, these graphs clearly show that the performance is significantly improved if the training dataset contains similar head poses to the input data. This problem can be efficiently addressed by using the synthesized training data, and the performance is further improved by utilizing the dense structure of the synthesized data.

### 6.3. Effect of dataset size

In this section, we evaluate the performance variation with varying dataset size. Figure 9 shows mean accuracy with respect to the number of training subjects. Due to memory constraints in our execution environment, raw intensity values were used instead of the intensity difference (Eq. (1)), and hence, the overall accuracy is slightly worse than the result shown in Fig. 7. In the figure, although the accuracy improvement becomes smaller at around 33 subjects, *i.e.*, the case of three-fold cross validation discussed above, it does not apparently converge even with 46 subjects (paired Wilcoxon test, $p = 0.03$). This result suggests the potential of achieving even greater accuracy by using a larger amount of training data.

## 7. Conclusion

We presented an appearance-based, person- and head pose-independent gaze estimation technique. In this technique, the gaze estimator is learned with random regression forests using a large amount of synthesized training data. Owing to the synthesized dataset, the estimation accuracy is significantly improved from the prior work, and

the learned estimator can estimate gaze directions for arbitrary head poses that are not contained in the original data. To the best of our knowledge, this is the first attempt to use the learning-by-synthesis approach in the context of appearance-based gaze estimation.

Our multi-view gaze dataset will be made publicly available for future researches. Since it has full 3D annotations and 3D reconstruction results, the applications of the dataset are not limited to our problem setting. Applications to eye alignment and tracking will be our important future work.

## Acknowledgment

This work was supported by CREST, JST.

## References

[1] A. T. Bahill, D. Adler, and L. Stark. Most naturally occurring human saccades have magnitudes of 15 degrees or less. *Investigative Ophthalmology & Visual Science*, 14(6):468–9, 1975.

[2] S. Baluja and D. Pomerleau. Non-intrusive gaze tracking using artificial neural networks. Technical Report CMU-CS-94-102, School of Computer Science, Carnegie Mellon University, 1994.

[3] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Q. Cai, D. Gallup, C. Zhang, and Z. Zhang. 3d deformable face tracking with a commodity depth camera. In *Proc. ECCV*, pages 229–242, 2010.

[5] J. Chen and Q. Ji. 3d gaze estimation with a single camera without IR illumination. In *Proc. ICPR*, 2008.

[6] J. Chen and Q. Ji. Probabilistic gaze estimation without active personal calibration. In *Proc. CVPR*, pages 609–616, 2011.

[7] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool. Real-time facial feature detection using conditional regression forests. In *Proc. CVPR*, pages 2578–2585, 2012.

[8] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool. Random forests for real time 3d face analysis. *Int'l Journal of Computer Vision*, 101(3):437–458, 2013.

[9] K. A. Funes Mora and J.-M. Odobez. Gaze estimation from multimodal kinect data. In *Proc. CVPR2012 Workshop on Gesture Recognition*, pages 25–30, 2012.

[10] K. A. Funes Mora and J.-M. Odobez. Person independent 3d gaze estimation from remote rgb-d cameras. In *Proc. ICIP*, 2013.

[11] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1362–1376, 2010.

[12] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008.

[13] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.0 beta. `http://cvxr.com/cvx`, sep 2013.

[14] E. D. Guestrin and E. Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Trans. on Biomedical Engineering*, 53(6):1124–1133, 2006.

[15] D. W. Hansen and Q. Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(3):478–500, 2010.

[16] T. Ishikawa, S. Baker, I. Matthews, and T. Kanade. Passive driver gaze tracking with active appearance models. In *Proc. 11th World Congress on Intelligent Transportation Systems*, 2004.

[17] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proc. fourth Eurographics symposium on Geometry processing*, pages 61–70, 2006.

[18] D. Levin. Mesh-independent surface interpolation. In *Geometric Modeling for Scientific Visualization*, Mathematics and Visualization, pages 37–49. Springer-Verlag Limited, 2004.

[19] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. Inferring human gaze from appearance via adaptive linear regression. In *Proc. ICCV*, pages 153–160, 2011.

[20] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. Head pose-free appearance-based gaze sensing via eye image synthesis. In *Proc. ICPR*, pages 1008–1011, 2012.

[21] F. Lu, Y. Sugano, O. Takahiro, and Y. Sato. A head pose-free approach for appearance-based gaze estimation. In *Proc. BMVC*, 2011.

[22] C. D. McMurrough, V. Metsis, D. Kosmopoulos, I. Maglogiannis, and F. Makedon. A dataset for point of gaze detection using head poses and eye images. *Journal on Multimodal User Interfaces*, 2013.

[23] E. Murphy-Chutorian and M. M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(4):607–626, 2009.

[24] A. Nakazawa and C. Nitschke. Point of gaze estimation through corneal surface reflection in an active illumination environment. In *Proc. ECCV*, pages 159–172, 2012.

[25] V. Ponz, A. Villanueva, and R. Cabeza. Dataset for the evaluation of eye detector for gaze estimation. In *Proc. 2nd International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction*, pages 681–684, 2012.

[26] R. Rodrigues, J. Barreto, and U. Nunes. Camera pose estimation using images of planar mirror reflections. In *Proc. ECCV*, pages 382–395, 2010.

[27] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927 – 941, 2008.

[28] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake. Efficient human pose estimation from single depth images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(12):2821 – 2840, 2013.

[29] B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar. Gaze locking: Passive eye contact detection for human-object interaction. In *Proc. 26th Symposium on User Interface Software and Technology*, pages 271–280, 2013.

[30] Y. Sugano, Y. Matsushita, and Y. Sato. Appearance-based gaze estimation using visual saliency. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(2):329–341, 2013.

[31] K.-H. Tan, D. Kriegman, and N. Ahuja. Appearance-based eye gaze estimation. In *Proc. 6th IEEE Workshop on Applications of Computer Vision*, pages 191–195, 2002.

[32] U. Weidenbacher, G. Layher, P. M. Strauss, and H. Neumann. A comprehensive head pose and gaze database. In *Proc. 3rd IET International Conference on Intelligent Environments*, pages 455–458, 2007.

[33] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.

[34] O. Williams, A. Blake, and R. Cipolla. Sparse and semi-supervised visual mapping with the $S^3$GP. In *Proc. CVPR*, volume 1, pages 230–237, 2006.

[35] L.-Q. Xu, D. Machin, and P. Sheppard. A novel approach to real-time non-intrusive gaze finding. In *Proc. BMVC*, 1998.

[36] 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 *Proc. Eye Tracking Research & Applications*, pages 245–250, 2008.