A 3D morphable eye region model for gaze estimation

Erroll Wood[1] Tadas Baltrušaitis[2] Louis-Philippe Morency[2]
Peter Robinson[1] Andreas Bulling[3]

[1] University of Cambridge, UK, {eww23,pr10}@cl.cam.ac.uk
[2] Carnegie Mellon University, USA, {tbaltrus,morency}@cs.cmu.edu
[3] Max Planck Institute for Informatics, Germany, bulling@mpi-inf.mpg.de

Abstract. Morphable face models are a powerful tool, but have previously failed to model the eye accurately due to complexities in its material and motion. We present a new multi-part model of the eye that includes a morphable model of the facial eye region, as well as an anatomy-based eyeball model. It is the first morphable model that accurately captures eye region shape, since it was built from high-quality head scans. It is also the first to allow independent eyeball movement, since we treat it as a separate part. To showcase our model we present a new method for illumination- and head-pose–invariant gaze estimation from a single RGB image. We fit our model to an image through analysis-by-synthesis, solving for eye region shape, texture, eyeball pose, and illumination simultaneously. The fitted eyeball pose parameters are then used to estimate gaze direction. Through evaluation on two standard datasets we show that our method generalizes to both webcam and high-quality camera images, and outperforms a state-of-the-art CNN method achieving a gaze estimation accuracy of 9.44° in a challenging user-independent scenario.

Keywords: Morphable model, gaze estimation, analysis-by-synthesis

1 Introduction

The eyes and their movements convey our attention, indicate our interests, and play a key role in communicating social and emotional information [1]. Estimating eye gaze is therefore an important problem for computer vision, with applications ranging from facial analysis [2] to gaze-based interfaces [3,4]. However, estimating gaze remotely under unconstrained lighting conditions and significant head-pose is a yet-outstanding challenge. Appearance-based methods that directly estimate gaze from an eye image have recently improved upon person- and device-independent gaze estimation by learning invariances from large amounts of labelled training data. In particular, Zhang et al. trained a multi-modal convolutional neural network with 200,000 images collected during everyday laptop use [5], and Wood et al. rendered over one million synthetic training images with artificial illumination variation [6]. It has been shown that the performance of such methods heavily depends on the head pose and gaze

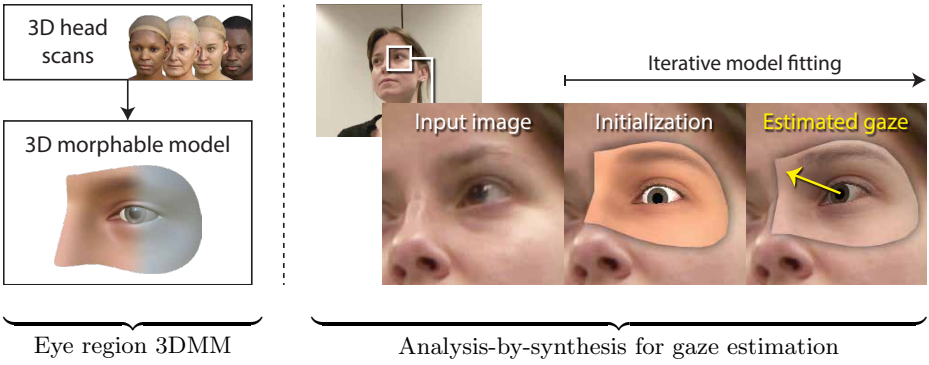

Fig. 1: Our generic gaze estimator is enabled by two contributions. First, a novel 3DMM of the eye built from high quality head scans. Second, a new method for gaze estimation – we fit our 3DMM to an image using analysis-by-synthesis, and estimate gaze from fitted parameters.

range that the training data covers – results are best when the training data closely matches the desired test condition [7]. This means a gaze estimator trained in one scenario does not perform well in another. Instead, we would prefer a generic gaze estimator that performs well in all conditions.

3D morphable models (3DMM) are a powerful tool as they combine a model of face variation with a model of image formation, allowing pose and illumination invariance. Since their introduction [8], they have become an established method for many tasks including inverse rendering [9,10], face recognition [11,12], and expression re-targeting [13]. Given a face image, such systems use model fitting to discover the most likely shape, texture, expression, pose, and illumination parameters that generated it. However, previous work has failed to accurately model the eyes, portraying them as a static geometry [8,11], or removing them from the face entirely [14,13]. This is a result of two complexities that are not handled by current methods: 1) The eyeball's materials make it difficult to reconstruct in 3D, leading to poor correspondence and loss of detail in the 3DMM, 2) Previous work uses blendshapes to model facial expression – a technique not compatible with independent eyeball movement. We make two specific contributions:

An eye region 3DMM Our first contribution is a novel multi-part 3DMM that includes the eyeball, allowing us to accurately model variation in eye appearance and eyeball pose (see Figure 1 left). Recent work presented a morphable shape model of the eye region, but did not capture texture variation [6]. We constructed a 3DMM of the facial eye region by carefully registering a set of high-quality 3D head scans, and extracting modes of shape and texture variation using PCA. We combined this with an anatomy-based eyeball model that can be posed separately to simulate changes in eye gaze.

Analysis-by-synthesis for gaze estimation Our second contribution is a novel method for gaze estimation: fitting our 3DMM to an input image using

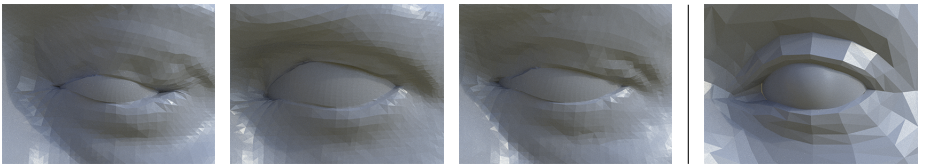

Fig. 2: A comparison between the Basel Face Model (BFM, left) [11], and our own (right). Note the BFM's lack of caruncle and unrealistic eyeball proxy geometry. Our model has well-defined correspondences for these difficult regions.

analysis-by-synthesis (see Figure 1 right). We solve for shape, texture, pose, and illumination simultaneously, so our fitted model parameters provide us with a robust estimate of where someone is looking in a 3D scene. Previous approaches for remote RGB gaze estimation can be categorized as either appearance-based, feature-based, or model-based [3]. Our method is first to combine the benefits of all three: 1) We minimize the appearance difference between synthesized and observed images using a dense image-error term. 2) We use sparse facial features localized with a face tracker [15] for initialization and regularization. 3) We use our morphable model to capture variation between people and eye motion itself. We iteratively fit our model using gradient descent with numerical derivatives efficiently calculated with a tailored GPU rasterizer.

2 Related work

2.1 3D morphable models

A 3D morphable model is a statistically-derived generative model, parameterized by shape and texture coefficients. They are closely related to their 2D analogue, active appearance models [16]. 3DMMs have been successfully applied to various face-related computer vision problems ranging from reconstruction [8,10] to recognition [11,12], and have also been extended to other body parts, such as the hand [17] as well as the entire body itself [18,19].

Blanz & Vetter built the first 3DMM from a set of 200 laser scans of faces with neutral expression [8]. They first computed a dense correspondences between the scans, then used PCA to extract modes of variation. Subsequent work with 3DMMs has followed the same approach, building similar models with higher quality scans [11], or more training samples [12,20]. However, despite advances in scanning technology, the eye remains problematic for 3D reconstruction, leading to poor correspondences and loss of quality in the 3DMM (see Figure 2).

3DMMs represent a face with neutral expression, so they are often combined with a model of facial motion. Vlasic et al. used a multi-linear model to separately encode identity and expression, and demonstrated its use in facial transfer [21]. More recent works have instead used blend shapes – an animation technique that stores a different version of a mesh for each expression, and interpolates between them [14]. However, while blend shapes work well for skin, they cannot

represent the independent motion of the eyeball. For these reasons, previous work either replaced the scanned eyeball with a proxy mesh [11] or completely removed the eye from the 3DMM mesh [13,22]. Bérard et al. recently presented a 3D morphable eyeball model [23] built from a database of eyeball scans [24], showing impressive results for high-quality semi-automatic eyeball reconstruction. Our work uses a simpler model that is sufficient for low-quality input data, and our fitting procedure is fully automatic.

2.2 Remote gaze estimation

Gaze estimation is a well established topic in computer vision (see [3,25] for reviews). Methods can be categorized as 1) *appearance-based* – map directly from image pixels to a gaze direction [5,26,27], 2) *feature-based* – localize facial feature points (e.g. pupil centre, eye corner) and map these to gaze [28,29], or 3) *model-based* – estimate gaze using a geometric model of the eye [30,31,32]. Some systems combine these techniques, e.g. using facial features for image alignment [26,33], mapping appearance to a 2D generative model [34], or combining head pose with image pixels in a multi-modal neural network [5]. To the best of our knowledge, no work so far has combined appearance, facial features, and a generative model into a single method, solving for shape, texture, eyeball pose, and illumination simultaneously.

The current outstanding challenge for remote RGB gaze estimation is achieving person- and device- independence under unconstrained conditions [5]. The state-of-the-art methods for this are appearance-based, attempting to learn invariances from large amounts of training data. However, such systems are still limited by their training data with respect to appearance, gaze, and head pose variation [5,27]. To address this, recent work used graphics to synthesize large amounts of training images. These learning-by-synthesis methods cover a larger range of head pose, gaze, appearance, and illumination variation without additional costs for data collection or ground truth annotation. Specifically, Wood et al. rendered 10K images and used them to pre-train a multi-modal CNN, significantly improving upon state-of-the-art gaze estimation accuracy [7]. They later rendered 1M images with improved appearance variation for training a k-Nearest-Neighbour classifier, again improving over state-of-the-art CNN results [6].

While previous work used 3D models to synthesise training data [6], ours is first to use analysis-by-synthesis – a technique where synthesis is used for gaze estimation itself. This approach is not constrained by a limited variation in training images but instead can, in theory, generalise to arbitrary settings. Additionally, while previous work strove for realism [7], our forward synthesis method focuses on speed in order to make analysis-by-synthesis tractable.

3 Overview

At the heart of our generic gaze estimator are two core contributions. In section 4 we present our first contribution: a novel multi-part eye region 3DMM. We

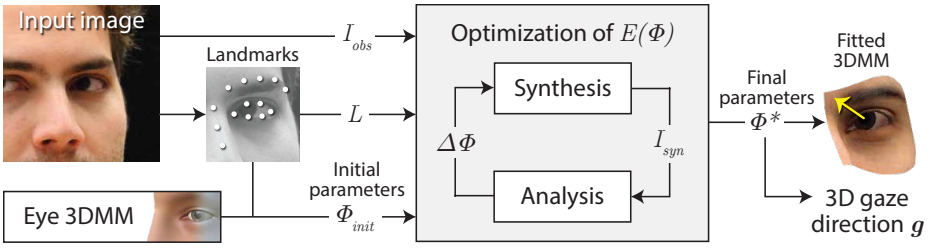

Fig. 3: An overview our fitting process: We localize landmarks L in an image, and use them to initialze our 3DMM. We then use analysis-by-synthesis to render an I_{syn} that best matches I_{obs}. We finally extract gaze $\boldsymbol{g}$ from fitted paramters Φ^*.

constructed this from 22 high-resolution face scans acquired from an online store[4], combined with an anatomy-based eyeball model. Our model is described by a set of parameters Φ that cover both geometric (shape, texture, and pose) and photometric (illumination and camera projection) variation.

In section 5 we present our second contribution: analysis-by-synthesis for gaze estimation (see Figure 3). The core idea is to fit our 3DMM to an image using *analysis-by-synthesis* – given an observed image I_{obs}, we wish to produce a synthesized image I_{syn} that matches it. We then estimate gaze from the fitted eyeball pose parameters. Key in this process is our objective function $E(\Phi)$, which considers both a local dense measure of appearance similarity, as well as a holistic sparse measure of facial feature-point similarity (see Equation 10).

4 3D eye region model

Our goal is to use a 3D eye region model to synthesize an image which matches an input RGB eye image. To render synthetic views, we used a multi-part model consisting of the facial eye region and the eyeball. These were posed in a scene, illuminated, and then rendered using a model of camera projection. Our total set of model and scene parameters Φ are:

$$\Phi = \{\beta, \tau, \theta, \iota, \kappa\}\,, \tag{1}$$

where β are the shape parameters, τ the texture parameters, θ the pose parameters, ι the illumination parameters, and κ the camera parameters. In this section we describe each part of our model, and the parameters that affect it.

Morphable facial eye region model – β, τ The first part of our model is a 3DMM of the eye region, and serves as a prior for facial appearance. While previous work used a generative shape model of the eye region [6], ours captures both shape and texture variation, allowing .

We started by acquiring 22 high-quality head scans as source data. The first stage of constructing a morphable model is bringing scan data into correspondence,

[4] Ten24 3D Scan Store – `http://3dscanstore.com/`

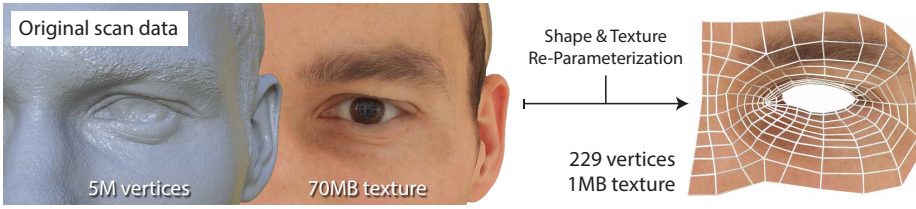

Fig. 4: We re-parameterize high-resolution 3D head scan data (left) into a more efficient lower resolution form (right). We use a carefully designed generic eye region topology [6] for consistent correspondences and realistic animation.

so a point in one face mesh is semantically equivalent to a point in another. While previous work computed a dense point-to-point correspondence from original scan data [8,11], we compute sparse correspondences that describe 3D shape more efficiently. We manually re-parameterised each original high-resolution scamn into a low resolution topology containing the eye region only (see Figure 4). This topology does not include the eyeball, as we wish to pose that separately to simulate its independent movement. Additionally, we maintain correspondences for detailed parts, e.g. the interior eyelid margins, which are poorly defined for previous models [11]. We *uv*-unwrap the mesh and represent color as a texture map, coupling our low-resolution mesh with a high-resolution texture.

Following this registration, the facial eye regions are represented as a combination of 3D shape $\boldsymbol{s}$ (n vertices) and 2D texture $\boldsymbol{t}$ (m texels), encoded as $3n$ and $3m$ dimensional vectors respectively,

$$\boldsymbol{s} = [x_1, y_1, z_1, x_2, ...y_n, z_n]^T \in \mathbb{R}^{3n} \tag{2}$$

$$\boldsymbol{t} = [r_1, g_1, b_1, r_2, ...g_m, b_m]^T \in \mathbb{R}^{3m} \tag{3}$$

where x_i, y_i, z_i is the 3D position of the ith vertex, and r_j, b_j, g_j is the color of the jth texel. We then performed *Principal Component Analysis* (PCA) on our set of c ordered scans to extract orthogonal shape and texture basis functions: $\boldsymbol{U} \in \mathbb{R}^{3n \times c}$ and $\boldsymbol{V} \in \mathbb{R}^{3m \times c}$. For each of the $2m$ shape and texture basis functions, we fit a Gaussian distribution to the original data. Using this we can construct linear models that describe variation in both shape $\mathcal{M}_s$ and texture $\mathcal{M}_t$,

$$\mathcal{M}_s = (\boldsymbol{\mu}_s, \boldsymbol{\sigma}_s, \boldsymbol{U}) \qquad \mathcal{M}_t = (\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t, \boldsymbol{V}) \tag{4}$$

where $\boldsymbol{\mu}_s \in \mathbb{R}^{3n}$ and $\boldsymbol{\mu}_t \in \mathbb{R}^{3m}$ are the average 3D shape and 2D texture, and $\boldsymbol{\sigma}_s = [\sigma_{s1} ... \sigma_{sc}]$ and $\boldsymbol{\sigma}_t = [\sigma_{t1} ... \sigma_{tc}]$ describe the Gaussian distributions of each shape and texture basis function. Figure 5 shows the mean shape and texture, along with the four most important modes of variation. Facial eye region shapes $\boldsymbol{s}$ and textures $\boldsymbol{t}$ can then be generated from shape ($\beta_{face} \subset \beta$) and texture coefficients ($\tau_{face} \subset \tau$) as follows:

$$\boldsymbol{s}(\beta_{face}) = \boldsymbol{\mu}_s + \boldsymbol{U}\, \mathrm{diag}(\boldsymbol{\sigma}_s)\, \beta_{face} \tag{5}$$

$$\boldsymbol{t}(\tau_{face}) = \boldsymbol{\mu}_t + \boldsymbol{V}\, \mathrm{diag}(\boldsymbol{\sigma}_t)\, \tau_{face} \tag{6}$$

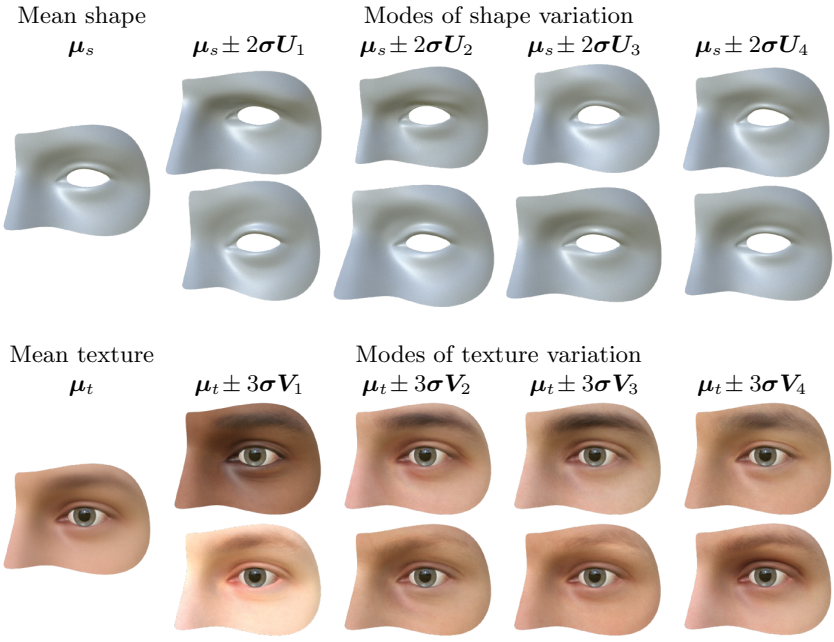

Fig. 5: The mean shape $\boldsymbol{\mu}_s$ and texture $\boldsymbol{\mu}_t$ along with the first four modes of variation. The first shape mode $\boldsymbol{U}_1$ varies between hooded and protruding eyes, and the first texture mode $\boldsymbol{V}_1$ varies between dark and light skin.

From our set of $c = 22$ scans, 90% of shape and texture variation can be encoded in 8 shape and 7 texture coefficients. This reduction in dimensionality is important for fitting our model efficiently. Additionally, as eyelashes can provide a visual cue to gaze direction, we model them model them using a semi-transparent mesh controlled by a simple hair simulation [6].

Parametric eyeball model – β, τ The second part of our multi-part model is the eyeball. Accurately recovering eyeball shape is difficult due to its complex structure [24], so instead we created a mesh using standard anatomical measurements [6] (see Figure 6). Eyeballs vary in shape and texture between different people. We model changes in iris size geometrically, by scaling vertices on the iris boundary about the 3D iris centre as specified by iris diameter β_{iris}. We used a collection of aligned high-resolution iris photos to build a generative model $\mathcal{M}_{iris}$ of iris texture using PCA,

$$\mathcal{M}_{iris} = (\boldsymbol{\mu}_{iris}, \boldsymbol{\sigma}_{iris}, \boldsymbol{W}) \tag{7}$$

This can be used to generate new iris textures $\boldsymbol{t}_{iris}$. As the "white" of the eye is not purely white, we model variations in sclera color by multiplying the eyeball texture with a tint color $\tau_{tint} \in \mathbb{R}^3$. In reality, the eyeball has a complex layered structure with a transparent cornea covering the iris. We avoid explicitly modelling this by computing refraction effects in texture-space [6,35].

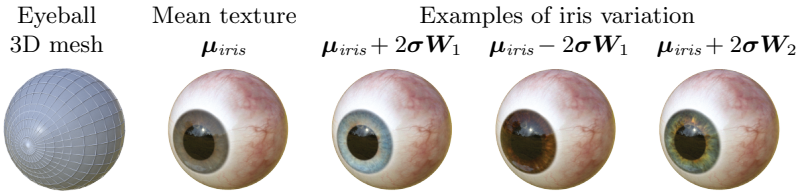

Fig. 6: Our eyeball mesh, mean iris texture $\boldsymbol{\mu}_{iris}$, and some examples of iris texture variation captured by our linear model $\mathcal{M}_{iris}$.

Posing our multi-part model – θ Global and local pose information is encoded by θ. Our model's parts are defined in a local coordinate system with origin at the eyeball centre, so we use model-to-world transforms $\boldsymbol{M}_{face}$ and $\boldsymbol{M}_{eye}$ to position them in a scene. The facial eye region part has degrees of freedom in translation and rotation. These are encoded as 4×4 homogenous transformation matrices $\boldsymbol{T}$ and $\boldsymbol{R}$, so model-to-world transform $\boldsymbol{M}_{face} = \boldsymbol{T}\boldsymbol{R}$. The eyball's position is anchored to the face model, but it can rotate separately through local pitch and yaw transforms $\boldsymbol{R}_x(\theta_p)$ and $\boldsymbol{R}_y(\theta_y)$, giving $\boldsymbol{M}_{eye} = \boldsymbol{T}\boldsymbol{R}_x\boldsymbol{R}_y$.

When the eye looks up or down, the eyelid follows it. Eyelid motion is modelled using procedural animation [6] – each eyelid vertex is rotated about the inter-eye-corner axis, with rotational amounts chosen to match measurements from an anatomical study [36]. As our multi-part model contains disjoint parts, we also "shrinkrwap" the eyelid skin to the eyeball, projecting eyelid vertices onto the eyeball mesh to avoid gaps and clipping issues.

Scene illumination – ι As we focus on a small region of the face, we assume a simple illumination model where lighting is distant and surface materials are purely Lambertian. Our illumination model consists of an ambient light with color $\boldsymbol{l}_{amb} \in \mathbb{R}^3$, and a directional light with color $\boldsymbol{l}_{dir} \in \mathbb{R}^3$ and 3D direction vector $\boldsymbol{L}$. We do not consider specular effects, global illumination, or self-shadowing, so illumination depends only on surface normal and albedo. Radiant illumination $\mathcal{L}$ at a point on the surface with normal $\boldsymbol{N}$ and albedo $\boldsymbol{c}$ is calculated as:

$$\mathcal{L}(\boldsymbol{n}, \boldsymbol{c}) = \boldsymbol{c}\,\boldsymbol{l}_{amb} + \boldsymbol{c}\,\boldsymbol{l}_{dir}\,(\boldsymbol{N} \cdot \boldsymbol{L}) \tag{8}$$

While this model is simple, we found it to be sufficient. If we considered a larger facial region, or fit models to both eyes at once, we would explore more advanced material or illumination models, as seen in previous work [13].

Camera projection – κ For a complete model of image formation, we also consider camera projection. We fix our axis-aligned camera at world origin, allowing us to set our world-to-view transform as the identity $\boldsymbol{I}_4$. We assume knowledge of intrinsic camera calibration parameters κ, and use these to build a full projection transform $\boldsymbol{P}$. A local point in our model can then be transformed into image space using the model-view-projection transform $\boldsymbol{P}\boldsymbol{M}_{\{face|eye\}}$.

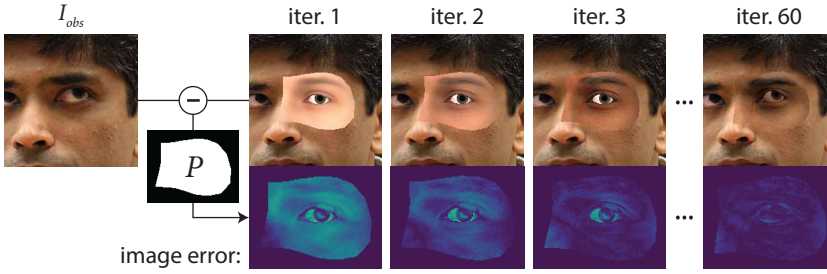

Fig. 7: We measure dense image-similarity as the mean absolute error between I_{obs} and I_{syn}, over a mask of rendered foreground pixels P (white). We ignore error for background pixels (black).

5 Analysis-by-synthesis for gaze estimation

Given an observed image I_{obs}, we wish to produce a synthesized image $I_{syn}\,(\Phi^*)$ that best matches it. 3D gaze direction $\boldsymbol{g}$ can then be extracted from eyeball pose parameters. We search for optimal model parameters Φ^* using *analysis-by-synthesis*. To do this, we iteratively render a synthetic image $I_{syn}\,(\Phi)$, compare it to I_{obs} using our energy function, and update Φ accordingly. We cast this as an unconstrained energy minimization problem for unknown Φ.

$$\Phi^* = \underset{\Phi}{\operatorname{argmin}}\; E(\Phi) \tag{9}$$

5.1 Objective function

Our energy is formulated as a combination of a dense *image similarity metric* E_{image} that minimizes difference in image appearance, and a sparse *landmark similarity metric* E_{ldmks} that regularizes our model against reliable facial feature points, and weight λ controlling their relative importance.

$$E(\Phi) = E_{image}(\Phi) + \lambda \cdot E_{ldmks}(\Phi,\, L) \tag{10}$$

Image similarity metric Our primary goal is to minimise the difference between I_{syn} and I_{obs}. This can be seen as an ideal energy function: if $I_{syn} = I_{obs}$, our model must have perfectly fit the data, so virtual and real eyeballs should be aligned. We approach this by including a dense photo-consistency term E_{image} in our energy function. However, as the 3DMM in I_{syn} does not cover the entire of I_{obs}, we split our image into two regions: a set of rendered foreground pixels P that we compute error over, and a set of background pixels that we ignore (see Figure 7). Image similarity is then computed as the mean absolute difference between I_{syn} and I_{obs} for foreground pixels $p \in P$.

$$E_{image}(\Phi) = \frac{1}{|P|} \sum_{p \in P} |I_{syn}(\Phi, p) - I_{obs}(p)| \tag{11}$$

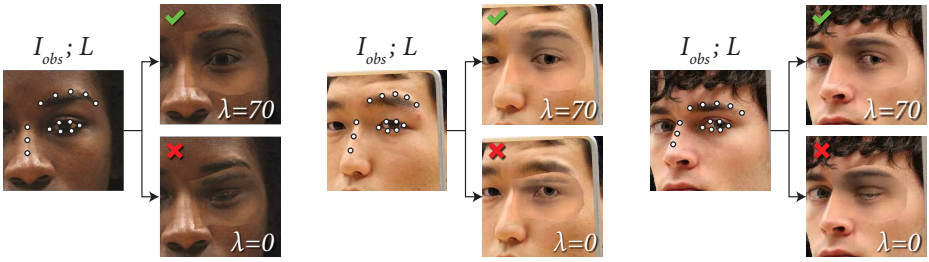

Fig. 8: I_{obs} with landmarks L (white dots), and model fits with our landmark similarity term (top), and without (bottom). Note how it prevents erroneous drift in global pose, eye region shape, and local eyelid pose.

Landmark similarity metric The face contains important *landmark* feature points that can be localized reliably [13]. These can be used to efficiently consider the appearance of the whole face, as well as the local appearance of the eye region. We use a state-of-the-art face tracker [15] to localize 14 landmarks L around the eye region in image-space (see Figure 8). For each landmark $l \in L$ we compute a corresponding synthesized landmark l' using our 3DMM. The sparse landmark-similarity term is calculated as the distance between both sets of landmarks, normalized by the foreground area to avoid bias from image or eye region size. This acts as a regularizer to prevent our pose θ from drifting too far from a reliable estimate.

$$E_{ldmks}(\Phi,\, L) = \frac{1}{|L|} \sum_{i=0}^{|L|} \|l_i - l'_i\| \tag{12}$$

5.2 Optimization procedure

We fit our model to the subject's left eye. This is a challenging non-convex, high-dimensional optimization problem. To approach it we use gradient descent (GD) with an annealing step size. Calculating analytic derivatives for a scene as complex as our eye region is challenging due to occlusions. We therefore use numeric central derivatives ∇E to guide our optimization procedure:

$$\Phi_{i+1} = \Phi_i - \boldsymbol{t} \cdot r^i \, \nabla E(\Phi_i) \qquad \text{where} \tag{13}$$

$$\nabla E(\Phi_i) = \left(\frac{\partial E}{\phi_1} \dots \frac{\partial E}{\phi_{|\Phi|}} \right) \quad \text{and} \quad \frac{\partial E}{\phi_j} = \frac{E(\Phi_i + h_j) - E(\Phi_i - h_j)}{2h_j} \tag{14}$$

where $\boldsymbol{t} = [t_1 ... t_{|\Phi|}]$ are per-parameter step-sizes, $\boldsymbol{h} = [h_1 ... h_{|\Phi|}]$ are per-parameter numerical values, and r the annealing rate. $\boldsymbol{t}$ and $\boldsymbol{h}$ were calibrated through experimentation. We explored alternate optimization techniques including LBFGS [37], and rprop [38] and momentum variants of GD, but we found these to be less stable, perhaps due to our use of numerical rather than analytical derivatives. Computing our gradients is expensive, requiring rendering and differencing two

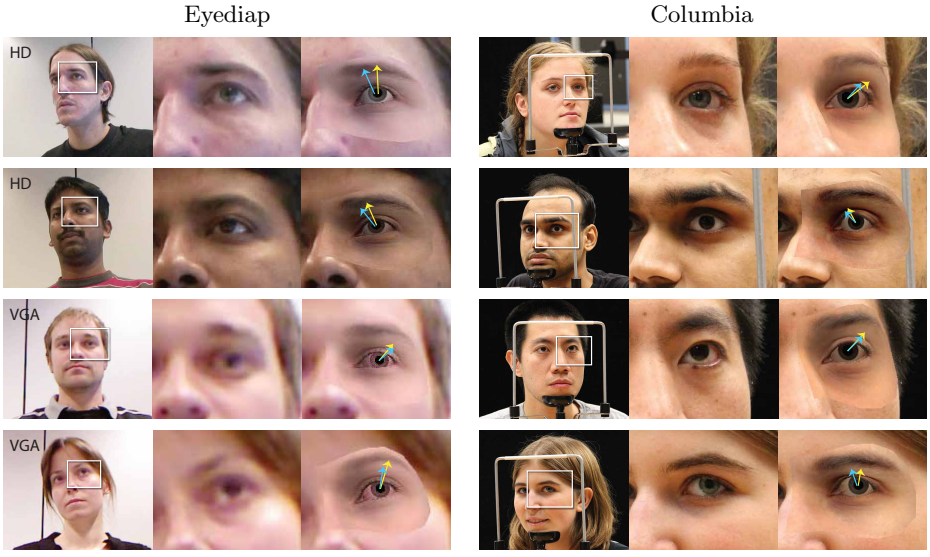

Fig. 9: Example model fits on gaze datasets Eyediap [39] (HD and VGA) and Columbia [40], showing estimated gaze (yellow) and labelled gaze (blue).

images per parameter. Their efficient computation is possible with our tailored GPU DirectX rasterizer that can render I_{syn} at over 5000fps.

Initialization As we perform local optimization, we require an initial model configuration to start from. We use 3D eye corner landmarks and head rotation from the face tracker [15] to initialize $\boldsymbol{T}$ and $\boldsymbol{R}$. We then use 2D iris landmarks and a single sphere eyeball model to initialize gaze [2]. β and τ are initialized to $\mathbf{0}$, and illumination $\boldsymbol{l}_{amb}$ and $\boldsymbol{l}_{dir}$ are set to $[0.8, 0.8, 0.8]$.

Runtime Figure 7 shows convergence for a typical input image, with I_{obs} size 800×533px, and I_{syn} size 125×87px. We converge after 60 iterations for 39 parameters, taking 3.69s on a typical PC (3.3Ghz CPU, GTX 660 GPU).

5.3 Extracting gaze direction

Our task is estimating 3D gaze direction $\boldsymbol{g}$ in camera-space. Once our fitting procedure has converged, $\boldsymbol{g}$ can be extracted by applying the eyeball model transform to a vector pointing along the optical axis in model-space: $\boldsymbol{g} = \boldsymbol{M}_{eye}\,[0, 0, -1]^T$.

6 Experiments

We evaluated our approach on two publicly available eye gaze datasets: Columbia [40] and Eyediap [39]. We chose these datasets as they show the full face, as required for our facial-landmark based initialization.

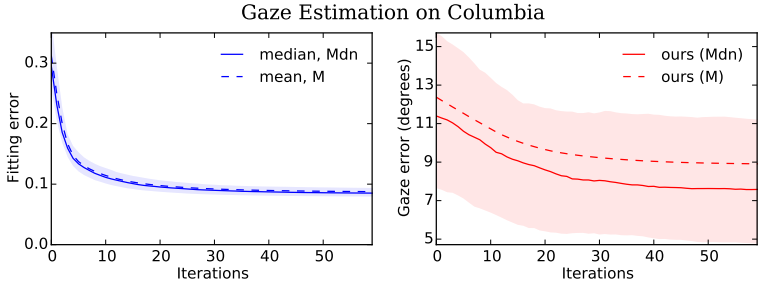

Fig. 10: Fitting error (left) and gaze estimation error (right). Note how gaze error improves from the initial estimate. Filled regions show inter-quartile range.

Columbia contains of images of 56 people looking at a target grid on the wall. The participants were constrained by a head-clamp device, and images were taken from five different head orientations (from $-30°$ to $30°$). Example fits can be seen in Figure 9 right. In our experiments we used a subset of 34 people (excluding those with eyeglasses) with 20 images per person, resulting in 680 images. As the images were taken by a high quality camera (5184×3456px), we downsampled them to 800×533px for faster processing.

Eyediap contains videos of 16 participants looking at two types of targets: *screen* targets on a monitor; and *floating* physical targets. Recordings were made with two cameras: a VGA camera (640×480px) below the screen, and a HD camera (1920×1080px) placed to the side. Example fits can be seen in Figure 9 left. Participants displayed both static and free head motion. We extracted images from the VGA videos for our experiment – 622 images with screen targets and 500 images with floating targets. In both cases we used a gradient descent step size of 0.0025 with an annealing rate of 0.95 that started after 10^{th} iteration.

6.1 Gaze estimation

In the first experiment we evaluated how well our method predicts gaze direction for Columbia. The results are shown in Figure 10, giving average gaze error of $\text{M} = 8.87°, \text{Mdn} = 7.54°$ after convergence. As we do not impose a prior on predicted gaze distribution, our system can produce outliers with extreme error, so we believe its performance is best represented by a median (Mdn) average. Note how the decrease in fitting error corresponds to a monotonic decrease in mean and median gaze errors. Furthermore, our approach outperformes the geometric approach used to initialize it [2], a recently proposed k-Nearest-Neighbour approach [6] ($\text{M} = 19.9°, \text{Mdn} = 19.5°$) and a naïve model that always predicts forwards gaze ($\text{M} = 12.00°, \text{Mdn} = 11.17°$).

The results for Eyediap VGA images can be seen in Figure 11. As before the decrease in pixel error corresponds in the decrease in gaze errors. Furthermore, our final gaze estimation error on the Eyediap *screen* condition ($\text{M} = 9.44°, \text{Mdn} = 8.63°$) outperfoms that reported in literature previously ($p < .0001$, independent

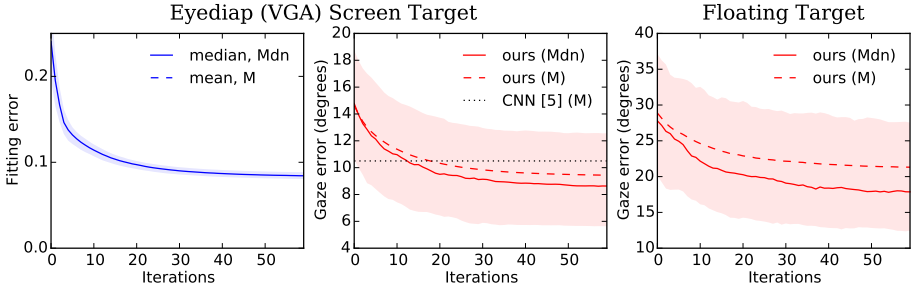

Fig. 11: Fitting (blue) and gaze estimation (red) error on Eyediap (VGA). We outperform a state-of-the-art CNN [5]. Additionally, the CNN was not able to generalize to the floating target condition, while ours can.

	ours	CNN	RF	kNN	ALR	SVR	synth.
Gaze error (M°)	**9.44**	10.5	12.0	12.2	12.7	15.1	19.9

Table 1: We outperform state-of-the-art cross-dataset methods trained on UT [27] and synthetic data [6]: CNN [5], Random Forests (RF) [27], kNN [5], Adaptive Linear Regression (ALR) [33], and Support Vector Regression (SVR) [26].

t-test) – 10.5° using a Convolutional Neural Network [5]. See Table 1 for other comparisons. We also outperform the initialization model, a kNN model (M = 21.49°, Mdn = 20.93°), and a naïve model (M = 12.62°, Mdn = 12.79°). The results for floating targets are less accurate but still improve upon our initialisation baseline. Zhang et al. [5] did not evaluate on floating targets due to head pose variations not present in their training set. Despite a drop in accuracy, our method can still generalize to this difficult scenario and outperforms a kNN model (M = 30.85°, Mdn = 28.92°), and a naïve model (M = 31.4°, Mdn = 31.37°).

We performed a similar experiment for Eyediap HD images that exhibit head pose, achieving a gaze error of M = 11.0°, Mdn = 10.4° for screen targets and M = 22.2°, Mdn = 19.0° for floating targets. Despite extreme head pose and gaze range, we still perform comparably with the state-of-the-art and outperform a kNN model (M = 29.39°, Mdn = 28.62° for screen, and M = 34.6°, Mdn = 33.19° for floating target), and a naïve model (M = 22.67°, Mdn = 22.06° for screen, and M = 35.08°, Mdn = 34.35° for floating target).

6.2 Morphable model evaluation

In addition to evaluating our system's gaze estimation capabilities, we performed experiments to measure the expressive power of our morphable model and the effect of including E_{ldmks} in our objective function.

First, we assessed the importance of our facial point similarity weight (λ) to gaze estimation accuracy on the Columbia dataset. We used the same fitting

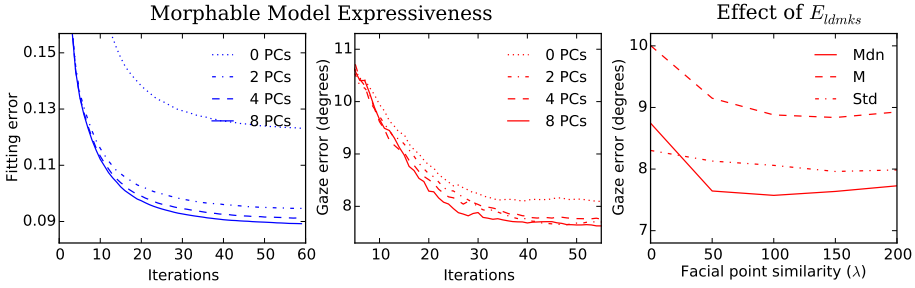

Fig. 12: As we include more shape and texture and shape principal components (PCs) in the facial morphable model, both fitting and gaze error decrease. Also note the effect of our landmark regularization term λ which decreases the error (and its standard deviation) by not allowing the fit to drift.

strategy, but varied λ. Results can be seen in Figure 12 (right). It is clear that λ has a positive impact on gaze estimation accuracy, by not allowing fits to drift too far from the reliable estimates and by reducing the variance of the error.

Second, we wanted to see if modelling more degrees of shape and appearance variation led to better image fitting and gaze estimation. We therefore varied the number of shape (β) and texture (τ) principal components (PCs) that our model was allowed to use during fitting on Columbia. We varied both the texture and shape PCs together, using the same number for both. As seen in Figure 12 (left), more PCs lead to better image fitting error, as I_{syn} matches I_{obs} better when allowed more variation. A similar downward trend can be seen for gaze error, suggesting better modelling of nearby facial shape and texture is important for correctly aligning the eyeball model, and thus determining gaze direction.

7 Conclusion

We presented the first multi-part 3D morphable model of the eye region. It includes a separate eyeball model, allowing us to capture gaze – a facial expression not captured by previous systems [13,14]. We then presented a novel approach for gaze estimation: fitting our model to an image with analysis-by-synthesis, and extracting the gaze direction from fitted parameters. Our method is the first to jointly optimize a dense image metric, a sparse feature metric, and a generative 3D model together for gaze estimation. It generalizes to different quality images and wide gaze ranges, and out-performs a state-of-the-art CNN method [5].

Limitations still remain. While other gaze estimation systems can operate in real time [2,5], ours takes several seconds per image. However, previous analysis-by-synthesis systems have been made real time through careful engineering [41]; we believe this is possible for our method too. Our method can also become trapped in local minima (see Figure 8). To avoid this and improve robustness, we plan to fit both eyes simultaneously in future work.

References

1. Kleinke, C.L.: Gaze and eye contact: a research review. Psychological bulletin **100**(1) (1986) 78–100
2. Baltrušaitis, T., Robinson, P., Morency, L.P.: Openface: an open source facial behavior analysis toolkit. In: IEEE WACV. (2016)
3. Hansen, D.W., Ji, Q.: In the eye of the beholder: A survey of models for eyes and gaze. Pattern Analysis and Machine Intelligence, IEEE Transactions on **32**(3) (2010) 478–500
4. Majaranta, P., Bulling, A.: Eye tracking and eye-based human–computer interaction. In: Advances in Physiological Computing. Springer (2014) 39–65
5. Zhang, X., Sugano, Y., Fritz, M., Bulling, A.: Appearance-based gaze estimation in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 4511–4520
6. Wood, E., Baltrušaitis, T., Morency, L.P., Robinson, P., Bulling, A.: Learning an appearance-based gaze estimator from one million synthesised images. In: Proc. ETRA. (2016)
7. Wood, E., Baltrušaitis, T., Zhang, X., Sugano, Y., Robinson, P., Bulling, A.: Rendering of eyes for eye-shape registration and gaze estimation. In: ICCV. (2015)
8. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: Conference on Computer graphics and interactive techniques, ACM (1999)
9. Romdhani, S., Vetter, T.: Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In: Proc. CVPR, 2005. Volume 2., IEEE (2005) 986–993
10. Aldrian, O., Smith, W.A.: Inverse rendering of faces with a 3d morphable model. Pattern Analysis and Machine Intelligence, IEEE Transactions on **35**(5) (2013) 1080–1093
11. Paysan, P., Knothe, R., Amberg, B., Romdhani, S., Vetter, T.: A 3d face model for pose and illumination invariant face recognition. Proc. AVSS (2009)
12. Yi, D., Lei, Z., Li, S.: Towards pose robust face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 3539–3545
13. Thies, J., Zollhöfer, M., Nießner, M., Valgaerts, L., Stamminger, M., Theobalt, C.: Real-time expression transfer for facial reenactment. ACM TOG (2015)
14. Cao, C., Weng, Y., Lin, S., Zhou, K.: 3d shape regression for real-time facial animation. ACM TOG (2013)
15. Baltrušaitis, T., Morency, L.P., Robinson, P.: Constrained local neural fields for robust facial landmark detection in the wild. In: IEEE ICCVW. (2013)
16. Cootes, T.F., Edwards, G.J., Taylor, C.J., et al.: Active appearance models. IEEE Transactions on pattern analysis and machine intelligence **23**(6) (2001) 681–685
17. Khamis, S., Taylor, J., Shotton, J., Keskin, C., Izadi, S., Fitzgibbon, A.: Learning an efficient model of hand shape variation from depth images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 2540–2548
18. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: Scape: shape completion and animation of people. In: ACM Transactions on Graphics (TOG). Volume 24., ACM (2005) 408–416
19. Hasler, N., Stoll, C., Sunkel, M., Rosenhahn, B., Seidel, H.P.: A statistical model of human pose and body shape. In: Computer Graphics Forum. Volume 28., Wiley Online Library (2009) 337–346
20. Booth, J., Roussos, A., Zafeiriou, S., Ponniah, A., Dunaway, D.: A 3d morphable model learnt from 10,000 faces. Proc. CVPR, 2016 (2016)

21. Vlasic, D., Brand, M., Pfister, H., Popović, J.: Face transfer with multilinear models. In: ACM Transactions on Graphics (TOG). Volume 24., ACM (2005) 426–433
22. Cao, C., Weng, Y., Zhou, S., Tong, Y., Zhou, K.: Facewarehouse: A 3d facial expression database for visual computing. TVGC **20**(3) (2014)
23. Bérard, P., Bradley, D., Gross, M., Beeler, T.: Lightweight eye capture using a parametric model. ACM Transactions on Graphics (TOG) **35**(4) (2016) 117
24. Bérard, P., Bradley, D., Nitti, M., Beeler, T., Gross, M.: Highquality capture of eyes. ACM Transactions on Graphics (2014)
25. Ferhat, O., Vilarino, F.: Low cost eye tracking: The current panorama. Journal of Computational Intelligence and Neuroscience **22**(23) 24
26. Schneider, T., Schauerte, B., Stiefelhagen, R.: Manifold alignment for person independent appearance-based gaze estimation. In: 2014 22nd International Conference on Pattern Recognition (ICPR), IEEE (2014) 1167–1172
27. Sugano, Y., Matsushita, Y., Sato, Y.: Learning-by-synthesis for appearance-based 3d gaze estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1821–1828
28. Sesma, L., Villanueva, A., Cabeza, R.: Evaluation of pupil center-eye corner vector for gaze estimation using a web cam. In: Proceedings of the symposium on eye tracking research and applications, ACM (2012) 217–220
29. Torricelli, D., Conforto, S., Schmid, M., DAlessio, T.: A neural-based remote eye gaze tracker under natural head motion. Computer methods and programs in biomedicine **92**(1) (2008) 66–78
30. Wood, E., Bulling, A.: Eyetab: Model-based gaze estimation on unmodified tablet computers. In: Proceedings of the Symposium on Eye Tracking Research and Applications, ACM (2014) 207–210
31. Wang, J., Sung, E., Venkateswarlu, R.: Eye gaze estimation from a single image of one eye. In: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, IEEE (2003) 136–143
32. Wu, H., Chen, Q., Wada, T.: Conic-based algorithm for visual line estimation from one image. In: Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on, IEEE (2004) 260–265
33. Lu, F., Sugano, Y., Okabe, T., Sato, Y.: Adaptive linear regression for appearance-based gaze estimation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **36**(10) (2014) 2033–2046
34. Mora, K., Odobez, J.M.: Geometric generative gaze estimation (g3e) for remote rgb-d cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1773–1780
35. Jimenez, J., Danvoye, E., von der Pahlen, J.: Photorealistic eyes rendering. In: SIGGRAPH Talks, Advances in Real-Time Rendering, ACM (2012)
36. Malbouisson, J.M., e Cruz, A.A.V., Messias, A., Leite, L.V., Rios, G.D.: Upper and lower eyelid saccades describe a harmonic oscillator function. Investigative ophthalmology & visual science **46**(3) (2005) 857–862
37. Liu, D.C., Nocedal, J.: On the limited memory bfgs method for large scale optimization. Mathematical programming **45**(1-3) (1989) 503–528
38. Riedmiller, M., Braun, H.: Rprop-a fast adaptive learning algorithm. In: Proc. of ISCIS VII), Universitat, Citeseer (1992)
39. Funes Mora, K.A., Monay, F., Odobez, J.M.: EYEDIAP: A Database for the Development and Evaluation of Gaze Estimation Algorithms from RGB and RGB-D Cameras. Proc. ETRA (2014)

40. Smith, B., Yin, Q., Feiner, S., Nayar, S.: Gaze Locking: Passive Eye Contact Detection for HumanObject Interaction. In: ACM Symposium on User Interface Software and Technology (UIST). (Oct 2013) 271–280
41. Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2face: Real-time face capture and reenactment of rgb videos. Proc. Computer Vision and Pattern Recognition (CVPR), IEEE **1** (2016)