2024 18th International Conference on Automatic Face and Gesture Recognition (FG)

An Active-gaze Morphable Model for 3D Gaze Estimation

Hao Sun, Nick Pears and Will Smith
Department of Computer Science, University of York, UK
hs1145@york.ac.uk, nick.pears@york.ac.uk, william.smith@york.ac.uk

***Abstract*— Gaze estimation methods typically regress gaze directions directly from images using a deep network. We show that equipping a deep network with an explicit 3D shape model can: i) improve gaze estimation accuracy, ii) perform well with lower resolution inputs at high frame rates and, importantly, iii) provide a much richer understanding of the eye-region and its constituent gaze system, thus lending itself to a wider range of applications. We use an 'eyes and nose' 3D Morphable Model (3DMM) to capture relevant local 3D facial geometry and appearance, and we equip this with a geometric vergence model of gaze to give an 'active-gaze 3DMM'. Latent codes are used to express eye-region shape, appearance, pose, scale and gaze directions, with these being regressed using a tiny Swin transformer. We achieve fast real time at 89 fps without fitted model rendering and 34 fps with rendering. Our system shows state-of-the-art results on the Eyediap dataset, which provides 3D training supervision and highly competitive results on ETH-XGaze, despite a lack of 3D supervision and without modelling the kappa angle. Indeed, our method can learn with only the ground truth gaze target point and the camera parameters, without access to the ground truth gaze origin points, thus significantly widening applicability.**

I. Introduction

The estimation of gaze direction enables the visual understanding of human intention, with high utility in human-computer interaction and XR. Active systems that project light onto the face/eye region either simplify image processing [1] or provide 3D information directly [2]. Previous passive systems have built an eye model [3], eye-region model [4], or a full head model [5], that can be fitted to given images, which thereby provides a gaze direction estimation. Many systems employ lightweight modelling in the sense that they use landmark extraction for the face, eyelids, iris contour and pupil contour [6], [7], [8].

Also, appearance-based methods that regress gaze directions directly from RGB input images using deep networks, but without the use of 3D shape models, have been popular [9]. Compared to these appearance-based methods, model-based methods are less competitive in regard to gaze estimation accuracy, due to a deep neural network's feature extraction and nonlinear fitting ability. However, most appearance-based gaze estimation methods predict only a gaze direction (azimuth-elevation orientation), but no other information about the 3D geometry of the gaze or the eye-region, which often has high utility, such as design of XR eyewear. Current literature has different gaze origin representations (*e.g.* eyeball centres or a point on the face), which requires additional effort to make performance comparisons [10].

We propose an end-to-end method, combining both appearance-based and model-based elements. Our method reconstructs the 3D eye-nose region, avoiding the highly-variable mouth-jaw area that deforms over expressions, so that it can more accurately predict gaze direction over a wide range of facial shapes and head poses. We employ an *eyes-and-nose* 3D Morphable Model (3DMM) and, crucially, we equip this with a geometric vergence model of gaze. We call this an *active-gaze 3DMM*. This enables the combined rotation of the eyeballs to define the gaze under certain geometric constraints, such as coplanarity of the gaze vectors. As a result, we can model the correlations between the face and the left and right eyeballs. This ensures both accurate gaze estimation and that the eyeball positions are consistent with both the most rigid part of the face geometry and the head pose, see Fig. 1.

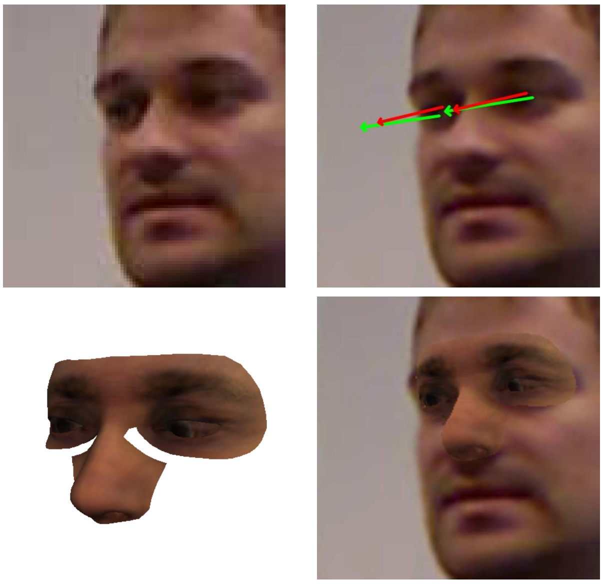

Fig. 1. Active-gaze 3DMM fitting: top left: input image; top-right: predicted gaze directions (red) and ground truth (green); bottom-left: predicted eye region model; bottom-right: eye-region model overlaid on input.

Adding a model to an appearance-based method provides richer information *e.g.* we can trivially estimate the subject's inter-ocular distance, leveraging the fitted model's correlation between shape and size. This is applicable even though one of our training datasets (ETH-XGaze) does not provide the required eyeball centre information. Also, when designing wearable devices such as smart glasses, both the eye-region geometry information and the gaze information are important. Under such circumstances, our approach provides both the eye-region geometry and a much more accurate gaze

979-8-3503-9494-8/24/$31.00 ©2024 IEEE

estimation when compared to purely model-based methods.

Most image autoencoder 3D reconstruction methods from monocular RGB images focus on faces [11], [12]. Typically, their 3D face models only model the eyeball surface area as part of the face, and the gaze directions are not explicitly modelled. Our method both takes advantage of the image autoencoder architecture *and* models the specific eye-region area, designing gaze information into the model.

Note that our system is more than just a gaze estimation system as it generates a 3D fitted model of the whole eye region and therefore has wider utility than a gaze-only estimation system. In a significant sense, therefore, it is not directly comparable to *gaze-only* systems, as the information content of our outputs are much richer. Our system is fast enough to keep pace with high-speed cameras of up to 89fps, which may better temporally localise eye saccades.

Our aim is to investigate what can be achieved when combining appearance-based with model-based system components. The nature of our system is that it explicitly generates a 3D geometric model and therefore we find that it works best when there is good 3D supervision built into the 3D training dataset. For example, we are able to demonstrate state-of-the-art accuracy on Eyediap [13], where 3D supervision is provided. Indeed, such information is readily available from any modern 3D capture system, whether it be RGB-D or multi-view for example. However, explicit 3D supervision is not strictly necessary, as we demonstrate on the ETH-XGaze dataset [14]. Here we demonstrate competitive performance without 3D supervision, although we cannot improve on the current *gaze-only* state-of-the-art system.

In summary, our main contributions are: i) An *active-gaze 3DMM* that focuses on the more rigid 'eyes and nose' region and that is equipped with a geometric eye vergence model for regularisation. ii) Demonstration that the active-gaze 3DMM increases gaze estimation accuracy and versatility. iii) Demonstration of the method's fast inference time for real-time performance and adaptability, when only ground truth 3D gaze targets are available, with no access to gaze origin information.

To the best of our knowledge, we are the first to propose a gaze estimation system that combines model-based gaze vergence constraints with the self-supervised appearance based constraints available in an autoencoder architecture.

II. Related Work

We summarise related work in 3DMMs, facial 3D reconstruction, and appearance-based gaze estimation.

A. 3D morphable models

Face 3DMMs were introduced more than two decades ago by Blanz and Vetter [15] and perhaps is the most widely-employed technique in recent statistical 3D face modelling applications. Such 3DMMs model a linear or non-linear 3D facial space using a latent representation that can be constructed in a number of different ways. Examples include PCA [16], [17], [5], dictionary learning [18], wavelet decomposition [19], Gaussian mixture models [20] and neural nets [12]. Apart from the general face 3DMMs, there are several approaches that bring more focus to eyeball modelling. Bérard *et al.* [21] were the first to build a parametric model of eyeballs. The quality of this eye model is high, but the reconstruction process is semi-automatic. Wood *et al.* [4], [22] attempt to build an eye-region model of the single eye and use model fitting to estimate gaze. Ploumpis *et al.* [5] propose a method for building a complete head morphable model that includes eyeballs. The eye-region modelling is similar to the approach of Wood *et al.* and is blended into the head model. Amongst the publicly-available 3DMMs, we choose the FLAME [23] model to build our eye-region model since it has both eyeballs and can form a minimal eye-region model for both eyes, see Fig. 1, bottom left.

B. 3D Face Reconstruction from Monocular RGB

Reconstruction methods generally fall into three categories: generative, regression and generative-regression hybrid. Generative methods focus on generating a 3D model to fit the target data [24]. The approaches proposed by Wood *et al.* [4] and Ploumpis *et al.* [5] both fall into this category. Regression methods, recently popular due to deep learning advances, focus on regressing the model parameters directly via deep networks [25], [26]. The third category was firstly proposed by Tewari *et al.* [11], and adopted by many other works [27], [12]. This approach usually trains a joint autoencoder model that encodes the model parameters via the regression method, decodes the regressed model parameters, and reconstructs the original input. In contrast to our work, all of the mentioned face autoencoders focus on full face reconstruction and use only a mesh surface to model the eyeball, and the appearance of different gaze directions is not present or modelled via texture.

C. Appearance-based Gaze Estimation Methods

Recent appearance-based gaze estimation methods usually use a deep neural network to regress gaze directions. A number of datasets containing RGB images and gaze labels have been published that have enabled rapid progress. Along with the datasets, various appearance-based methods use different input representations, *e.g.*, eye images [28], [29], [30], face images [31], [14] or both [32], [33]. They also use different network architectures (CNNs, attention-based or combined) and different gaze representations (gaze originates from the eyeball or gaze originates from face centre) [10]. Prediction using high-resolution eye images has been the mainstream of this area, but face features were found to provide additional information for gaze estimation [34]. Recent work has focused on using super-resolution techniques for gaze estimation on very low resolution images [35]. Another recent work focuses on selecting relevant features in the latent code to facilitate good cross-domain performance [36].

On the more traditional side, model-based gaze estimation often involves a geometric eye model that is fitted to detected eye features, such as corneal reflections, pupil centre, iris contour, and eye landmarks [4], [5], [37].

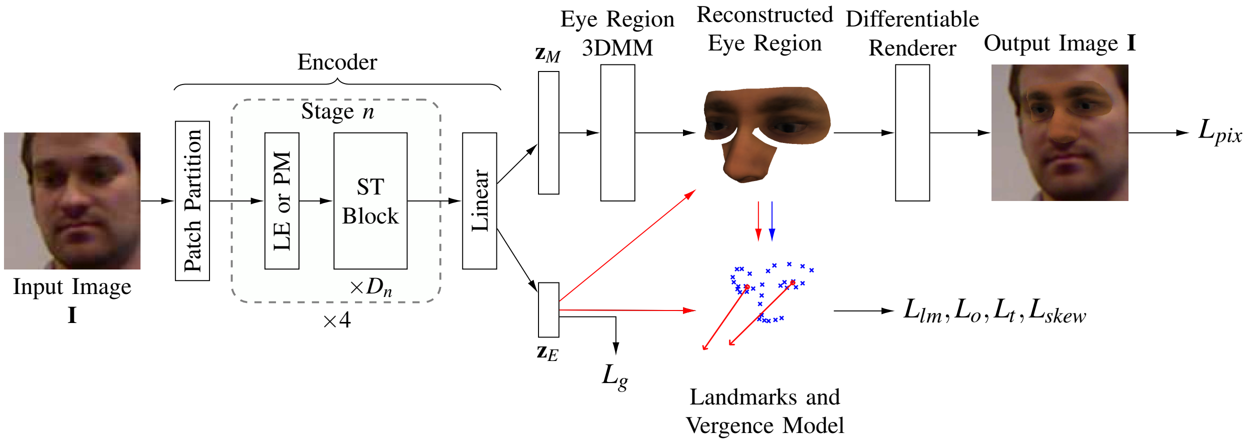

Fig. 2. Autoencoder with tiny version of the Swin Transformer. *LE* (Linear Embedding) is used in stage one and *PM* (Patch Merging) is used in stages $2-4$. *ST Block* is a Swin Transformer block. Blue points are 3D model landmarks projected to the image plane. 'L' terms show where training losses are generated. The latent vector Z_M encodes the 3D shape and color-texture of the eye region, while Z_E is vector that encodes the eyeball orientations (azimuth and elevation for both eyes) that correspond to the gaze direction.

III. Proposed Method

Our architecture (Fig. 2) shows that the raw image $\mathbf{I}$ is fed to the encoder to regress eye-region reconstruction parameters $\mathbf{z}_M$ and eye rotation parameters $\mathbf{z}_E$. The eye-region parameters are defined as follows: $\mathbf{z}_M = (\mathbf{z}_S, \mathbf{z}_A, \mathbf{r}, \mathbf{T}, f)^T$, where $\mathbf{z}_S$ are shape parameters, $\mathbf{z}_A$ are texture parameters, $\mathbf{r}, \mathbf{T}$ are head pose parameters describing rotation and translation respectively and f is the scale factor due to projection. We use the Swin transformer [38] as our encoder network. The eye-region reconstruction parameters $\mathbf{z}_M$ are used to reconstruct a textured eye-region 3D mesh, thus providing predicted 3D eyeball centres as gaze origins (eyeball vertex means), and a set of 2D projected landmarks for eye-region alignment. The eye rotation parameters $\mathbf{z}_E$ predict the gaze vectors for both eyes. Using the gaze origins and gaze vectors, we employ a geometric vergence model to constrain the gaze directions of both eyes jointly. Finally, we use a differentiable renderer to render the output image for pixel-wise comparison to the input. We now elaborate each pipline component.

A. Pipeline components

Encoder. The input image is first divided into non-overlapping patch tokens. This is followed by four Swin transformer blocks. We define D_i as the number of blocks at stage i, where $D_{1\ldots4} = (2,2,6,2)$, and use the *Tiny* network structure provided by the authors. For the first stage, the linear embedding module is applied before the transformer blocks, and for the other three stages, a patch merging module is applied before each set of transformer blocks to reduce the output dimensionality. These four stages jointly produce a feature map that is fed to a linear layer to regress a semantically-meaningful feature vector. This is then divided into two parts: eye-region reconstruction parameters $\mathbf{z}_M$ and both eyes' gaze directions, $\mathbf{z}_E$. Directions are defined by azimuth and elevation, hence $\mathbf{z}_E$ is a 4-vector.

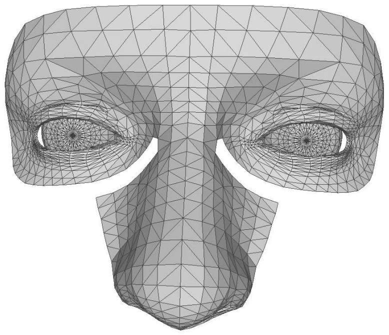

Fig. 3. The mean eye-region mesh, extracted from the FLAME model [23] and incorporated into our active-gaze 3DMM fitting system, which has rotatable eyeballs

Eye-Region 3D Morphable Model (3DMM). The eye-region 3DMM is constructed by selecting the relevant vertices and their topology from the FLAME [23] model. As shown in Fig. 3, both eyeballs, the eye-region and the nose are selected. Eyeballs, which are simple spheres, are used to model gaze directions, eyeball sizes, and inter-ocular distances. The eye-region contains 22 landmarks on the eyebrows and eye contours, which is used to model eyeball positions and head poses. We omit the remaining parts of the FLAME head model, firstly to enable a more compact and efficient learning process, and secondly since they have much higher variance in features (*e.g.* mouth/jaw variations due to speech and/or facial expressions) that are not relevant to gaze modelling, and may introduce confounding factors. Notably, the largely rigid nose area, which contains nine landmarks on the nose ridge and the philtrum area, is added to strengthen the head pose prediction. We also use the albedo model presented by [39] to enable differentiable rendering of the eye-region model.

We reconstruct the 3DMM's shape $\mathbf{S} \in \mathbb{R}^{N\times 3}$ from the standard FLAME-basis shape parameters $\mathbf{z}_S$ and the texture $\mathbf{A} \in \mathbb{R}^{512\times 512\times 3}$ from texture parameters $\mathbf{z}_A$ as follows:

$$\mathbf{S} = \mu_S + \mathbf{U}_S\mathbf{z}_S \tag{1}$$

$$\mathbf{A} = \mu_A + \mathbf{U}_A\mathbf{z}_A, \tag{2}$$

where N is the number of vertices in the eye-region shape model, $\mu_{\{S,A\}}$ and $\mathbf{U}_{\{S,A\}}$ are the mean and principal components provided by the shape and texture 3DMMs respectively. Then the eye-region shape $\mathbf{S}$ is transformed with rotation $\mathbf{R}$, translation $\mathbf{T}$ and scale f to the camera coordination system by:

$$\mathbf{S}' = f\mathbf{S}\mathbf{R}^T + \mathbf{1}\mathbf{T}, \tag{3}$$

where $\mathbf{R} \in \mathrm{SO}(3)$ is the rotation matrix derived from the Euler angle rotation $\mathbf{r} \in \mathbb{R}^3$ and $\mathbf{1} \in \mathbb{R}^{N\times 1}$ is the vector of all ones. Finally, given the camera calibrations are available, we construct a full perspective projection $\mathbf{\Pi} \in \mathbb{R}^3 \to \mathbb{R}^2$ to project the eye-region shape in 3D camera space $\mathbf{S}'$ to image plane, thus obtaining the predicted 2D landmarks $\hat{\mathscr{L}}$ on image plane. We use a differentiable renderer DR implemented by PyTorch3D [40], with the same projection model, to form image $\hat{\mathbf{I}}$ as:

$$\hat{\mathbf{I}} = \mathrm{DR}\left(\mathbf{S}', \mathbf{A}, \mathbf{\Pi}\right). \tag{4}$$

All previous works on 3D face reconstruction that involve differentiable rendering assume a Lambertian surface. This is not well-suited to the eyeball due to its moisture, which causes specularities. Our experiments shows that the geometric vergence constraints contribute significantly to gaze estimation accuracy, thus we choose the ambient Phong lighting model.

With the reconstructed 3D eye-region, we form the 3D gaze origin loss function L_o as

$$L_o = \|\hat{\mathbf{o}} - \mathbf{o}\|_1^1, \tag{5}$$

where $\mathbf{o}$ is a 3D ground truth gaze origin provided by the training dataset (if available) and $\hat{\mathbf{o}}$ is some point derived by the eye-region shape; *e.g.* a predicted eyeball centre is obtained by averaging all eyeball vertices. With such a design, our method becomes universally applicable to any gaze origin definition, as provided by the dataset; for example, both eyeball-centered and face-centered have been used in the literature. This obviates the conversion step described by Chen *et al.* [10] that converts gaze ground truth between datasets using different gaze representations.

With the projection model 2D projected landmarks can be obtained, giving the 2D landmark loss function L_{lm} as:

$$L_{lm} = \|\hat{\mathscr{L}} - \mathscr{L}\|_2^2, \tag{6}$$

where $\mathscr{L}$, the ground truth 2D landmarks, are either provided by the dataset or generated before training using PyTorch Face Landmark [41] with a pre-trained MobileNetV2 [42] as the backbone network. The predicted 2D landmarks $\hat{\mathscr{L}}$ are obtained by projecting selected vertices in the eye-region shape $\mathbf{S}'$ onto the image plane via the perspective projection, $\mathbf{\Pi}$. We employ the Multi-PIE [43] definition of 68 face landmarks and select 31 corresponding points on the eye-region 3DMM and input images.

Finally, the pixel loss L_{pix} for rendered eye-region images is formed as:

$$L_{pix} = \|\hat{\mathbf{I}} - \mathbf{I}\|_2^2. \tag{7}$$

Vergence model. The predicted gaze rotations $\mathbf{z}_E = (\mathbf{r}_l, \mathbf{r}_r)^T$ are azimuths and elevations for both eyes (*e.g.* $\mathbf{r}_l = (r_{le}, r_{la})^T$). An eyeball rotation matrix $\mathbf{R}^e_{\{l,r\}}$ is derived for each eye using each pair of these Euler rotation angles. We assume that the gaze direction is a vector originating from the centre of the eyeball, and pointing towards the iris centre and we initialise this in the global camera frame to be pointing towards the camera *i.e.* $\mathbf{g}_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$. Thus, *face frame* gaze vectors for both eyes are calculated as: $\mathbf{g}_i = \mathbf{R}^e_i \mathbf{g}_0, i \in \{l, r\}$. They originate from both eyeballs' centre $\mathbf{o}_{\{l,r\}}$ respectively. The eyeball rotation matrices $\mathbf{R}^e_{\{l,r\}}$ are also applied to the *front-facing* (*i.e.* unrotated) eyeball shapes of the reconstructed 3D eye-region shape to rotate the eyeballs to produce a plausible appearance.

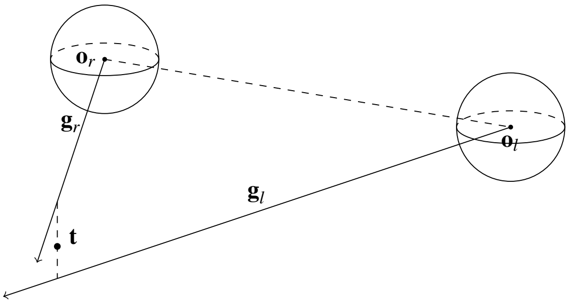

Fig. 4. The vergence model of gaze for the active-gaze 3DMM, showing eyeball origins ($\mathbf{o}_{l,r}$), gaze directions ($\mathbf{g}_{l,r}$) and viewing target $\mathbf{t}$ in the global camera frame. In general, the regressed gaze directions are skew and the loss function penalises this lack of coplanarity. Note that all parameters are defined in the camera coordinate system.

As shown in Fig. 4, we equip our system with geometric constraints so that both eye gazes are mutually constraining each other via a mutual gaze target $\hat{\mathbf{t}}$. Due to the nature of human gazes, there are three underlying constraints for this vergence model: i) both gaze vectors are directed in a forward direction away from the head; ii) the gaze vectors are coplanar; iii) the gaze vectors intersect at the gaze target $\hat{\mathbf{t}}$, unless they are parallel. These three constraints can be satisfied during the process of calculating the gaze target $\hat{\mathbf{t}}$, which is defined as the closest point between the two gaze vectors. We define $\mathbf{K}_i = \mathbf{o}_i + k_i\mathbf{g}_i, i \in \{l, r\}$ as the two end points of the shortest segment connecting left and right gazes. Therefore,

$$\hat{\mathbf{t}} = (\mathbf{K}_l + \mathbf{K}_r)/2. \tag{8}$$

Since the shortest segment must be perpendicular to both gaze vectors, we can derive the shortest distance d as:

$$d := \|\mathbf{K}_l - \mathbf{K}_r\| = k_{lr}\left(\mathbf{g}_r \times \mathbf{g}_l\right), \tag{9}$$

where k_l, k_r and k_{lr} can be solved by:

$$\begin{bmatrix} k_l & k_r & k_{lr} \end{bmatrix}^T = \begin{bmatrix} \mathbf{g}_l & -\mathbf{g}_r & \mathbf{g}_r \times \mathbf{g}_l \end{bmatrix}^{-1} (\mathbf{o}_r - \mathbf{o}_l). \tag{10}$$

We design three loss terms based on the underlying constraints of the geometric vergence model. Firstly, the gaze skew loss, $L_{skew} = d^2$, encourages the two gaze vectors to be coplanar. Secondly, the predicted gaze target, $\hat{\mathbf{t}}$, along with the 3D ground truth target, $\mathbf{t}$, forms a gaze target loss $L_t = \left\|\hat{\mathbf{t}} - \mathbf{t}\right\|_1^1$. Finally, a gaze pose loss is given as $L_g = \left\|\mathbf{z}_E - \mathbf{r}_{gt}\right\|_1^1$, where $\mathbf{r}_{gt}$ is the ground truth eyeball rotation. All of these losses reduce gaze error, while preventing the physically impossible case of the gaze being directed into and behind the head.

Regulariser. In addition to the previously stated loss function terms, we employ a regulariser on the 3D eye-region shape and texture latent code $\mathbf{z}_S$ and $\mathbf{z}_A$ to encourage the reconstructed eye-region shape and texture to stay within the model space. The regulariser is defined as follows:

$$L_{reg} = \|\mathbf{z}_S\|_2^2 + \|\mathbf{z}_A\|_2^2. \tag{11}$$

Complete loss function. Finally, all the losses are combined linearly as $L = \Lambda^T L_{vec}$ where $\Lambda = [\lambda_1 \dots \lambda_7]^T$ are the hyperparameter weights required to balance each loss component in the loss vector $L_{vec} = [L_{pix}, L_{lm}, L_o, L_t, L_{skew}, L_g, L_{reg}]^T$.

B. Implementation Details

For our Swin transformer encoder, we use the *tiny* configuration with the pretrained weights on ImageNet [44]. We use the Adam optimiser [45] with learning rate set to 5×10^{-5} and weight decay set to 1×10^{-4} to train our model for 70 epochs. The hyper-parameters $\lambda_1 \dots \lambda_7$ to weight all loss function components are set to 1, 0.5, 1×10^3, 2.5×10^3, 5×10^2, 1 and 5×10^{-2} respectively for the Eyediap dataset.

Tuning hyperparameters was straightforward, as our system is highly constrained by geometry. We initialised all the weights to the same order of magnitude (0-1) and then we varied each independently to find some improvement. We also tried auto-tuning [46] and got almost as good performance (auto 4.8 degrees vs manual 4.55).

IV. Evaluation

We employ two datasets for evaluation: Eyediap and ETH-XGaze. Eyediap [13] is a dataset containing videos of 16 subjects looking at various targets. We use the floating ball target videos. A static head pose session and a dynamic head pose session are recorded for each subject, resulting in 28 sessions of, on average, 2701 frames per session. We use the *low-resolution* VGA version (640×480) for our experiments. This low resolution, along with the fact that we are using dynamic head poses as well as static ones (some authors use static only), means that we are aiming to solve the hardest version of the gaze estimation problem over this dataset.

During training and testing, we utilise all validated frames except those not detected by the face landmark localisation algorithm. We perform cross-subject evaluations on this dataset, using a leave-two-subjects-out strategy by using two subjects' both static and dynamic head pose sessions as the test set, and the remainder as the training set. We train on $\sim$ 61k frames and test on $\sim$ 14k frames.

ETH-XGaze [14] is a large dataset covering a wide range of head poses, with over one million images from 110 participants. The evaluation on the test set is performed on an online platform provided by the authors. We use the standard 15 participants as the within-domain test set and the standard set of 80 subjects as the training set. We also use the landmarks provided by this dataset to train our model.

The loss function we defined introduces seven hyperparameters, which may suggests a difficult tuning process. However, many losses are imposing the same model restriction that help stabilise the training process, we group empirically correlated losses into 3 groups and treat each group as a single joint loss, to make hyperparameter tuning process simpler. We also perform ablation studies with the three groups in Sec. V.

A. Quantitative Evaluation

We compare our results with some previous methods with the commonly-adopted angular error metric. This error metric measures the angle between the predicted gaze vector and the ground truth gaze vector. Results on Eyediap for the *floating ball* experiment are given in Table I and show that our method gives the lowest mean error. We also include a baseline which uses the Swin transformer to regress gaze rotation only. This demonstrates that incorporation of our geometric model improves accuracy over an equivalent *gaze-only* system. This is studied further in the ablation studies in section V.

For Table I (Eyediap) standardisation of evaluation is impossible as Eyediap is a *dataset* but *not a benchmark*. There is no standardised train/test split across the published literature. Also, different papers use different video session types (static head pose only or static-plus-dynamic head) and different image resolutions (640x480 vs 1920x1080). However, unlike some competing works in Table I, we solve *the most difficult problem* in that we both use the *smaller image resolution* and, like [6], we evaluate over *both static and dynamic head poses* - not just static only. Furthermore, we evaluate on around 14,000 frames. Despite the hard version of the problem over many frames, we still get state-of-the-art performance, albeit in the context of cautiously reporting the accuracy of competing systems for reference against ours.

For the more recent ETH-XGaze dataset, we show competitive gaze results, see Table II, compared to purely appearance-based methods, while providing much richer information via full parameterisation of our active-gaze 3DMM. In other words, unlike other methods, we also get the eye region shape and texture and the 3D eyeball positions. For the results in Table II (ETH-XGaze dataset), the training and test sets are *identical* across all methods and therefore they are *directly comparable i.e.* the standard ETH-XGaze evaluation benchmark is employed. Note also that we do not exploit modelling of the kappa angle (the angle between the visual axis and the pupillary axis) and only employ a

TABLE I. Angle error (°) on gaze vectors originating from the eyeballs: Eyediap dataset, floating ball target experiment. Note that, unlike some other systems, our system solves the most difficult problem in that it employs only low resolution images on both static and dynamic poses, over 14K test images. Note that some ‘appearance-based’ methods may use landmarks, eg [29] and that there are some variations in (pre)training data volume.

	Method	mean ± std	median
Appearance-based Methods	Palmero *et al.* [6]	5.19	\
	Zhang *et al.* [34]$^{\#}$	6.76	\
	Cheng *et al.* [31]	5.17	\
	Zhang *et al.* [47]	7.37	\
	Sinha *et al.* [29]	4.62 ± 2.93	\
	Gaze360 [48]$^{\#}$	5.58	\
	RT-Gene [49]$^{\#}$	6.30	\
	Dilated-Net [50]$^{\#}$	6.57	\
	Baseline	5.25 ± 3.58	4.45
Model-based Methods	PR-ALR [13]*	8.1	\
	Wood *et al.* [4]*	9.44	8.63
	Ploumpis *et al.* [5]	8.85	\
	Park *et al.* [37]$^{*+}$	11.9	\
Combined Method	Ours	$\mathbf{4.55 \pm 3.29}$	**3.82**

* Eval. on static head pose only. $^{\#}$ Converted from face gaze by Cheng *et al.* [10]. $^{+}$ Trained on synthetic data only.

lightweight backbone network with relatively low training demands. Finally, note that Cheng *et al.* [31] is *not* included in Table II as they only pretrain on ETH-XGaze, and don’t evaluate on that dataset.

TABLE II. Angle error (°) ETH-XGaze dataset.

Method	mean	std
PureGaze [51]	6.79	\
Zhang *et al.* [14]	4.50	\
Gaze360 [48]	4.46	\
Zhang *et al.* [34]	7.38	\
Cai *et al.* [52]	**3.11**	\
Ours	5.80	4.95

There are two types of task for gaze vector estimation: i) the gaze originates from the eyes and ii) the gaze originates from faces [10]. While our method is successful on the eye gaze task, it does not have an advantage from accurately predicting the face gaze. This is due to only one gaze vector being available and our model takes advantage of the correlations between both gaze vectors originating from the eyes. In the Eyediap ablation study, we found that the gaze origin (*i.e.* eyeball centre) loss is required for better performance (there is a more than 60% increase in error without this loss component), and the ETH-XGaze dataset *does not* provide any eyeball centre information. However, our method does not require explicit conversion between the eye gaze task and the face gaze task. Moreover, during training our method approaches the ground truth very quickly and we obtain our results with training for only 20 epochs on 10% of the training set (approx. 60,000 images) randomly sampled every batch.

We now report our reconstructed model’s quality. Our face patches on the Eyediap dataset have 96×96 pixels, our predicted face landmarks are filtered manually to remove frames with obstacles in front of the face. The average landmark error in pixels is 4.84 pixels per landmark. We further normalise pixel landmark errors by dividing the distance between the left eye’s left corner and the right eye’s right corner, which results in a proportion of 0.113.

B. Cross-dataset comparative performance

It is instructive to consider the comparative performance of our system across the two datasets: why is our Eyediap gaze performance better than ETH-Gaze and what are the implications of this? First, our method is explicitly 3D. Rich, explicit 3D information concerning eyeball positions and eye-region shape is often more useful than gaze direction only. In terms of gaze accuracy, such a system is always going to be more successful when it has strong 3D supervision, which is the case for Eyediap but not for ETH-Gaze. In ETH-XGaze, we only have supervisory 3D information from a 3D morphable model fitted to some facial landmarks on a *single* 2D image, where overall 3D scale is not accurate. The ETH-XGaze dataset capture used 18 views so much more accurate 3D information could have been supplied with this dataset. If it was, we would expect a lowering of our overall error. However, despite poor quality 3D supervision, we still obtain competitive results compared with systems directly regressing gaze without use of an explicit geometric model. Whilst it is true that strong 3D supervision for state-of-the-art performance is some form of limitation, multi-view 3D and RGB-D are now well-developed, accurate and accessible, and this conveys a significant advantage on our system - a much richer, more explicit and more useful explanation of image content.

C. Inference speed comparisons

To our knowledge, our system has the fastest reported inference rate at 89fps (averaged over 4403 frames) for gaze computation. This is without rendering the fitted model. Comparisons with other system’s inference speed are given in Table III, along with the reported computational platforms. We run on a single RTX 3090 graphics card. Our system has a significant advantage in applications where high frame rate ($>$ 30fps) cameras are required. For example, our system may be useful to interpolate when rapid eye saccades happen. Indeed, three gaze directions are capable of being captured for a 40ms saccade. However, this assumes that camera image blur is not a significant issue. A video of real-time performance is presented in the supplementary, using the same colour scheme as the qualitative results in Fig. 5.

TABLE III. Inference speed, frames/sec (fps), compared to other systems. The implementation platform reported in each paper is specified.

Method	fps	Reported platform
Park *et al.* [37]	26	Intel i7-4770 + Nvidia 1080Ti
Fischer *et al.* [49]	25.3	Intel i7-6900K + Nvidia 1070
Wood *et al.* [4]	0.27	3.3Ghz CPU, GTX 660 GPU
Ploumpis *et al.* [5]	0.2	Intel Core i7 3.8 GHz + RTX 2080 Ti
Ours	**89**	RTX 3090 card

D. Qualitative Evaluation (Eyediap)

For qualitative evaluation on Eyediap, results are presented in Fig. 5. This shows predicted gaze vectors relative to their ground truth, predicted locations of vertices on the 3D model (projected into the image plane i.e. predicted landmark locations) and, in the final column, the rendered 3DMM using the chosen albedo model is shown.

Although our estimated gaze vectors (red) are consistent with the ground truth directions (green), we note that the rendered models in the final column are not visually appealing when compared to state of the art photorealism. For example, due to the nature of the albedo model we used, the eyeball's sclera region appears to be slightly cloudy. However, the photometric loss generated by the rendering serves to regularize the 3D model fitting and the gaze estimation accuracy is not adversely affected.

V. Ablation Studies

Although a typical ablation study only removes one loss component at a time in order to isolate the effect of that component, we take a different approach here. We observe that various subsets of our loss components are entangled and serve the same task within our system, where each task aids gaze estimation. Therefore, we perform a higher-level ablation study to investigate the utility of these tasks in the context of the overall gaze accuracy, with the results presented in Table IV.

We used a randomly selected subject's static and dynamic head pose sessions (approx 7K frames over pose variations) as the test set for all ablation experiments.

The three tasks, along with the loss components that serve them are as follows:

1) appearance-based gaze estimation with L_g (used by itself in the *baseline model* system)
2) constraining eye vergence with L_t, L_o and L_{skew} (used by itself in the *vergence model* system)
3) eye-region reconstruction with L_{pix}, L_{lm} and L_{reg} (combined with tasks 1 and 2 above, and used for our final system (*Ours*))

First, we construct a baseline that comprises only our vision backbone network (*i.e.* Swin transformer) which predicts two eyeball rotations. It is trained with only the gaze pose loss function L_g, thus it employs Task 1 (above) only. We denote this experiment as *Baseline model* in Table IV.

Then we construct our system with the vergence model only, *i.e.* the model solves Task 2 only. Since the predicted gaze origins (*i.e.* eyeball centres) are not available if no 3D eye-region model is reconstructed, we predict the eyeball centres directly using the backbone network. This experiment is denoted as *Vergence model* in Table IV.

Then we report our proposed method with all loss terms (*i.e.* aimed at solving all three tasks simultaneously), denoted as *Ours*, shown in the final row of Table IV.

Starting from our full system, we remove only the loss term L_o to let the model learn without ground truth eyeball positions. This experiment is denoted as *Ours w/o L_o*.

Finally, we evaluate the value of the Swin transformer against a former popular vision backbone ResNet-18 [53]. This is denoted *Ours - ResNet18* in Table IV.

TABLE IV. Angle error (°) on subject 15 from Eyediap dataset for the ablation study

Method	Tasks solved	Losses used (max 7)	Mean ± Std (degs)	Med (degs)
Baseline model	1	1 (L_g)	5.60 ± 3.28	5.01
Vergence model	2	3 (L_t, L_o, L_{skew})	4.80 ± 3.07	4.23
Ours w/o L_o	1,2,3	6	6.64 ± 4.92	5.26
Ours-ResNet18	1,2,3	7	4.94 ± 3.16	4.34
Ours	1,2,3	7	$\mathbf{4.11 \pm 2.93}$	**3.42**

The performance of our system (*Ours*) shows that combining Task 3 (eye region reconstruction) with Task 1 (appearance) and Task 2 (vergence) gives better performance than using only Task 1 or Task 2 alone, thus showing the effectiveness of our *multi-task* method.

The vanilla appearance-based method (*baseline*) cannot perform competitively when using low resolution images. Indeed, our vergence model, even when operating on its own, performs better than the appearance based baseline.

We observe that the gaze origin loss L_o is important. It provides guidance to both gaze direction and 3D eye-region reconstruction. Since the ETH-XGaze dataset does not provide ground truth 3D eyeball centres, this explains why the results on ETH-XGaze dataset are not as good as the results on the Eyediap dataset.

The attention mechanism of the Swin transformer [38] vision backbone network may be aiding the gaze estimation task. However, since the Swin transformer has around 29 million trainable parameters, while ResNet18 only has around 11.5 million, we cannot disentangle the attention effect from the overall size of the backbone network.

VI. Conclusions

Our approach reconstructs the 3D eye-region as well as the gaze direction by exploiting the advantages of both appearance-based and model-based techniques. Results show state-of-the-art accuracy on Eyediap as well as fast, real-time inference.

Our results on ETH-XGaze are competitive, despite the fact that strong 3D supervision is not provided with the dataset in terms of eyeball origins. By utilising a state-of-the-art vision backbone (the Swin transformer), we close the gap on the gaze estimation task where model-based methods lack the raw feature extraction ability.

In addition to HCI applications, our work can be further applied to inter-ocular distance prediction, ear-to-ear face region modelling, with associated accurate and high-speed gaze estimation. As such, our work has the potential to contribute to human eye-region understanding and can serve as a useful design tool for various categories of eyewear.

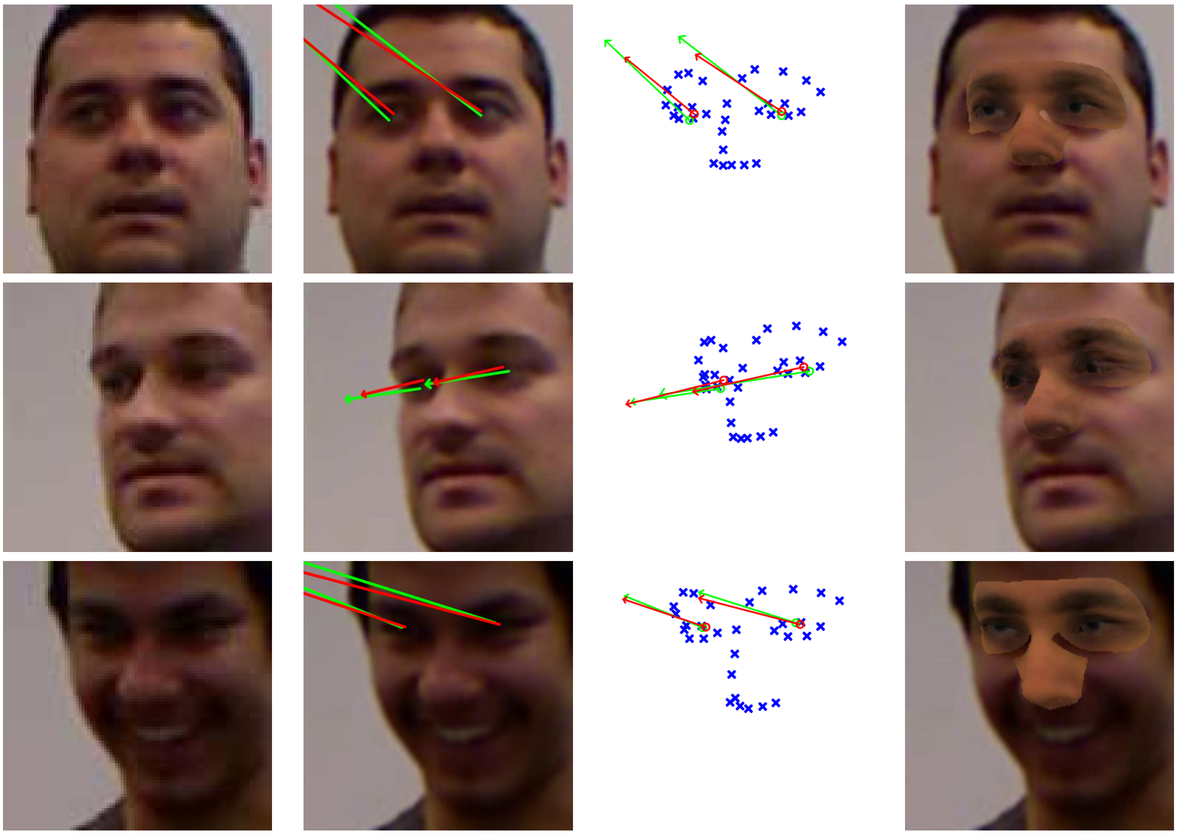

Fig. 5. Three subjects (one subject per row) focusing on a floating target (orange ball) in the Eyediap dataset. For each row: (i) input image (far left); (ii) input image with gaze vectors superimposed (ground truth gaze vectors are green, predicted gaze vectors are red); (iii) projected 3D model with predicted landmark positions (blue crosses); (iv) rendered eye-region model superimposed on input image (far right). Note that the predicted gaze vectors (red) have strong agreement with ground truth (green). However, the rendered model on the right is not competitive in terms of photorealism. This is not a goal of our system, which merely employs the rendered image as a 3D model fitting regularizer using a photometric loss.

References

[1] D. Li, D. Winfield, D. Parkhurst, Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops, 2005, pp. 79–79.

[2] K. A. Funes Mora, J.-M. Odobez, Geometric generative gaze estimation (g3e) for remote rgb-d cameras, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1773–1780.

[3] H. Kaur, S. Jindal, R. Manduchi, Rethinking model-based gaze estimation, Proc. ACM Comput. Graph. Interact. Tech. 5 (2) (2022).

[4] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, A. Bulling, A 3D morphable eye region model for gaze estimation, in: European Conference on Computer Vision, Springer, 2016, pp. 297–313.

[5] S. Ploumpis, E. Ververas, E. O'Sullivan, S. Moschoglou, H. Wang, N. Pears, W. Smith, B. Gecer, S. P. Zafeiriou, Towards a complete 3D morphable model of the human head, IEEE transactions on pattern analysis and machine intelligence (2020).

[6] C. Palmero, J. Selva, M. A. Bagheri, S. Escalera, Recurrent CNN for 3D gaze estimation using appearance and shape cues, in: British Machine Vision Conference, 2018.

[7] K. Wang, R. Zhao, Q. Ji, A hierarchical generative model for eye image synthesis and eye gaze estimation, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 440–448. doi:10.1109/CVPR.2018.00053.

[8] Y. Yu, G. Liu, J.-M. Odobez, Deep multitask gaze estimation with a constrained landmark-gaze model, Vol. 11130 of Lecture Notes in Computer Science, SPRINGER INTERNATIONAL PUBLISHING AG, Cham, 2018, pp. 456–474.

[9] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521 (7553) (2015) 436–444.

[10] Y. Cheng, H. Wang, Y. Bao, F. Lu, Appearance-based gaze estimation with deep learning: A review and benchmark, arXiv preprint arXiv:2104.12668 (2021).

[11] A. Tewari, M. Zollhofer, H. Kim, P. Garrido, F. Bernard, P. Perez, C. Theobalt, Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 1274–1283.

[12] L. Tran, X. Liu, Nonlinear 3D face morphable model, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7346–7355.

[13] K. A. Funes Mora, F. Monay, J.-M. Odobez, Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras, in: Proc Symposium on Eye Tracking Research and Applications, 2014, pp. 255–258.

[14] X. Zhang, S. Park, T. Beeler, D. Bradley, S. Tang, O. Hilliges, Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation, in: European Conference on Computer Vision (ECCV), 2020.

[15] V. Blanz, T. Vetter, A morphable model for the synthesis of 3D faces, in: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, 1999, pp. 187–194.

[16] J. Booth, E. Antonakos, S. Ploumpis, G. Trigeorgis, Y. Panagakis, S. Zafeiriou, 3D face morphable models" in-the-wild", in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 48–57.

[17] H. Dai, N. Pears, W. Smith, C. Duncan, Statistical modeling of

craniofacial shape and texture, International Journal of Computer Vision 128 (2) (2019) 547–571.

[18] C. Ferrari, G. Lisanti, S. Berretti, A. Del Bimbo, Dictionary learning based 3D morphable model construction for face recognition with varying expression and pose, in: 2015 International Conference on 3D Vision, IEEE, 2015, pp. 509–517.

[19] A. Brunton, T. Bolkart, S. Wuhrer, Multilinear wavelets: A statistical shape space for human faces, in: European Conference on Computer Vision, Springer, 2014, pp. 297–312.

[20] P. Koppen, Z.-H. Feng, J. Kittler, M. Awais, W. Christmas, X.-J. Wu, H.-F. Yin, Gaussian mixture 3D morphable face model, Pattern Recognition 74 (2018) 617–628.

[21] P. Bérard, D. Bradley, M. Gross, T. Beeler, Lightweight eye capture using a parametric model, ACM Transactions on Graphics (TOG) 35 (4) (2016) 1–12.

[22] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, A. Bulling, A 3D morphable model of the eye region, Optimization 1 (2016) 0.

[23] T. Li, T. Bolkart, M. J. Black, H. Li, J. Romero, Learning a model of facial shape and expression from 4D scans, ACM Transactions on Graphics, (Proc. SIGGRAPH Asia) 36 (6) (2017) 194:1–194:17.

[24] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, M. Nießner, Face2face: Real-time face capture and reenactment of rgb videos, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2387–2395.

[25] Y. Feng, F. Wu, X. Shao, Y. Wang, X. Zhou, Joint 3D face reconstruction and dense alignment with position map regression network, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 534–551.

[26] E. Richardson, M. Sela, R. Or-El, R. Kimmel, Learning detailed face reconstruction from a single image, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1259–1268.

[27] H. Sun, N. Pears, H. Dai, A human ear reconstruction autoencoder, in: 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Springer International Publishing, 2021.

[28] S. Park, E. Aksan, X. Zhang, O. Hilliges, Towards end-to-end video-based eye-tracking, in: European Conference on Computer Vision (ECCV), 2020.

[29] N. Sinha, M. Balazia, F. Bremond, Flame: Facial landmark heatmap activated multimodal gaze estimation, in: 2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2021.

[30] Y. Yu, J.-M. Odobez, Unsupervised representation learning for gaze estimation, in: Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition, 2020, pp. 7314–7324.

[31] Y. Cheng, F. Lu, Gaze estimation using transformer, arXiv preprint arXiv:2105.14424 (2021).

[32] Y. Bao, Y. Cheng, Y. Liu, F. Lu, Adaptive feature fusion network for gaze tracking in mobile tablets, in: 2020 25th Int Conf on Pattern Recognition (ICPR), IEEE, 2021, pp. 9936–9943.

[33] Y. Cheng, S. Huang, F. Wang, C. Qian, F. Lu, A coarse-to-fine adaptive network for appearance-based gaze estimation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 10623–10630.

[34] X. Zhang, Y. Sugano, M. Fritz, A. Bulling, It's written all over your face: Full-face appearance-based gaze estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 51–60.

[35] Y.-S. Yun, N. Y, K. H. H, K. H-L, Y. S. B, HAZE-Net: High-frequency attentive super-resolved gaze estimation in low-resolution face images, in: ACCV, 2022.

[36] L. I, J. J-S, K. H. H, N. Y, Y. S. B, Latentgaze: Cross-domain gaze estimation through gaze-aware analytic latent code manipulation, in: ACCV, 2022.

[37] S. Park, X. Zhang, A. Bulling, O. Hilliges, Learning to find eye region landmarks for remote gaze estimation in unconstrained settings, in: Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, Association for Computing Machinery, New York, NY, USA, 2018. doi:10.1145/3204493.3204545.
URL https://doi.org/10.1145/3204493.3204545

[38] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 10012–10022.

[39] W. A. P. Smith, A. Seck, H. Dee, B. Tiddeman, J. Tenenbaum, B. Egger, A morphable face albedo model, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5011–5020.

[40] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, G. Gkioxari, Accelerating 3D deep learning with PyTorch3D, arXiv:2007.08501 (2020).

[41] C. Chen, PyTorch Face Landmark: A fast and accurate facial landmark detector, open-source software available at https://github.com/cunjian/pytorch_face_landmark (2021).

[42] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: CVPR, 2018.

[43] R. Gross, I. Matthews, J. Cohn, T. Kanade, S. Baker, Multi-pie, in: 2008 8th IEEE International Conference on Automatic Face Gesture Recognition, 2008, pp. 1–8. doi:10.1109/AFGR.2008.4813399.

[44] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012) 1097–1105.

[45] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[46] A. Kendall, Y. Gan, R. Cipolla, Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, in: Proc of the IEEE conference on computer vision and pattern recognition, 2018.

[47] X. Zhang, Y. Sugano, M. Fritz, A. Bulling, Appearance-based gaze estimation in the wild, in: Proc of the IEEE conference on computer vision and pattern recognition, 2015, pp. 4511–4520.

[48] P. Kellnhofer, A. Recasens, S. Stent, W. Matusik, A. Torralba, Gaze360: Physically unconstrained gaze estimation in the wild, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6912–6921.

[49] T. Fischer, H. J. Chang, Y. Demiris, Rt-gene: Real-time eye gaze estimation in natural environments, in: Proc of the European Conf on Computer Vision (ECCV), 2018, pp. 334–352.

[50] Z. Chen, B. E. Shi, Appearance-based gaze estimation using dilated-convolutions, in: Asian Conference on Computer Vision, Springer, 2018, pp. 309–324.

[51] Y. Cheng, Y. Bao, F. Lu, Puregaze: Purifying gaze feature for generalizable gaze estimation, arXiv:2103.13173 (2021).

[52] X. Cai, B. Chen, J. Zeng, J. Zhang, Y. Sun, X. Wang, Z. Ji, X. Liu, X. Chen, S. Shan, Gaze estimation with an ensemble of four architectures, arXiv preprint arXiv:2107.01980 (2021).

[53] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.