Edge-Guided Near-Eye Image Analysis for Head Mounted Displays

Zhimin Wang[1] * Yuxin Zhao[1] † Yunfei Liu[1] ‡ Feng Lu[1,2, §]

[1] State Key Laboratory of VR Technology and Systems, School of CSE, Beihang University
[2] Peng Cheng Laboratory, Shenzhen, China

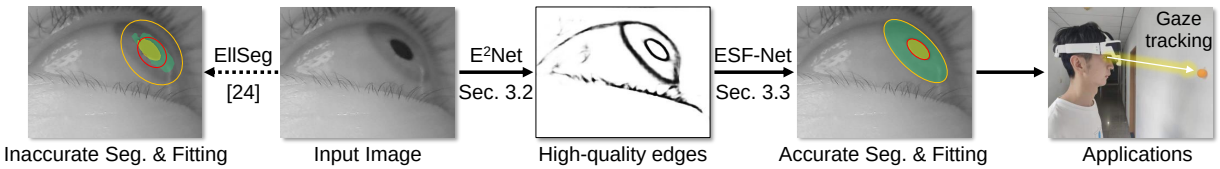

Figure 1: To improve the accuracy of eye tracking in AR, we propose a novel eye segmentation and fitting method that estimates pupil and iris parameters on the guidance of task-related edges. We train networks on synthetic datasets. Then we evaluate them on publicly available real datasets and customized AR Head Mounted Display (HMD) devices. The results show our methods outperform current state-of-the-art methods.

Abstract

Eye tracking provides an effective way for interaction in Augmented Reality (AR) Head Mounted Displays (HMDs). Current eye tracking techniques for AR HMDs require eye segmentation and ellipse fitting under near-infrared illumination. However, due to the low contrast between sclera and iris regions and unpredictable reflections, it is still challenging to accomplish accurate iris/pupil segmentation and the corresponding ellipse fitting tasks. In this paper, inspired by the fact that most essential information is encoded in the edge areas, we propose a novel near-eye image analysis method with edge maps as guidance. Specifically, we first utilize an Edge Extraction Network (E^2-Net) to predict high-quality edge maps, which only contain eyelids and iris/pupil contours without other undesired edges. Then we feed the edge maps into an Edge-Guided Segmentation and Fitting Network (ESF-Net) for accurate segmentation and ellipse fitting. Extensive experimental results demonstrate that our method outperforms current state-of-the-art methods in near-eye image segmentation and ellipse fitting tasks, based on which we present applications of eye tracking with AR HMD.

Index Terms: Augmented Reality—Eye tracking—Near-eye image analysis—Edge Extraction; Human Computer Interaction (HCI)

1 Introduction

Interaction techniques seek to enrich the user experiences, which is important when using Augmented Reality (AR) Head Mounted Display (HMD) devices [30]. Among different interaction techniques, eye gaze tracking requires less physical demand, and provides more natural experience, thus is potentially an effective channel in AR HMDs. More recent efforts have focused on taking the advantage of gaze interaction in AR systems [3, 55, 57]. However, the insufficient eye tracking accuracy always degrades the user experience [30].

Researchers have made efforts to develop robust near-eye tracking techniques. These methods need to compute gaze-relative features, *e.g.*, pupil center, pupil ellipse and iris ellipse, from infrared (IR) eye images, and then use them to build certain models to compute gaze positions in the scene images [25]. A series of methods use the Pupil Center Corneal Reflection (PCCR) to compute the pupil center [19], while recent CNN-based methods directly regress pupil centers from eye images [26, 34, 53].

On the other hand, many works focus on pupil ellipse fitting. Some of them search the pupil ellipse based on the morphological processing [16, 45], while other CNN-based methods such as DeepVoG [62] uses the U-Net [44] to segment out the pupil area and then fit an ellipse on it. To obtain the iris ellipse, the model-based methods are proposed to fit the iris boundary [20] especially when the images are captured by RGB cameras. Kothari *et al.* proposed the EllSeg method that segments out complete pupil and iris structures, and showed its effectiveness when dealing with partially occluded pupil or iris contours [29].

No matter which eye feature is needed, *e.g.* pupil center, pupil ellipse or iris ellipse, the key is to develop an effective tool for near-eye image analysis. To achieve this goal, taking Fig. 1 as an example, we observe that most discriminate information in the eye image is encoded in certain edge areas, including two eyelids, pupil contour and iris contour. Such edges are highly related to the high-level semantic tasks. A similar observation has already been demonstrated and used in common image segmentation tasks [50]. However, directly applying this idea to the near-eye image scenario is difficult since eye image also contains inevitable and undesirable edges, as shown in Fig. 3b. It is therefore important to only extract those four ideal edges in the eye while removing the others, and then use them for eye image analysis.

In this paper, we propose a novel near-eye image analysis method that estimates pupil and iris ellipse parameters on the guidance of ideally produced edge maps, aiming at improving the accuracy of eye region segmentation and fitting for eye tracking in AR. As illustrated in Fig. 1, our method first extracts high-quality edges that contain only the upper and lower eyelids and the visible boundary on the eyeball, *e.g.*, iris and pupil contours. Then we utilize the edge information to guide the segmentation and ellipse fitting. The method can be well trained on a synthetic dataset after our proposed preprocessing, and assessed on four publicly available real datasets and a customized AR device. Overall, the summary of our contributions is as follows.

1. We propose a novel near-eye image analysis method with edge maps as guidance. It facilitates natural eye tracking and interaction in AR.

2. We propose an Edge Extraction Network (E^2-Net) to pro-

*joint first author, e-mail: zm.wang@buaa.edu.cn
†joint first author, e-mail: zyuxin@buaa.edu.cn
‡e-mail: lyunfei@buaa.edu.cn
§corresponding author, e-mail: lufeng@buaa.edu.cn

duce high-quality edge maps, which only contain eyelids and iris/pupil contours without other undesired edges. The network is trained on sufficient realistic training images produced by our Image Intensity Transfer (I^2T) technique from synthetic images.

3. We demonstrate the advantage of using such high-quality edge maps in eye image segmentation and ellipse fitting. An Edge-Guided Segmentation and Fitting Network (ESF-Net) is proposed to accomplish this task.

4. Extensive experimental results are obtained on four publicly available datasets and our captured videos. Our method outperforms current state-of-the-art methods in near-eye image segmentation and ellipse fitting tasks. Applications of eye-tracking in AR HMD are presented.

2 Related Work

In this section, we review gaze interaction in AR and near-infrared eye tracking techniques, and discuss edge detection and semantic segmentation approaches.

2.1 Gaze Interaction in AR

Gaze-based interaction requires less physical demand, and provides more natural experience than hand gesture or speech input, and thus is potentially an effective channel in AR systems [41]. Gaze pointing employs eye tracking techniques to identify where a person is looking. More recent efforts have focused on taking the advantage of gaze interaction in AR systems [3, 55, 57]. For instance, Kytö *et al.* [30] leveraged eye gaze to point the object and used secondary modalities to confirm the selection [30]. However, the insufficient eye tracking accuracy always degrades the user experience [57]. Many researchers make efforts to achieve robust gaze estimation by developing near eye tracking technology.

2.2 Near-Eye Image Analysis

Gaze estimation techniques can be broadly categorized in two types: 1) appearance-based methods, 2) model-based methods [48]. Appearance-based methods, which learn a direct mapping from eye images to gaze directions, were proposed for the low-resolution eye images in far-distance scenarios [2, 63]. Model-based methods use eye features to fit 3D eye model [9], or directly compute gaze direction [34]. For near-eye high-resolution images, model-based methods can achieve higher degree accuracy than appearance-base methods [48]. Therefore, we mainly discuss the model-based methods.

Near-eye tracking techniques compute gaze-relative features *e.g.*, pupil center, pupil/iris ellipse, then transform the center coordinates on the eye images to 2D gaze positions on the scene images [25], or infer 3D eye geometric model to obtain the optical axis for 3D gaze estimation [8, 59]. A few efforts have been made to obtain the pupil centers. Pupil Center Corneal Reflection (PCCR) uses one or more infrared light sources to illuminate the eye, and estimate the centers of pupil and glints in images captured by one or more IR cameras [19]. Recent CNN-based methods directly regress pupil centers from eye images, such DeepEye [53], NVGaze [26]. Many works have focused on the pupil ellipse fitting. The traditional methods search the pupil ellipse based on the morphological processing [16, 45]. DeepVOG uses the U-Net to segment the pupil area and fit ellipse [62]. For iris ellipse, Hansen *et al.* used the model-based methods to fit the iris boundary [20], while it is more suitable for the images captured by RGB camera. The partial pupil or iris occlusion due to eyelashes or half-open eyelids often degrades the accuracy of these methods. Kothari *et al.* proposed the EllSeg that predicts full pupil and iris structures, and showed it effectiveness to occlusions [29]. However, due to the low amount of contrast between sclera and iris regions in the near-infrared illumination, these CNN based methods still suffer from coarse segmentation boundaries and insufficient ellipse fitting accuracy. We observed edges in the images include lots of boundary information. To this end, we propose a novel eye segmentation and ellipse fitting with edge maps as guidance.

2.3 Edge Detection and Semantic Segmentation

Recent works have shown that edge maps contain more detailed structure or geometry information, and could benefit semantic segmentation [50], image inpainting [31], and image super-resolution [38].

Current edge detection methods can be classified into three categories: 1) traditional edge detectors, 2) feature based methods, and 3) recent deep learning methods. The traditional edge detectors spot edges by finding the gradient changes in colors, intensities and textures [6, 54]. Feature based methods design hand engineered features, and train a classifier to determine whether each patch belongs to an edge or not [10, 11]. CNN based methods utilize supervised networks and automatically extract hierarchical features [42, 60]. Due to the progressive growth of GANs, The GAN based methods have also explored edge detection. Yang *et al.* employ an encoder-decoder model to generate edge maps of input images, and a discriminator network to distinguish the predicted edge maps from the ground truth edge maps [61]. We utilize GAN for detecting edge information in the eye images.

Semantic pixel-wise segmentation based on deep learning methods has shown significant advantages over the traditional image segmentation [4]. The common used network architecture is the encoder-decoder structure [32, 44]. The decoder network learns to decode the low resolution encoded feature maps for pixel-wise classification. DeconvNet employs multiple deconvolution layers in the decoder to improve segmentation performance [40]. In SegNet, the decoder upsamples the features from its encoder using the pool indices [4]. Recent works explore multi-modal image segmentation for multi-input tasks, *e.g.*, RGB-Depth segmentation [21], and Magnetic Resonance Imaging (MRI) segmentation [12]. The multi-modal *early fusion* strategy is designed to stack the original inputs directly. This method is appropriate for linear relationships that exist between low-level features in same modalities [47]. For different modalities, Nie *et al*. proposed the *late fusion* strategy, where different modalities are processed separately by the CNNs whose high-level outputs are fused [39]. For the integration of edge maps and eye images, our ESF-Net adopts the *late fusion* strategy.

3 Methodology

3.1 Overview

We propose a novel near-eye images analysis method including eye segmentation as well as the ellipse parameter fitting of pupil and iris. The pipeline of our work is shown in Fig. 2. We first propose an edge extraction network (E^2-Net). The network is optimized with adversarial learning to produce high-quality edge maps. To acquire sufficient realistic training images, we further propose the Image Intensity Transfer (I^2T) approach for generating realistic images from synthetic images. We then propose an edge-guided segmentation and fitting network (ESF-Net). Both eye images and the generated edge maps are fed into the network to perform multi-task learning. The network respectively generates the eye segmentation and regresses the ellipse parameters of pupil and iris. In addition, we notice the regressed parameters usually are inaccurate. We also propose the Adaptive Search Module (ASM) to search the optimal ellipse parameters with the guidance of the segmentation maps.

The rest of this section is organized as follows. we introduce the E^2-Net in the first subsection including the architecture of E^2-Net and the detail of I^2T method. The ESF-Net is introduced in the second subsection. We also describe the ASM in this subsection.

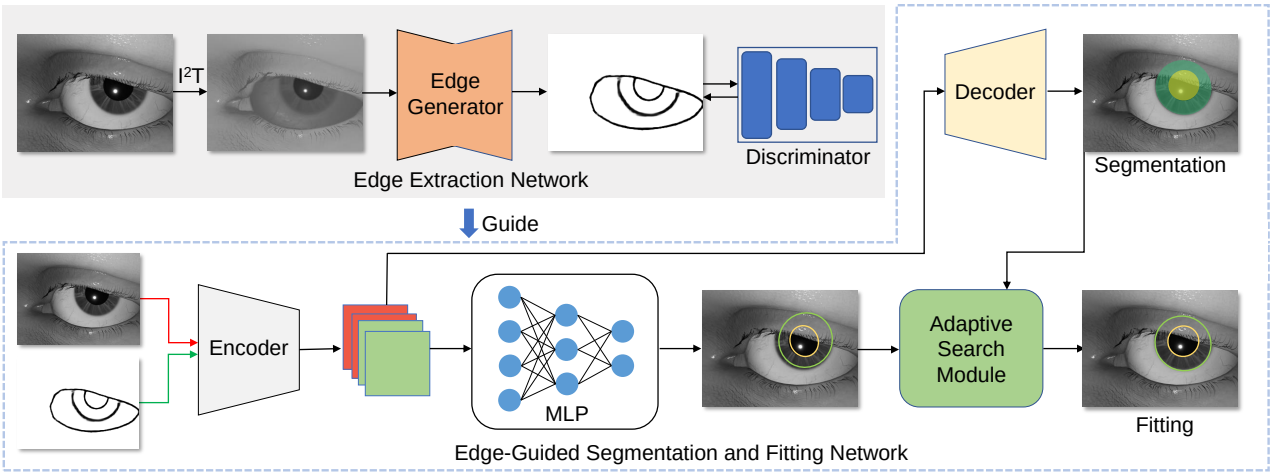

Figure 2: Overview of our method. We propose the ESF-Net that can obtain complete semantic maps of iris and pupil, and accurate ellipse parameters. We also design the E^2-Net that aims to recognize unbroken task-related edges in eye images and overlook the edges of eyelashes, glasses and reflections. ASM fits the optimal ellipse parameters with the guidance of segmentation results and regression results. Note that ASM is actually independent of the segmentation and has no learnable parameters.

Finally, we implement a 2D gaze estimation task in the customized AR HMD device. The detail of the implementation is introduced in the third subsection. We also provide the implementation details of E^2-Net and ESF-Net in the last.

3.2 Edge Detection

To detect edge maps from eye images, we respectively propose the E^2-Net for edge extraction and the I^{2}T for realistic training images generation. We first introduce the E^2-Net.

3.2.1 Edge Extraction Network

Our task is to analyze near-eye images. Therefore, the goal of E^2-Net is to extract the task-related edges, which contain the upper and lower eyelids, and the visible boundary inside the eyeball, *e.g.*, iris and pupil contours. There are two challenges in the task-related edge extraction: 1) To eliminate task-unrelated edges. We show the result of Canny detection [6] in Fig. 3b. Many undesired edges such as eyelashes, glasses and glints are extracted. These edges are mixed with the task-related edges and complicate the task-related edge extraction task. 2) To complete task-related edges. Some environmental factors such as ambient infrared illumination have large impact on the edge extraction. For example, as shown in the bottom of Fig. 3a, the limbus (border of the iris and sclera) is blurry due to the ambient infrared illumination. This causes the task-related edges lost (red box in Fig. 3b) and require methods to complete the task-related edges. To handle the two challenges, we integrate edge generation networks with adversarial learning [18] and propose the E^2-Net. The top of Fig. 2 shows the overall structure of E^2-Net.

Architecture: The E^2-Net contains an edge generator G and a discriminator D. Let $\{(\boldsymbol{I}_i, \boldsymbol{E}_i)\}$ denotes a set of pairs of corresponding images in the training set $\mathbb{T}$, where $\boldsymbol{I}_i$ is an eye image and $\boldsymbol{E}_i$ is the corresponding edge map. The generator G generates edge map $\hat{\boldsymbol{E}}_i$ from images, *i.e.*, $\hat{\boldsymbol{E}}_i := G_{\boldsymbol{\theta}}(\boldsymbol{I}_i)$, where $\boldsymbol{\theta}$ represents function parameters. The discriminator $D_{\boldsymbol{\phi}}$ performs binary classification, where $\boldsymbol{\phi}$ represents the parameters of the discriminator. It is fed with edge maps, and aims to distinguish the ground truth edge map $\boldsymbol{E}_i$ from the predicted one $\hat{\boldsymbol{E}}_i$,

Loss functions: We optimize the generator by minimizing the combination of cross-entropy loss and adversarial loss:

$$\mathcal{L}_G(\boldsymbol{\theta}) = \alpha \sum \ell_{\text{bce}}\left(\hat{\boldsymbol{E}}_i, \boldsymbol{E}_i\right) + \beta \sum \ell_{\text{pure}}\left(\boldsymbol{\theta}; \boldsymbol{I}_i\right), \tag{1}$$

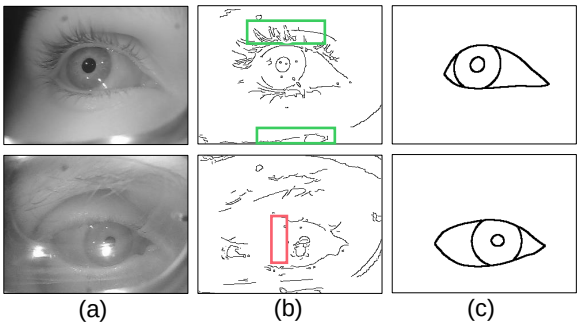

Figure 3: Edge extraction from eye images. (a) Eye images captured in AR HMD device. (b) Results of edge extraction utilizing the Canny detection [6]. (c) Task-related edges. Notice the edges in green boxes represent task-unrelated edges. We seek to employ E^2-Net to spot task-related edges and restore the lost edges in red box.

where α and β are fixed weight parameters. The binary cross-entropy loss, ℓ_{bce} minimizes the difference between the predicted edge map and the ground truth edge map. The formulation of the ℓ_{bce} can refer to equation (10) in [22].

The adversarial loss ℓ_{pure} forces the generator to produce desired edge maps. The regular generative adversarial network (GAN) suffers from vanishing gradients due to the cross entropy loss function [36], which makes it difficult to update the generator. Inspired by the Least Squares GAN (LSGAN) [35], we utilize the least squares loss function to formulate adversarial loss. The formulation is:

$$\ell_{\text{pure}}(\boldsymbol{\theta}; \boldsymbol{I}_i) = (D_{\boldsymbol{\phi}}(R_{\boldsymbol{\theta}}(\boldsymbol{I}_i)) - 1)^2 = (D_{\boldsymbol{\phi}}(\hat{\boldsymbol{E}}_i) - 1)^2. \tag{2}$$

The discriminator network aims to correctly distinguish inputs. It is optimized with:

$$\mathcal{L}_D(\boldsymbol{\phi}) = \sum (D_{\boldsymbol{\phi}}(\boldsymbol{E}_i) - 1)^2 + \sum (D_{\boldsymbol{\phi}}(\hat{\boldsymbol{E}}_i))^2. \tag{3}$$

3.2.2 Image Intensity Transfer

We use sufficient synthetic images for training the E^2-Net. However, there is large difference between the real and synthetic images. The

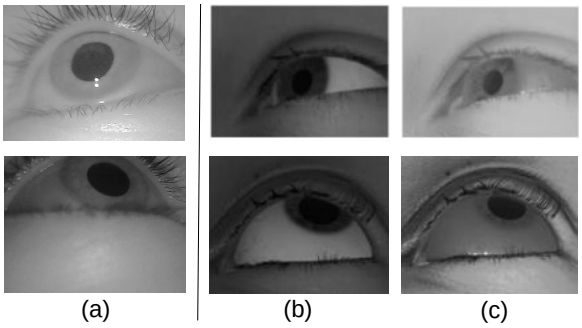

Figure 4: Eye appearance of real images and synthetic images. (a) Real images from different datasets. (b) Synthetic images from RIT-Eyes. (c) The corresponding transferred images employing the Image Intensity Transfer (I^2T) module. Notice the sclera and iris regions in the transferred images are more similar to the real images than the synthetic images.

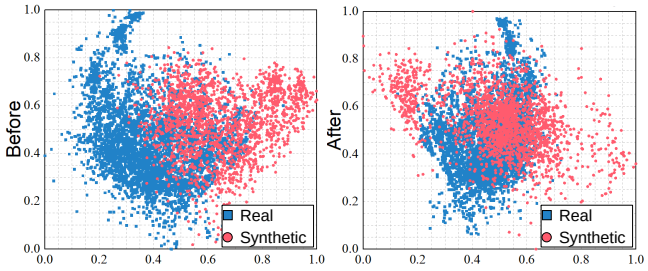

Figure 5: Visualization of real and synthetic images in 2-D space vis t-SNE. Left: The 2-D feature distribution of synthetic and real images before I^2T. Right: The 2-D feature distribution after I^2T. It is obvious that out method shortens the gap between synthetic and real domains.

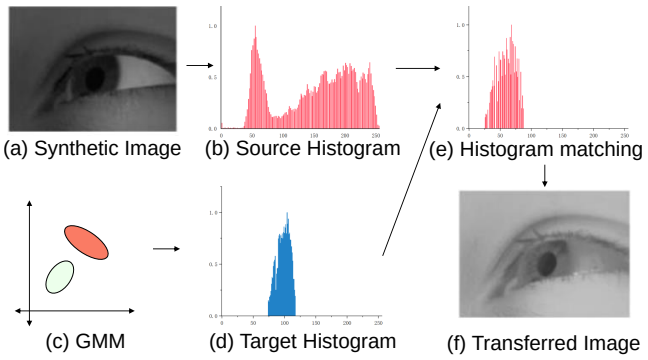

Figure 6: The procedure of aligning image intensity. (a) One synthetic image. The synthetic image is divided into three subregions. (b) The histogram of sclera and iris region. (c) The fitted GMM. (d) The new target histogram samples from the GMM. (e) The histogram matching performs a mapping that transforms the intensities of source to the target. (f) The image is transferred on all three subregions.

large difference usually degrades the model performance and even leads to an incorrect estimation when the model is used in real world. To solve this problem, we propose the I^2T method for producing realistic images from synthetic images.

The I^2T method is designed based on our observation, synthetic images preserve the similar eye architecture while having obvious intensity difference with real images. We also show examples in Fig. 4. The core idea of I^2T method is to align the intensity distribution of synthetic images with real images. We use the dimensionality reduction to visualize the synthetic and real images in 2-D space before and after I^2T, as shown in Fig. 5. The comparison in Fig. 5, also shows that I^2T shortens the gap between synthetic and real images. The dimensionality reduction method we use is t-distributed Stochastic Neighbor Embedding (t-SNE) [52]. The I^2T contains a total of four steps.

1) Sample data: We first sample some real images to model the distribution. To ensure the diversity of real images, we separately sample images from four real near-eye datasets. A total of 10000 near-eye images are sampled as the basic near-eye image library.

2) Compute histogram: Given a real image, we divide it into three subregions which respectively contain iris and sclera, skin, and pupil. In the I^2T, the intensity distribution is separately transferred in each region. We split the sclera and iris into the same subregion for blurring the limbus of synthetic image. We calculate the intensity histogram $\boldsymbol{x^i} \in \mathbb{R}^{256}$ of each region, where $i \in \{1,2,3\}$. We simply refer all the three intensity histograms as $\boldsymbol{x}$ in the rest.

3) Fit mixture gaussian distributions: The basic near-eye image library contains finite eye images. To enlarge the library, we fit three mixture gaussian distributions (GMM) which corresponds to the distributions of three subregion histograms. We can sample from the fitted GMMs for infinite intensity histograms. Besides, the different combinations of three subregion histograms also enrich the targets.

4) Align image intensity: The procedure is shown in Fig. 6. Given a synthetic image, we first compute the intensity histogram of each subregion of synthetic image. On the other hand, we sample the intensity histogram as target from the fitted GMM. Then we employ the histogram matching algorithm [46] and perform a mapping that transforms intensities of the source image towards the target.

3.3 Edge-Guided Segmentation and Fitting Network

The edge extraction network can provide accurate task-related edge maps. Then, We proposed the ESF-Net, which utilizes the edge maps to guide the eye segmentation and the ellipse parameter fitting. The architecture of ESF-Net is shown in Fig. 2.

The input of ESF-Net is near-eye images and the generated edge maps. We feed the two images into an encoder for feature extraction. We concatenate their extracted feature maps in channel dimension to form eye feature. The eye feature contain edge information and semantic information of the near-eye images, and we use the feature to perform two tasks. The feature is first fed into a decoder to perform eye segmentation. The decoder is composed of multi-deconvolution layers. The output of decoder is a three-channel segmentation map, which contains: iris, pupil and background. The iris and pupil centers are derived from segmentation maps using the weighted summation of pixel coordinates [29]. The feature is second fed into a Multilayer Perceptron (MLP) to regress the ellipse axes and orientation. The loss functions of ESF-Net are

$$\mathcal{L} = \mathcal{L}_{seg} + \mathcal{L}_{fit}, \tag{4}$$

where $\mathcal{L}_{seg}$ denotes the loss of segmentation maps, $\mathcal{L}_{fit}$ represents the loss of ellipse parameter fitting. The $\mathcal{L}_{seg}$ we use is proposed in RITnet [7], it constrain these problems such as the blurry boundary of segmentation map and the class imbalance of each area. $\mathcal{L}_{fit}$ is the L1 Loss, which computes the errors of ellipse parameters.

Based on the observation in the final outputs, we found the ellipse of segmentation map is more accurate than the fitting results of regression module. However, the parameters (a,b,θ) are hard to obtain directly from the segmentation map, *e.g.*, occasionally broken segmentation maps (Fig. 7). To this end, we propose an Adaptive Search Module (ASM) to search the optimal ellipse parameters (a,b,θ) on the guidance of segmentation maps, as illustrated in Algorithm 1. The searching goal is to maximize the value of Intersection Over Union (IoU) between the segmentation map and the

Algorithm 1: Adaptive Search Module (ASM)

Data: Regression ellipse parameters $P = [a, b, \beta]$, segmentation map (M), max number of steps (T)
Result: Optimal ellipse parameters $P^* = [a^*, b^*, \beta^*]$

Initialization step size $D = [1, 1, 1]$;
$\Delta B^* \leftarrow -\infty$;
for $t \leftarrow 1$ **to** T **do**
  for *each* P_i *in* P **do**
    $P_i \leftarrow P_i - D_i$;
    if $IoU(P, M) > \Delta B^*$ **then**
      continue;
    end
    $P_i \leftarrow P_i + 2 \times D_i$;
    if $IoU(P, M) > \Delta B^*$ **then**
      continue;
    end
    $P_i \leftarrow P_i - D_i$;
    $D_i \leftarrow D_i \times 0.8$;
  end
  $\Delta B^* \leftarrow$IoU(P, M);
end
$E^* \leftarrow P$;

ellipse formed by parameters. Specifically, we first keep the ellipse center fixed, and set the regressed (a, b, θ) as the initial parameters. Then in each iteration, parameters are changed based on the step denoted as D. The iteration direction depends on the IoU. The D will gradually decay as the searching process. The algorithm will be iterated for T times, and then an optimal ellipse parameter is found.

3.4 Application

We will make use of the iris parameters to reconstruct the 3D eye geometric model and estimate the optical axis in future work. In this paper, we implement 2D gaze estimation based on a customized AR device for demonstrating the effectiveness of our method. The 2D gaze estimation task transforms the pupil centers on the eye image to 2D gaze positions on the scene camera image, using the polynomial mapping function denoted as equation (4) in [5]. Note that the mentioned mapping function only computes the monocular gaze position. We obtain the binocular gaze position by averaging the gaze positions of left eye and right eye.

3.5 Implementation Details

E^2-Net: We implement our network using PyTorch. The backbone generator G could employ any edge detection network, such as the commonly used HED [60], DexiNed [42]. We adopt the Bi-Directional Cascade Network (BDCN) [22] as our edge generator network G. The BDCN learns the multi-scale representations using a shallow network. We set the batch size to 48 for all the experiments. We observe the unbalanced ratio of edge/non-edge pixels of eye image, and set the balance parameter λ as 3. We sum the binary cross-entropy loss ℓ_{bce}, which contains 320×240 (image size) loss items. In order to train the E^2-Net stably, we fix the weights of generator loss to $\alpha = 0.2$ and $\beta = 1000$. We set the initial learning rate of generator G and discriminator D to 1e-6 and 1e-4 separately, which decreases by half after every 30 iterations. For other parameters, we keep the same setting as the original network. We apply a multi-scale discriminator architecture [24, 56] to guide the generator to produce pure task-related edge map. Edge detection experiments are conducted on an NVIDIA GeForce RTX 3090 GPU with 24 GB memory.

ESF-Net: Indeed, the encoder, decoder and regression module can be arbitrary. In this paper, we employ the DenseEINet as the backbone, two-layer MLP for regression module similar to EllSeg [29]. All networks are trained with a constant 5×10^{-4} learning rate and 48 batch size using ADAM optimizer [28]. We empirically set T as 40 in adaptive search module. Segmentation and fitting experiments are conducted on two NVIDIA GeForce RTX 3090 GPUs with 48 GB memory.

Table 1: Summary of train and test datasets. Note that we discard images without valid pupil and iris fits. * indicates we employ annotations presented in TEyeD.

Dataset	Source	Purpose	Image Count	Sample Count
RITEyes-General	Synthetic	Train	45516	45516
NVGaze-AR*	AR HMD	Test	2265127	11051
OpenEDS	VR HMD	Test	11202	11200
LPW*	Head-mounted Eye tracker	Test	130856	10865
Fuhl*	Head-mounted Eye tracker	Test	5665053	11197

4 Experiments

In this section, we compare our method with existing methods through extensive experiments. We use the publicly-available codes with recommended parameter settings. This section is organized as follows. Firstly, we introduce the datasets and data processing methods. Then we make quantitative and qualitative comparisons with the state-of-the-art methods in segmentation and fitting results. Afterwards, we report our edge detection performance on real near-eye datasets. Besides, we also provide ablation studies to validate the effectiveness of different modules. Finally, We demonstrate the robustness and benefit of our method to gaze-tracking tasks in real AR device.

4.1 Datasets

In this work, we employ the synthetic eye dataset as our train dataset, which has accurate and abundant label information. The synthetic datasets can be categorized in two types: one based on natural light [48, 58], and another based on near-infrared light [26, 49]. Due to near-infrared camera used by AR HMD, we use the RITEyes-General [37]. We choose the following real near-eye datasets for our experiments: NVGaze-AR [26], OpenEDS [17], LPW [51], ElSe [16], ExCuSe [14] and PupilNet [15]. We combine ElSe, ExCuSe with PupilNet, and cite them as Fuhl. These datasets other than OpenEDS only annotate pupil center. Recent work TEyeD [13] adopts semi-automated annotation method and provides annotations of pupil, iris and eyelids for NVGaze-AR, LPW, and Fuhl. We employ annotations presented in TEyeD as their ground truth. For OpenEDS, we provide entire semantic masks by applying elliptical fitting algorithm [43]. We utilize the same image preprocessing and data augmentation for ESF-Net as EllSeg. For E^2-Net, we increase the probability (30%) of occurrence, and add the random crop. Because the number of both NVGaze-AR and Fuhl exceeds 2 million, and the similarity between frames is quite high, we reduce the number of test datasets by fixed-interval sampling. We summarize more details about each dataset in Table 1.

4.2 Eye Segmentation and Fitting

We compare our approach with state-of-the-art methods including DeepVOG [62], RITNet [7] and EllSeg [29]. DeepVOG divides images into two classes: pupil and background, *i.e.*, non-iris. RITNet denotes semantic maps as four classes: pupil, iris, sclera, and

Model \ Metric	$IoU_{pupil}\uparrow$	$IoU_{iris}\uparrow$	$PE_{pupil}\downarrow$	$PE_{iris}\downarrow$	$BIoU_{pupil}\uparrow$	$BIoU_{iris}\uparrow$	$IoU_{pupil}\uparrow$	$IoU_{iris}\uparrow$	$PE_{pupil}\downarrow$	$PE_{iris}\downarrow$	$BIoU_{pupil}\uparrow$	$BIoU_{iris}\uparrow$
Benchmark	**LPW**						**NVGaze**					
DeepVOG	0.833	-	4.66	-	-	-	0.867	-	1.23	-	-	-
RITNet	0.822	0.509	7.47	12.28	-	-	0.881	0.773	1.85	3.66	-	-
EllSeg	0.876	0.527	5.17	12.29	0.679	0.622	0.878	0.758	1.25	3.21	0.758	0.718
Ours (E)	0.885	**0.689**	4.41	11.84	**0.780**	0.700	**0.890**	**0.814**	1.24	**2.90**	**0.765**	**0.747**
Ours (E+I)	**0.896**	0.688	**3.68**	**10.27**	0.762	**0.745**	0.884	0.812	**1.20**	3.24	0.763	0.745
Benchmark	**OpenEDS**						**Fuhl**					
DeepVOG	0.890	-	1.41	-	-	-	0.856	-	4.38	-	-	-
RITNet	0.889	0.611	2.41	6.24	-	-	0.862	0.718	5.09	7.91	-	-
EllSeg	0.921	0.740	1.39	**5.32**	0.799	0.727	0.893	0.737	3.70	8.02	0.778	0.739
Ours (E)	0.915	**0.850**	1.44	6.06	0.810	0.778	0.893	0.795	2.84	7.63	0.790	0.750
Ours (E+I)	**0.925**	0.821	**1.27**	5.40	**0.813**	**0.780**	**0.904**	**0.813**	**2.80**	**7.25**	**0.800**	**0.780**

Table 2: Quantitative comparison between DeepVOG, RITNet, EllSeg and our methods (along rows) in LPW, NVGaze, OpenEDS and Fuhl dataset. ↑ & ↓ represent larger and smaller is better, respectively. Bold values emphasize the best performance within each dataset. Because DeepVOG and RITNet was not trained to fit the pupil and iris ellipse, we are unable to provide BIoU scores. Ours (E) means the result of our model that only inputs edge maps, Ours (E+I) means the result of our model that combines images with edge maps. "PE" represents the Euclidean distance between centers, denoted as Pixel Error. "BIoU" denotes the Bounding box overlap IoU metric.

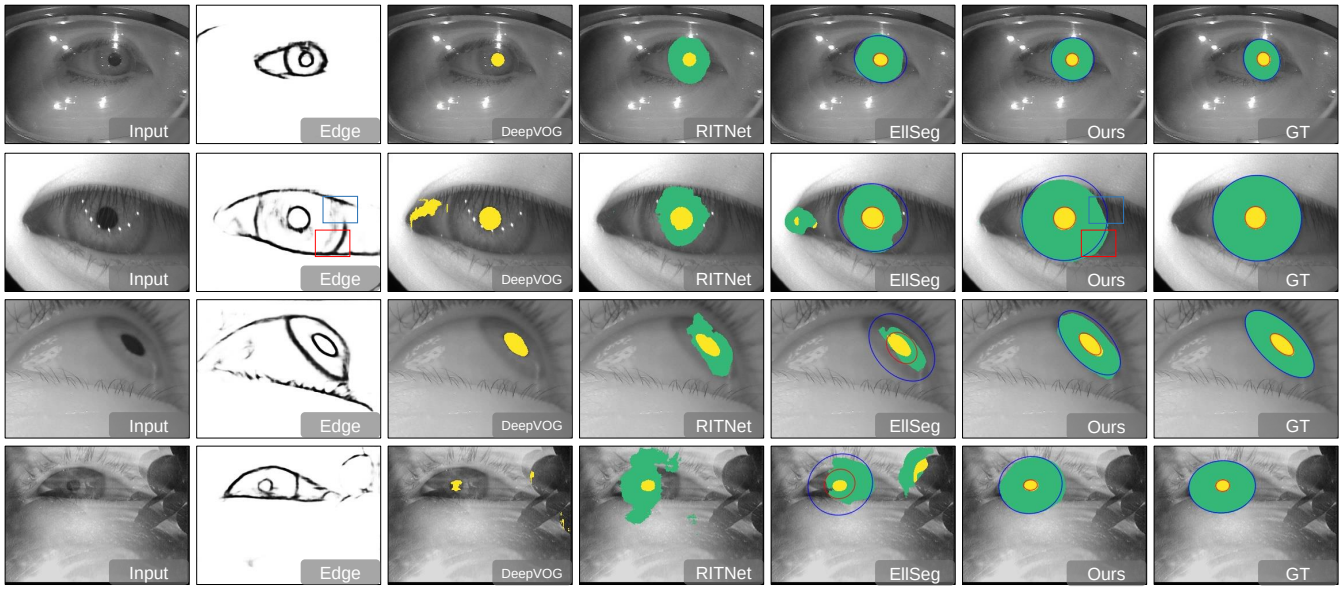

Figure 7: Visual comparisons of segmentation and fitting on NVGaze-AR, OpenEDS, LPW and Fuhl. Obviously, compared to other methods, our approach smooths the contours of segmentation map and produce more accurate ellipse shape. Besides, our method is rarely affected by disturbances, such as the reflections on the glasses or eye corner. The red and blue boxes indicate the good and absent edges, respectively.

background, but no fitting parameters. EllSeg defines three classes: pupil, iris and background, and also output the fitting parameters. The results of our ESF-Net are consistent with EllSeg.

Evaluation Metrics: We reports our segmentation and fitting results with commonly used evaluation metrics: 1) IoU: All segmentation performance is assessed by the mean Intersection Over Union (IoU) scores . 2) PE: The accuracy of pupil and iris centers is evaluated as the Euclidean distance between the predicted value and their corresponding annotations, denoted as Pixel Error (PE). 3) BIoU: Ellipse parameters accuracy is measured by a Bounding box overlap IoU metric (BIoU) proposed by EllSeg [29]. The bounding box uses a minimal rectangle to enclose elliptical structure, and therefore, it can evaluate ellipse parameters (a, b, θ).

Comparison with state-of-the-art: Quantitative comparison results are shown in Table 2, from which we make the following observations. 1) Among all the 24 metrics from four datasets, our methods outperform other methods in 23 metrics, meaning that our methods in general improve the segmentation and ellipse fitting accuracy significantly. 2) The only metric on which our methods fail to be the best is PE_{iris}. Our (E+I) achieves 5.4 pixel error, higher than EllSeg by only 0.08 pixel. Such a difference is not obvious. 3) In average, our methods outperform the second best method by 1% and 15% in pupil and iris segmentation accuracy, respectively. 4) In average, our methods surpass the second best method by 4% and 9% in pupil and iris ellipse fitting accuracy, respectively. 5) Our (E) and Our (E+I) achieve generally similar average results. This demonstrates the advantage of using our extracted high-quality edges in segmentation and ellipse fitting, even without using the original image as input.

Visual examples are shown in Fig. 7. As shown, ESF-Net preserves the structural integrity of segmentation map, and improves the accuracy of ellipse fitting. Specifically, our method surpasses the other methods in two key aspects. On the one hand, our method is more interested in boundary region, thus smooths the contours of

Table 3: Quantitative comparison between Canny, Sobel, BDCN and E^2-Net (Our) in NVGaze-AR, OpenEDS, LPW and Fuhl datasets. Bold values indicate the best performance within each dataset. The higher the better for all metrics.

Dataset	Method	ODS ↑	OIS ↑	AP ↑
NVGaze-AR	Canny	0.182	0.182	0.071
	Sobel	0.230	0.239	0.166
	BDCN	0.512	0.522	0.487
	Ours	**0.550**	**0.559**	**0.532**
OpenEDS	Canny	0.173	0.173	0.075
	Sobel	0.187	0.200	0.113
	BDCN	0.496	0.504	0.487
	Ours	**0.541**	**0.547**	**0.541**
LPW	Canny	0.122	0.122	0.063
	Sobel	0.180	0.190	0.120
	BDCN	0.396	0.411	0.380
	Ours	**0.417**	**0.426**	**0.402**
Fuhl	Canny	0.154	0.154	0.060
	Sobel	0.177	0.185	0.096
	BDCN	0.460	0.470	0.429
	Ours	**0.475**	**0.484**	**0.444**

segmentation map (the 1st and 2nd rows in Fig. 7) and segments more accurate ellipse shape (the 3rd row in Fig. 7). The 2nd row in Fig. 7 distinctly illustrates the importance of task-related edges. The red box indicates that our ESF-Net correctly segments the map in this subregion due to the good edge, while the blue box reveals that the segmentation map is deficient due to the absent edge. On the other hand, the ESF-Net learns the appearance restrain between ellipses and eyelids from high-quality edge maps, and is rarely affected by disturbances, *e.g.*, the reflections on the glasses (the 4th row in Fig. 7) and the eye corner (the 2nd row in Fig. 7).

4.3 Edge Detection

We compare our approach with traditional edge detection methods including Canny detection [6], Sobel detection [54] and deep learning based BDCN [22] on all real near-eye datasets.

Evaluation Metrics: The predicted edge map is an edge probability map (EPM), but not a binary edge map (BEM). We need to apply a threshold η on the EPM to obtain a BEM. There are two options to set η: the first one is called optimal dataset scale (ODS) which utilizes a fixed η for all edge maps from the same dataset. The second one is known as optimal image scale (OIS), which selects the best η for each image [33]. We employ F-measure ($\frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$) of both ODS and OIS in our experiments. Besides, we also calculate the average precision (AP), which corresponds to the area under the precision-recall curve.

Comparison with state-of-the-art: Quantitative comparison results are shown in Table 3. As shown, our method significantly outperforms other methods in all quality metrics. Specifically, in terms of ODS, E2-Net outperforms the BDCN by 7%, 9%, 5% and 4% on four datasets. The OIS and AP of E^2-Net have also the better results than other methods. The results illustrate that E^2-Net can effectively detect task-related edges. Fig. 8 presents visual comparison. Our method solves the two challenges mentioned in Section 3.2. Firstly, E^2-Net ignores the edges of eyelashes, glasses, and reflections, as illustrated in the green boxes of Fig. 8. Secondly, E^2-Net can accurately extract blurry task-related edges in eye images, as indicated in the red boxes of Fig. 8.

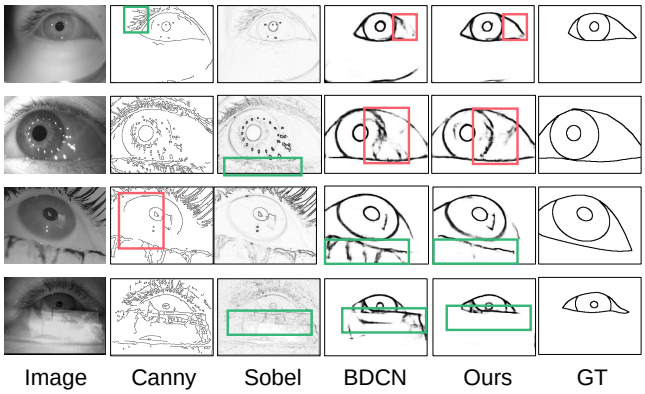

Figure 8: Visual comparison of edge detection on NVGaze-AR, LPW, OpenEDS and Fuhl. Notice the edges in green boxes represent the comparison of task-unrelated edges. The red boxes illustrate the comparison of task-related edges. E^2-Net can accurately spot unbroken task-related edges in eye images and overlook the edges of eyelashes, glasses and reflections.

Table 4: Comparison of using different modules in eye edge detection network.

G	I^{2}T	D	ODS ↑	OIS ↑	AP ↑
✓			0.396	0.411	0.380
✓	✓		0.406	0.416	0.395
✓	✓	✓	**0.417**	**0.426**	**0.402**

Table 5: Comparison of using different inputs in ESF-Net. "I" means the images, "E" means edge maps.

I	E	IoU_{pupil}	IoU_{iris}	PE_{pupil}	PE_{iris}	$BIoU_{pupil}$	$BIoU_{iris}$
✓		0.876	0.527	5.17	12.29	**0.679**	0.622
	✓	0.885	**0.689**	4.41	11.84	0.668	0.670
✓	✓	**0.896**	0.688	**3.68**	**10.27**	0.665	**0.674**

Table 6: Ablation study for Adaptive Search Module (ASM). The ASM searches the optimal ellipse parameters, thus can improve the BIoU.

I	E	ASM	$BIoU_{pupil}$↑	$BIoU_{iris}$↑
	✓		0.668	0.670
	✓	✓	**0.780**	**0.700**
✓	✓		0.665	0.674
✓	✓	✓	**0.762**	**0.745**

4.4 Ablation study

We design ablation studies on LPW dataset to validate the effectiveness of Image Intensity Transfer (I^{2}T) and GAN for edge detection, and edge maps for segmentation and fitting. The BDCN [22] is used as the baseline of edge detection network. The EllSeg [29] is used as the baseline of segmentation and fitting network.

Effectiveness of I^{2}T and GAN for edge detection: The I^{2}T module is only utilized in edge detection task for extracting high-quality edge maps in real images. As shown in Table 4, our baseline only includes a generator G, which yield the result of ODS = 0.396. By adding the I^{2}T module, our method achieves a 2% improvement. By introducing the discriminator D, our system achieves the best results with 5% improvement than baseline. This hence verifies the importance of I^{2}T and GAN in edge detection.

Effectiveness of Edge maps: To validate the contribution of task-related edges, we separately feed 1) the eye image, 2) the edge map, and 3) the eye image and edge map into the network. As

shown in Table 5, the segmentation metrics (IoU_{pupil} and IoU_{iris}) are significantly improved by introducing the edge maps. After fusing the features of eye images and edge maps, the accuracy of pupil and iris center (PE_{pupil} and PE_{iris}) is further improved. Note that although the $BIoU_{pupil}$ of baseline is higher than ours with 0.01, we observed that the regressed parameter θ is usually inaccurate. Therefore, we propose the ASM to solve this problem. The above experiments show that utilizing high-quality edge maps can greatly improve the results of eye segmentation and fitting.

Effectiveness of ASM: We evaluate the effectiveness of ASM for ellipse fitting. According to Table 6 (the first two rows), we can see that after using the ASM, the $BIoU_{pupil}$ is increased from 0.668 to 0.780 with 17% improvement. The $BIoU_{iris}$ is also improved with 4% improvement. The comparison between the last two rows shows that $BIoU_{pupil}$ and $BIoU_{iris}$ are improved with 14% and 11% respectively. The experimental results prove that the ASM can search the optimal ellipse parameters (a, b, θ) on the guidance of segmentation map and the regressed results.

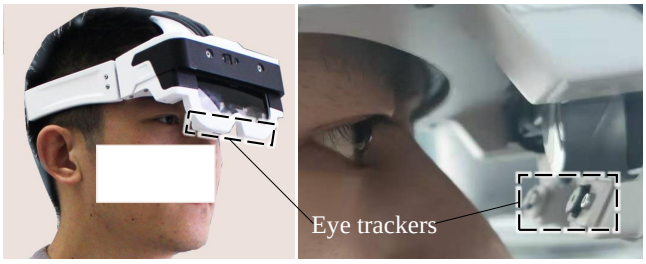

Figure 9: We validate the effectiveness of our methods on the customized AR device. Eye trackers are mounted below the user's eyes.

4.5 Application

To demonstrate the usability of our method, we assessed EllSeg and ESF-Net in the customized AR device, as shown in Fig 9. The diagonal line of AR display images is 80 inches, and the images are shown at a depth of 3m with 1024×768 resolution. We collected user videos under indoor and outdoor illumination environments. The participants can wear glasses or make-up. We visually compare segmentation and ellipse fitting results, as illustrated in Fig 10. As shown, our method obtains accurate segmentation maps and ellipse parameters. The videos of specific comparison can be found in supplemental videos.

We also implemented the 2D gaze estimation task in the AR device. A user was select to complete this test. We used 9 calibration points to calibrate eye gaze, and set these points as references. We visualize the eye gaze, as shown in Fig 11. Our system estimates gaze with an error of only about 0.38° on average.

5 Conclusion

We presented a novel near-eye image analysis method including eye segmentation and the ellipse fitting of pupil and iris with edge maps as guidance. To this end, We first showed the E^2-Net that is optimized with adversarial learning to produce task-related edge maps. The network is trained on sufficient realistic training images produced by the I^2T approach from synthetic images. We then introduced the ESF-Net. Both eye images and the predicted edge maps are fed into the network to conduct multi-task learning. The ESF-Net generates the segmentation maps and regress the ellipse parameters. However, we notice the regresses parameters are usually inaccurate. We also proposed the ASM to search the optimal ellipse parameters with the guidance of segmentation maps. Extensive experimental results show that our method outperforms current state-of-the-art methods in near-eye image segmentation and ellipse fitting tasks. We also provide the applications of eye-tracking in AR HMD.

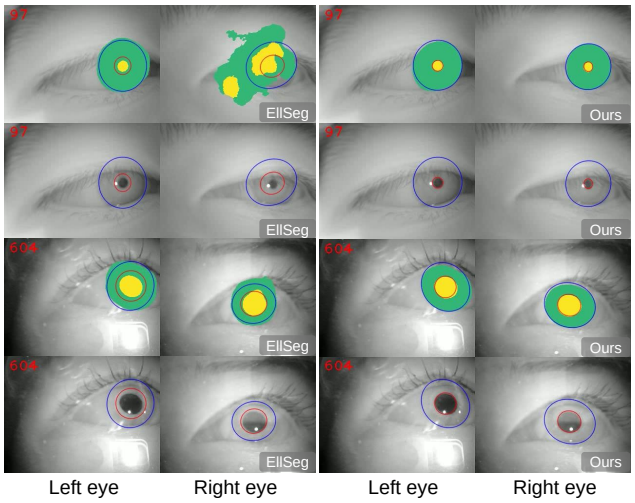

Figure 10: Left: The result of baseline method. Right: The result of our method. The visual comparison demonstrates the advantage of using such high-quality edge maps in eye image segmentation and ellipse fitting.

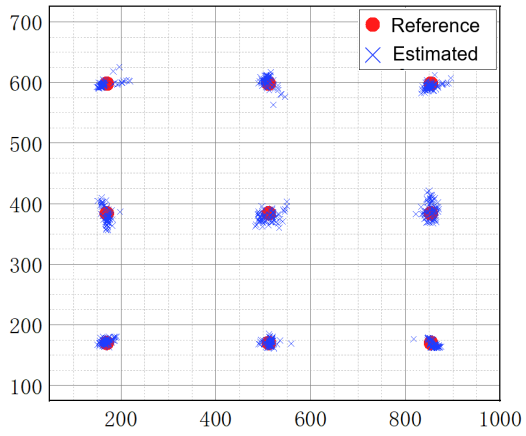

Figure 11: Gaze Error visualization in the real AR devices. Reference and estimated points are marked with red circle and blue diagonal cross, separately.

Future work: 1) Currently, we collected data by using our device and assessed the networks offline. In the future, we will conduct more online tests with the AR device in real time. Besides, we can use knowledge distillation to optimize our network for energy-efficient application [23]. 2) Our method provides different features, *e.g.*, pupil center, pupil ellipse, and iris ellipse. These features can support many applications [1, 27]. We will fit 3D eye model based on a set of eye features, which can mitigate the effect of device slippage and improve the stability of gaze estimation [9].

References

[1] Microsoft HoloLens 2, (2021).

[2] A. A. Akinyelu and P. J. Blignaut. Convolutional neural network-based methods for eye gaze estimation: A survey. *IEEE Access*, 8:142581–142605, 2020.

[3] M. Bâce, T. Leppänen, D. G. De Gomez, and A. R. Gomez. ubigaze: ubiquitous augmented reality messaging using gaze gestures. In *SIG-*

GRAPH ASIA 2016 Mobile Graphics and Interactive Applications, pp. 1–5. 2016.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[5] P. Blignaut. A new mapping function to improve the accuracy of a video-based eye tracker. In *Proceedings of the south african institute for computer scientists and information technologists conference*, pp. 56–59, 2013.

[6] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[7] A. K. Chaudhary, R. Kothari, M. Acharya, S. Dangi, N. Nair, R. Bailey, C. Kanan, G. Diaz, and J. B. Pelz. Ritnet: real-time semantic segmentation of the eye for gaze tracking. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 3698–3702. IEEE, 2019.

[8] K. Dierkes, M. Kassner, and A. Bulling. A novel approach to single camera, glint-free 3d eye model fitting including corneal refraction. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, pp. 1–9, 2018.

[9] K. Dierkes, M. Kassner, and A. Bulling. A fast approach to refraction-aware eye-model fitting and gaze prediction. In K. Krejtz and B. Sharif, eds., *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, pp. 23:1–23:9. ACM, 2019.

[10] P. Dollar, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 1964–1971. IEEE, 2006.

[11] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1558–1570, 2014.

[12] J. Dolz, I. B. Ayed, and C. Desrosiers. Dense multi-path u-net for ischemic stroke lesion segmentation in multiple image modalities. In *International MICCAI Brainlesion Workshop*, pp. 271–282. Springer, 2018.

[13] W. Fuhl, G. Kasneci, and E. Kasneci. Teyed: Over 20 million real-world eye images with pupil, eyelid, and iris 2d and 3d segmentations, 2d and 3d landmarks, 3d eyeball, gaze vector, and eye movement types. *arXiv preprint arXiv:2102.02115*, 2021.

[14] W. Fuhl, T. Kübler, K. Sippel, W. Rosenstiel, and E. Kasneci. Excuse: Robust pupil detection in real-world scenarios. In *International conference on computer analysis of images and patterns*, pp. 39–51. Springer, 2015.

[15] W. Fuhl, T. Santini, G. Kasneci, W. Rosenstiel, and E. Kasneci. Pupilnet v2. 0: Convolutional neural networks for cpu based real time robust pupil detection. *arXiv preprint arXiv:1711.00112*, 2017.

[16] W. Fuhl, T. C. Santini, T. Kübler, and E. Kasneci. Else: Ellipse selection for robust pupil detection in real-world environments. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pp. 123–130, 2016.

[17] S. J. Garbin, O. Komogortsev, R. Cavin, G. Hughes, Y. Shen, I. Schuetz, and S. S. Talathi. Dataset for eye tracking on a virtual reality platform. In *ACM Symposium on Eye Tracking Research and Applications*, pp. 1–10, 2020.

[18] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[19] E. D. Guestrin and M. Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on biomedical engineering*, 53(6):1124–1133, 2006.

[20] D. W. Hansen and A. E. Pece. Eye tracking in the wild. *Computer Vision and Image Understanding*, 98(1):155–181, 2005.

[21] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian conference on computer vision*, pp. 213–228. Springer, 2016.

[22] J. He, S. Zhang, M. Yang, Y. Shan, and T. Huang. Bi-directional cascade network for perceptual edge detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3828–3837, 2019.

[23] A. Holliday, M. Barekatain, J. Laurmaa, C. Kandaswamy, and H. Prendinger. Speedup of deep learning ensembles for semantic segmentation using a model compression technique. *Computer Vision and Image Understanding*, 164:16–26, 2017.

[24] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 172–189, 2018.

[25] M. Kassner, W. Patera, and A. Bulling. Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication*, pp. 1151–1160, 2014.

[26] J. Kim, M. Stengel, A. Majercik, S. De Mello, D. Dunn, S. Laine, M. McGuire, and D. Luebke. Nvgaze: An anatomically-informed dataset for low-latency, near-eye gaze estimation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2019.

[27] T. Kim and E. Lee. Experimental verification of objective visual fatigue measurement based on accurate pupil detection of infrared eye image and multi-feature analysis. *Sensors*, 20:4814, 08 2020.

[28] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, iclr2015. *arXiv preprint arXiv:1412.6980*, 9, 2015.

[29] R. S. Kothari, A. K. Chaudhary, R. J. Bailey, J. B. Pelz, and G. J. Diaz. Ellseg: An ellipse segmentation framework for robust gaze tracking. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2757–2767, 2021.

[30] M. Kytö, B. Ens, T. Piumsomboon, G. A. Lee, and M. Billinghurst. Pinpointing: Precise head-and eye-based target selection for augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2018.

[31] J. Li, F. He, L. Zhang, B. Du, and D. Tao. Progressive reconstruction of visual structure for image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5962–5971, 2019.

[32] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5168–5177, 2017.

[33] Y. Liu, M. Cheng, X. Hu, J. Bian, L. Zhang, X. Bai, and J. Tang. Richer convolutional features for edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1939–1946, 2019.

[34] C. Lu, P. Chakravarthula, Y. Tao, S. Chen, and H. Fuchs. Improved vergence and accommodation via purkinje image tracking with multiple cameras for ar glasses. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 320–331. IEEE, 2020.

[35] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2794–2802, 2017.

[36] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

[37] N. Nair, A. K. Chaudhary, R. S. Kothari, G. J. Diaz, J. B. Pelz, and R. Bailey. Rit-eyes: realistically rendered eye images for eye-tracking applications. In *ACM Symposium on Eye Tracking Research and Applications*, pp. 1–3, 2020.

[38] K. Nazeri, H. Thasarathan, and M. Ebrahimi. Edge-informed single image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 3275–3284, 2019.

[39] D. Nie, L. Wang, Y. Gao, and D. Shen. Fully convolutional networks for multi-modality isointense infant brain image segmentation. In *2016 IEEE 13Th international symposium on biomedical imaging (ISBI)*, pp. 1342–1345. IEEE, 2016.

[40] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pp. 1520–1528, 2015.

[41] H. M. Park, S. H. Lee, and J. S. Choi. Wearable augmented reality system using gaze interaction. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 175–176. IEEE, 2008.

[42] X. S. Poma, E. Riba, and A. Sappa. Dense extreme inception network: Towards a robust cnn model for edge detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp.

1923–1932, 2020.

[43] D. K. Prasad, M. K. Leung, and C. Quek. Ellifit: An unconstrained, non-iterative, least squares based geometric ellipse fitting method. *Pattern Recognition*, 46(5):1449–1465, 2013.

[44] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

[45] T. Santini, W. Fuhl, and E. Kasneci. Pure: Robust pupil detection for real-time pervasive eye tracking. *Computer Vision and Image Understanding*, 170:40–50, 2018.

[46] D. Shapira, S. Avidan, and Y. Hel-Or. Multiple histogram matching. In *2013 IEEE International Conference on Image Processing*, pp. 2269–2273. IEEE, 2013.

[47] N. Srivastava, R. Salakhutdinov, et al. Multimodal learning with deep boltzmann machines. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, vol. 1, pp. 2231–2239. Citeseer, 2012.

[48] Y. Sugano, Y. Matsushita, and Y. Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1821–1828. IEEE Computer Society, 2014.

[49] L. Swirski and N. A. Dodgson. Rendering synthetic ground truth images for eye tracker evaluation. In P. Qvarfordt and D. W. Hansen, eds., *Proceedings of the 7th ACM Symposium on Eye Tracking Research & Applications*, pp. 219–222. ACM, 2014.

[50] H. Tang, X. Qi, D. Xu, P. H. Torr, and N. Sebe. Edge guided gans with semantic preserving for semantic image synthesis. *arXiv preprint arXiv:2003.13898*, 2020.

[51] M. Tonsen, X. Zhang, Y. Sugano, and A. Bulling. Labelled pupils in the wild: a dataset for studying pupil detection in unconstrained environments. In *Proceedings of the ninth biennial ACM symposium on eye tracking research & applications*, pp. 139–142, 2016.

[52] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(86), 2008.

[53] F. J. Vera-Olmos, E. Pardo, H. Melero, and N. Malpica. Deepeye: Deep convolutional network for pupil detection in real environments. *Integrated Computer-Aided Engineering*, 26(1):85–95, 2019.

[54] O. R. Vincent, O. Folorunso, et al. A descriptive algorithm for sobel image edge detection. In *Proceedings of informing science & IT education conference (InSITE)*, vol. 40, pp. 97–107. Informing Science Institute California, 2009.

[55] P. Wang, X. Bai, M. Billinghurst, S. Zhang, W. He, D. Han, Y. Wang, H. Min, W. Lan, and S. Han. Using a head pointer or eye gaze: The effect of gaze on spatial ar remote collaboration for physical tasks. *Interacting with Computers*, 32(2):153–169, 2020.

[56] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8798–8807, 2018.

[57] Z. Wang, H. Yu, H. Wang, Z. Wang, and F. Lu. Comparing single-modal and multimodal interaction in an augmented reality system. In *2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 165–166. IEEE, 2020.

[58] E. Wood, T. Baltrusaitis, X. Zhang, Y. Sugano, P. Robinson, and A. Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *2015 IEEE International Conference on Computer Vision*, pp. 3756–3764. IEEE Computer Society, 2015.

[59] Z. Wu, S. Rajendran, T. Van As, V. Badrinarayanan, and A. Rabinovich. Eyenet: A multi-task deep network for off-axis eye gaze estimation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 3683–3687. IEEE, 2019.

[60] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 1395–1403, 2015.

[61] H. Yang, Y. Li, X. Yan, and F. Cao. Contourgan: Image contour detection with generative adversarial network. *Knowledge-Based Systems*, 164:21–28, 2019.

[62] Y.-H. Yiu, M. Aboulatta, T. Raiser, L. Ophey, V. L. Flanagin, P. Zu Eulenburg, and S.-A. Ahmadi. Deepvog: Open-source pupil segmentation and gaze estimation in neuroscience using deep learning. *Journal of neuroscience methods*, 324:108307, 2019.

[63] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Appearance-based gaze estimation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4511–4520. IEEE Computer Society, 2015.