2015 IEEE International Conference on Computer Vision

Rendering of Eyes for Eye-Shape Registration and Gaze Estimation

Erroll Wood[1], Tadas Baltrušaitis[1], Xucong Zhang[2], Yusuke Sugano[2], Peter Robinson[1], Andreas Bulling[2]

University of Cambridge, United Kingdom {eww23,tb346,pr10}@cam.ac.uk
Max Planck Institute for Informatics, Germany {xczhang,sugano,bulling}@mpi-inf.mpg.de

Abstract

Images of the eye are key in several computer vision problems, such as shape registration and gaze estimation. Recent large-scale supervised methods for these problems require time-consuming data collection and manual annotation, which can be unreliable. We propose synthesizing perfectly labelled photo-realistic training data in a fraction of the time. We used computer graphics techniques to build a collection of dynamic eye-region models from head scan geometry. These were randomly posed to synthesize close-up eye images for a wide range of head poses, gaze directions, and illumination conditions. We used our model's controllability to verify the importance of realistic illumination and shape variations in eye-region training data. Finally, we demonstrate the benefits of our synthesized training data (SynthesEyes) by out-performing state-of-the-art methods for eye-shape registration as well as cross-dataset appearance-based gaze estimation in the wild.

1. Introduction

The eyes and their movements convey our attention and play a role in communicating social and emotional information [1]. Therefore they are important for a range of applications including gaze-based human-computer interaction [2], visual behavior monitoring [3], and – more recently – collaborative human-computer vision systems [4, 5]. Typical computer vision tasks involving the eye include *gaze estimation*: determining where someone is looking, and *eye-shape registration*: detecting anatomical landmarks of the eye, often as part of the face (e.g. eyelids).

Machine learning methods that leverage large amounts of training data currently perform best for many problems in computer vision, such as object detection [6], scene recognition [7], or gaze estimation [8]. However, capturing data for supervised learning can be time-consuming and require accurate ground truth annotation. This annotation process can be expensive and tedious, and there is no guarantee that human-provided labels will be correct. Ground truth annotation is particularly challenging and error-prone for learning tasks that require accurate labels, such as tracking facial landmarks for expression analysis, and gaze estimation.

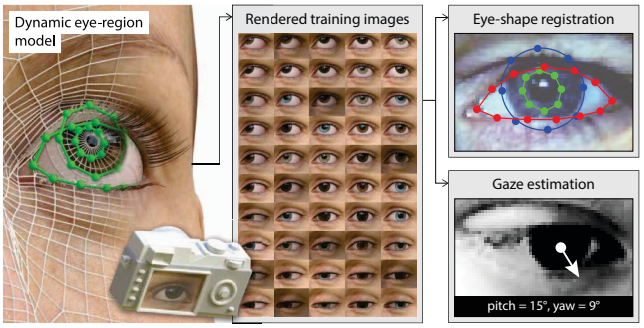


Figure 1: We render a large number of photorealistic images of eyes using a dynamic eye region model. These are used as training data for eye-shape registration and appearance-based gaze estimation.

To address these problems, researchers have employed *learning-by-synthesis* techniques to generate large amounts training data with computer graphics. The advantages of this approach are that both data collection and annotation require little human labour and image synthesis can be geared to specific application scenarios. The eye-region is particularly difficult to model accurately given the dynamic shape changes it undergoes with facial motion and eyeball rotation, and the complex material structure of the eyeball itself. For this reason, recent work on learning-by-synthesis for gaze estimation employed only fundamental computer graphics techniques – rendering low-resolution meshes without modeling illumination changes or accounting for the varying material properties of the face [9]. In addition, these models are not fully controllable and the synthesized datasets contain only gaze labels, limiting their usefulness for other computer vision problems, such as facial landmark registration.

We present a novel method for rendering realistic eye-region images at a large scale using a collection of dynamic

1550-5499/15 $31.00 © 2015 IEEE
DOI 10.1109/ICCV.2015.428

Figure 2: An overview of our model preparation process: Dense 3D head scans (1.4 million polygons) (a) are first retopologised into an optimal form for animation (9,005 polygons) (b). High resolution skin surface details are restored by displacement maps (c), and 3D iris and eyelid landmarks are annotated manually (d). A sample rendering is shown (e).

and controllable eye-region models. In contrast to previous work, we provide a comprehensive and detailed description of the model preparation process and rendering pipeline (see Figure 2 for an overview of the model preparation process and Figure 4 for the eye model used). We then present and evaluate two separate systems trained on the resulting data (*SynthesEyes*): an eye-region specific deformable model and an appearance-based gaze estimator. The controllability of our model allows us to quickly generate high-quality training data for these two disparate tasks. Please note that our model is not only limited to these scenarios but can potentially be used for other tasks that require realistic images of eyes, e.g. gaze correction or evaluation of iris-biometrics or geometry-based gaze estimation [10].

The specific contributions of this work are threefold. We first describe in detail our novel but straight-forward techniques for generating large amounts of synthesized training data, including wide degrees of realistic appearance variation using image-based-lighting. We then demonstrate the usefulness of SynthesEyes by out-performing state-of-the-art methods for eye-shape registration as well as challenging cross-dataset appearance-based gaze estimation in the wild. Finally, to ensure reproducibility and stimulate research in this area, we will make the eyeball model and generated training data publicly available at time of publication.

2. Related Work

Our work is related to previous work on 1) learning using synthetic data and 2) computational modeling of the eyes.

2.1. Learning Using Synthetic Data

Despite their success, the performance of learning-based approaches critically depends on how well the test data distribution is covered by the training set. Since recording training data that covers the full distribution is challenging, synthesized training data has been used instead. Previous work demonstrates such data to be beneficial for tasks such as body pose estimation [11, 12], object detection/recognition [13, 14, 15, 16], and facial landmark localization [17, 18]. Since faces exhibit large color and texture variability, some approaches side-stepped this by relying on depth images [17, 19], and synthesizing depth images of the head using existing datasets or a deformable head-shape model. Recent work has also synthesized combined color and geometry data by sampling labelled 3D-videos for training a dense 3D facial landmark detector [18].

As discussed by Kaneva et al. [20], one of the most important factors is the realism of synthesized training images. If the object of interest is highly complex, like the human eye, it is not clear whether we can rely on overly-simplistic object models. Zhang et al. [8] showed that gaze estimation accuracy significantly drops if the test data is from a different environment. Similarly to facial expression recognition [21], illumination effects are a critical factor. In contrast, our model allows synthesizing realistic lighting effects – an important degree of variation for performance improvements in eye-shape registration and gaze estimation.

Most similar to this work, Sugano et al. [9] used 3D reconstructions of eye regions to synthesize multi-view training data for appearance-based gaze estimation. One limitation of their work is that they do not provide a parametric model. Their data is a set of rigid and low-resolution 3D models of eye regions with ground-truth gaze directions, and hence cannot be easily applied to different tasks. Since our model instead is realistic and fully controllable, it can be used to synthesize close-up eye images with ground-truth eye landmark positions. This enables us to address eye shape registration via learning-by-synthesis for the first time.

2.2. Computational Modeling of the Eyes

The eyeballs are complex organs comprised of multiple layers of tissue, each with different reflectance properties and levels of transparency. Fortunately, given that realistic eyes are important for many fields, there is already a large body of previous work on modeling and rendering eyes (see Ruhland et al. [22] for a recent survey).

Eyes are important for the entertainment industry, who want to model them with potentially dramatic appearance. Bérard et al. [23] represents the state-of-the-art in capturing eye models for actor digital-doubles. They used a hybrid reconstruction method to separately capture both the transparent corneal surface and diffuse sclera in high detail, and recorded deformations of the eyeball's interior structures.

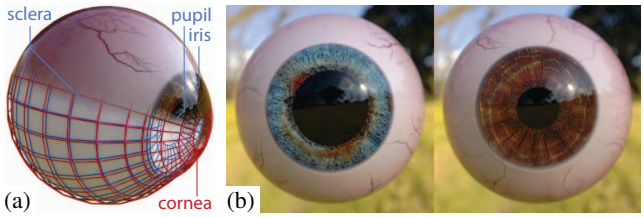


Figure 4: Our eye model includes the sclera, pupil, iris, and cornea (a) and can exhibit realistic variation in both shape (pupillary dilation) and texture (iris color, scleral veins) (b).

Visually-appealing eyes are also important for the video-game industry. Jimenez et al. [24] recently developed techniques for modeling eye wetness, refraction, and ambient occlusion in a standard rasterization pipeline, showing that approximations are sufficient in many cases.

Aside from visual effects, previous work has used 3D models to examine the eye from a medical perspective. Sagar et al. [25] built a virtual environment of the eye and surrounding face for mechanically simulating surgery with finite element analysis. Priamikov and Triesch [26] built a 3D biomechanical model of the eye and its interior muscles to understand the underlying problems of visual perception and motor control. Eye models have also been used to evaluate geometric gaze estimation algorithms, allowing individual parts of an eye tracking system to be evaluated separately. For example, Świrski and Dodgson [10] used a rigged head model and reduced eyeball model to render ground truth images for evaluating pupil detection and tracking algorithms.

3. Dynamic Eye-Region Model

We developed a realistic dynamic eye-region model which can be randomly posed to generate fully labeled training images. Our goals were realism and controllability, so we combined 3D head scan geometry with our own posable eyeball model – Figure 2 provides an overview of the model preparation process. For the resulting training data to be useful, it should be representative of real-world variety. We therefore aimed to model the continuous changes in appearance that the face and eyes undergo during eye movement, so they are accurately represented in close-up synthetic eye images. This is more challenging than simply rendering a collection of static models, as dynamic geometry must be correctly topologized and rigged to be able to deform continuously. Next, we present our anatomically inspired eyeball model and the procedure for converting a collection of static 3D head scans into dynamic eye-region models.

3.1. Simplified Eyeball Model

Our eye model consists of two parts (see Figure 4a). The outer part (red wireframe) approximates the eye's overall shape with two spheres ($r_1 = 12$mm, $r_2 = 8$mm [22]), the latter representing the corneal bulge. To avoid a discontinuous seam between spheres, their meshes were joined, and the vertices along the seam were smoothed to minimize differences in face-angle. This outer part is transparent, refractive ($n = 1.376$), and partially reflective. The sclera's bumpy surface is modeled with smoothed solid noise functions, and applied using a *displacement map* – a 2D scalar function that shifts a surface in the direction of its normal [27]. The inner part (blue wireframe) is a flattened sphere – the planar end represents the iris and pupil, and the rest represents the sclera, the white of the eye. There is a 0.5mm gap between the two parts which accounts for the thickness of the cornea.

Eyes vary in both shape (pupillary dilation) and texture (iris color and scleral veins). To model shape variation we use *blend shapes* to interpolate between several different poses created for the same topological mesh [28]. We created blend shapes for dilated and constricted pupils, as well as large and small irises to account for a small amount (10%) of variation in iris size. We vary the texture of the eye by compositing images in three separate layers: *i*) a *sclera* tint layer (white, pink, or yellow); *ii*) an *iris* layer with four different photo-textures (amber, blue, brown, grey); and *iii*) a *veins* layer (blood-shot or clear).

3.2. 3D Head Scan Acquisition

For an eye-region rendering to be realistic, it must also feature realistic nearby facial detail. While previous approaches used lifelike artist-created models [10], we rely on high-quality head scans captured by a professional photogrammetry studio (10K diffuse color textures, 0.1mm resolution geometry)[1]. Facial appearance around the eye varies dramatically between people as a result of different eye-shapes (e.g. round vs hooded), orbital bone structure (e.g. deep-set vs protruding), and skin detail (wrinkled vs smooth). Therefore our head models (see Figure 3) cover gender, ethnicity and age. As can be seen in Figure 2a, the cornea of the original head scan has been incorrectly reconstructed by the optical scanning process. This is because transparent surfaces are not directly visible, so cannot be reconstructed in the same way as diffuse surfaces, such as skin. For images to represent a wide range of gaze directions, the eyeball needed to be posed separately from the face geometry. We therefore removed the scanned eyeball from the mesh, and placed our own eyeball approximation in its place.

3.3. Eye-Region Geometry Preparation

While the original head scan geometry is suitable for being rendered as a static model, its high resolution topology cannot be easily controlled for changes in eye-region shape. Vertical saccades are always accompanied by eyelid motion, so we need to control eyelid positions according to the gaze vector. To do this, we need a more efficient (low-resolution) geometric representation of the eye-region, where edge loops

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

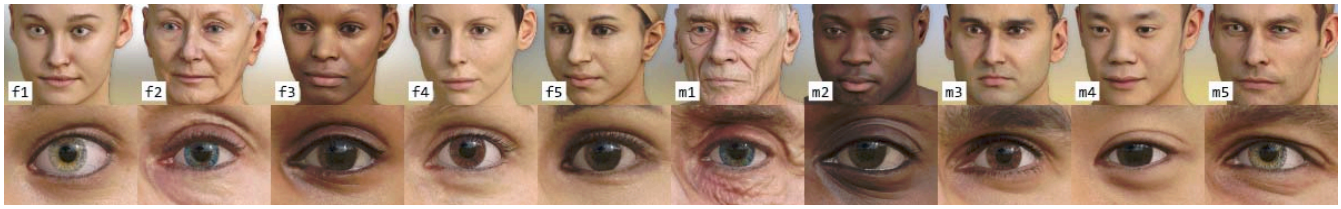


Figure 3: Our collection of head models and corresponding close-ups of the eye regions. The set exhibits a good range of variation in eye shape, surrounding bone structure, skin smoothness, and skin color.

flow around the natural contours of facial muscles. This leads to more realistic animation as mesh deformation matches that of actual skin tissue and muscles [28].

We therefore *retopologized* the face geometry using a commercial semi-automatic system[2], and transferred the original 10K color textures. As can be seen in Figure 2b, this way edge loops followed the exterior eye muscles, allowing for realistic eye-region deformations. This retopologized low-poly mesh ($\sim$10K polys) has lost the skin detail of the original scan, like wrinkles (see Figure 2c). These were restored with a displacement map computed from the scanned geometry [27]. There is normally no visible gap between eyeball and skin. However, as a consequence of removing the eyeball from the original scan, the retopologized mesh did not necessarily meet the eyeball geometry (see Figure 2b). To compensate for this, the mesh's eyelid vertices were automatically displaced along their normals to their respective closest positions on the eyeball geometry (see Figure 2c). This prevented unwanted gaps between the models, even after changes in pose. The face geometry was then assigned physically-based materials, including subsurface scattering to approximate the penetrative light transfer properties of skin, and a glossy component to simulate its oily surface.

3.4. Modeling Eyelid Motion and Eyelashes

We model eyelid motion using blend shapes for upwards-looking and downwards-looking eyelids, and interpolating between them based on the global pitch of the eyeball model. This makes our face-model dynamic, allowing it to continuously deform to match eyeball poses. Rather than rendering a single or perhaps several discrete head scans representing a particular gaze vector [9], we can instead create training data with a dense distribution of facial deformation. Defining blend shapes through vertex manipulation can be a difficult and time-consuming task but fortunately, only two are required and they have small regions of support. As the tissue around the eye is compressed or stretched, skin details like wrinkles and folds are either attenuated or exaggerated (see Figure 5). We modeled this by using smoothed color and displacement textures for downwards-looking eyelids, removing any wrinkles. These blend shape and texture modifications were carried out using photos of the same heads looking up and down as references.

[2]ZBrush ZRemesher 2.0, Pixologic, 2015

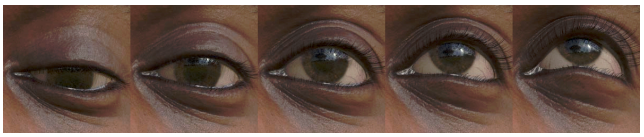


Figure 5: Eyelids are posed by interpolating between blend shapes based on gaze direction (`m2` as example).

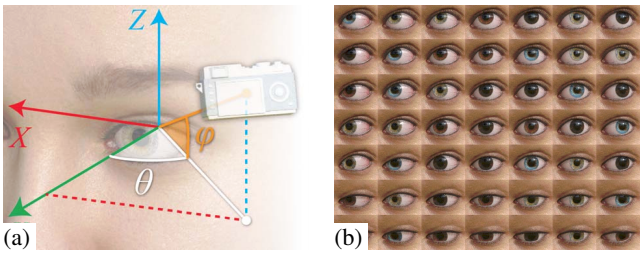


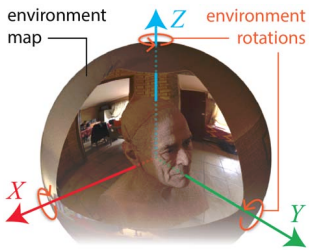
Figure 6: The camera is positioned to simulate changes in head pose (a). At each position, we render many eye images for different gaze directions by posing the eyeball model (b).

Eyelashes are short curved hairs that grow from the edges of the eyelids. These can occlude parts of the eye and affect eye tracking algorithms, so are simulated as part of our comprehensive model. We followed the approach of Świrski and Dodgson [10], and modeled eyelashes using directed hair particle effects. Particles were generated from a control surface manually placed below the eyelids. To make them curl, eyelash particles experienced a slight amount of gravity during growth (negative gravity for the upper eyelash).

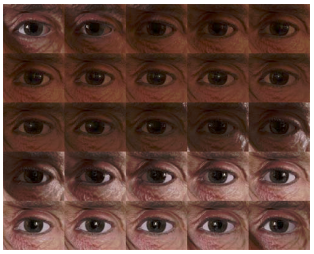
4. Training Data Synthesis

In-the-wild images exhibit large amounts of appearance variability across different viewpoints and illuminations. Our goal was to sufficiently sample our model across these degrees of variation to create representative image datasets. In this section we first describe how we posed our viewpoint and model, and explain our approach for using image-based lighting [29] to model a wide range of realistic environments. We then describe our landmark annotation process and finally discuss the details of our rendering setup.

(a) The four HDR environment maps we use for realistic lighting: bright/cloudy outdoors, and bright/dark indoors



(b) The environment is rotated to simulate different head poses



(c) Renders using a single environment, rotated about Z

Figure 7: Appearance variation from lighting is modelled with poseable high dynamic range environment maps [29].

4.1. Posing the Model

For a chosen eye-region model, each rendered image is determined by parameters $(\mathbf{c}, \mathbf{g}, L, E)$: 3D camera position $\mathbf{c}$; 3D gaze vector $\mathbf{g}$; lighting environment L; and eye model configuration E. Camera positions $\mathbf{c}$ were chosen by iterating over spherical coordinates (θ, ϕ), centered around the eyeball center (see Figure 6). We used orthographic rendering, as this simulates an eye region-of-interest being cropped from a wide-angle camera image. At each camera position $\mathbf{c}$, we rendered multiple images with different 3D gaze vectors to simulate the eye looking in different directions. Examples with fixed L are shown in Figure 6b. Gaze vectors $\mathbf{g}$ were chosen by first pointing the eye directly at the camera (simulating eye-contact), and then modifying the eyeball's pitch (α) and yaw (β) angles over a chosen range. Within E we randomly configure iris color and pose eyelids according to $\mathbf{g}$. For our generic dataset, we rendered images with up to $45°$ horizontal and vertical deviation from eye-contact, in increments of $10°$. As we posed the model in this way, there was the possibility of rendering "unhelpful" images that either simulate impossible scenarios or are not useful for training. To avoid violating anatomical constraints, we only rendered images for valid eyeball rotations $|\alpha| \leq 25°$ and $|\beta| \leq 35°$ [30]. Before rendering, we also verified that the projected 2D pupil center in the image was within the 2D boundary of the eyelid landmarks – this prevented us from rendering images where too little of the iris was visible.

4.2. Creating Realistic Illumination

One of the main challenges in computer vision is illumination invariance – a good system should work under a range of real-life lighting conditions. We realistically illuminate our eye-model using *image-based lighting*, a technique where high dynamic range (HDR) panoramic images are used to provide light in a scene [29]. This works by photographically capturing omni-directional light information, storing it in a texture, and then projecting it onto a sphere around the object. When a ray hits that texture during rendering, it takes that texture's pixel value as light intensity. At render time we randomly chose one of four freely available HDR environment images[3] to simulate a range of different lighting conditions (see Figure 7). The environment is then randomly rotated to simulate a continuous range of head-pose, and randomly scaled in intensity to simulate changes in ambient light. As shown in Figure 7c, a combination of hard shadows and soft light can generate a range of appearances from only a single HDR environment.

4.3. Eye-Region Landmark Annotation

For eye shape registration, we needed additional ground-truth annotations of eye-region landmarks in the training images. As shown in Figure 2d, each 3D eye-region was annotated once in 3D with 28 landmarks, corresponding to the eyelids (12), iris boundary (8), and pupil boundary (8). The iris and pupil landmarks were defined as a subset of the eyeball geometry vertices, so deform automatically with changes in pupil and iris size. The eyelid landmarks were manually labelled with a separate mesh that follows the seam where eyeball geometry meets skin geometry. This mesh is assigned shape keys and deforms automatically during eyelid motion. Whenever an image is rendered, the 2D image-space coordinates of these 3D landmarks are calculated using the camera projection matrix and saved.

4.4. Rendering Images

We use Blender's[4] inbuilt Cycles path-tracing engine for rendering. This Monte Carlo method traces the paths of many light rays per pixel, scattering light stochastically off physically-based materials in the scene until they reach illuminants. A GPU implementation is available for processing large numbers of rays simultaneously $(150/\text{px})$ to achieve noise-free and photorealistic images. We rendered a generic SynthesEyes dataset of 11,382 images covering $40°$ of viewpoint (i.e. head pose) variation and $90°$ of gaze variation. We sampled eye colour and environmental lighting randomly for each image. Each 120×80px rendering took 5.26s on average using a commodity GPU (Nvidia GTX660). As a result we can specify and render a cleanly-labelled dataset in under a day on a single machine – a fraction of the time taken by traditional data collection procedures [8].

[3] http://adaptivesamples.com/category/hdr-panos/
[4] The Blender Project – http://www.blender.org/

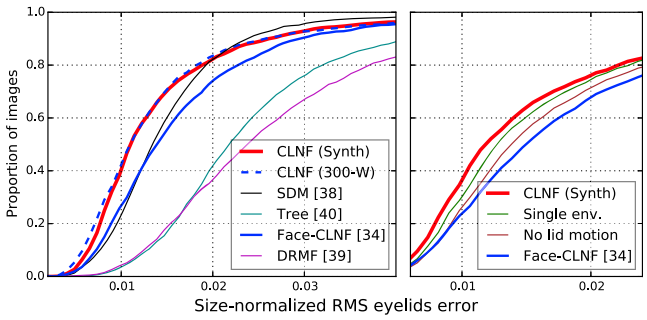


Figure 8: We outperform the state-of-the-art for eyelid-registration in the wild. The right plot shows how performance degrades for training data without important degrees of variation: realistic lighting and eyelid movement.

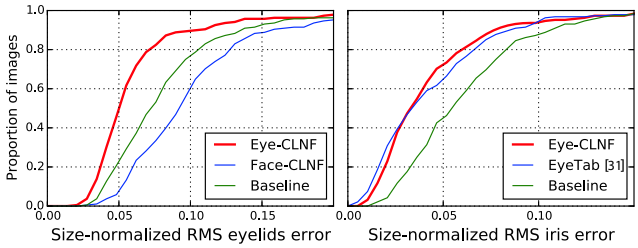


Figure 9: We perform comparably with state-of-the-art for iris-registration on in-the-wild webcam images.

5. Experiments

We evaluated the usefulness of our synthetic data generation method on two sample problems, eye-shape registration and appearance-based gaze estimation.

Eye-shape registration attempts to detect anatomical landmarks of the eye – eyelids, iris and the pupil. Such approaches either attempt to model the shape of the eye directly by relying on low-level image features, e.g. edges [31, 32] or by using statistically learnt deformable models [33]. Compared to Alabort-i Medina et al. [33], our dataset has been automatically labelled. This guarantees consistent labels across viewpoints and people, avoiding human error.

Appearance-based gaze estimation systems learn a mapping directly from eye image pixels to gaze direction. While most previous approaches focused on *person-dependent* training scenarios which require training data from the target user, recently more attention has been paid to *person-independent* training [8, 9, 34, 35]. The training dataset is required to cover the potential changes in appearance with different eye shapes, arbitrary head poses, gaze directions, and illumination conditions. Compared to Sugano et al. [9], our method can provide a wider range of illumination conditions which can be beneficial to handle the unknown illumination condition in the target domain.

5.1. Eye-Shape Registration

As our method can reliably generate consistent landmark location training data, we used it for training a Constrained Local Neural Field (CLNF) [36] deformable model. We conducted experiments to evaluate the generalizability of our approach on two different use cases: eyelid registration in-the-wild, and iris tracking from webcams.

Eyelid Registration In the Wild We performed an experiment to see how our system generalizes on unseen and unconstrained images. We used the validation datasets from the 300 Faces In-the-Wild (300-W) challenge [37] which contain labels for eyelid boundaries. We tested all of the approaches on the 830 (out of 1026) test images. We discarded images that did not contain visible eyes (occluded by hair or sunglasses) or where face detection failed for other comparison systems used in our experiment.

We trained CLNF patch experts using the generic SynthesEyes dataset and used the 3D landmark locations to construct a Point Distribution Model (PDM) using Principal Component Analysis. As our rendered images did not contain closed eyes we generated extra closed eye landmark labels by moving the upper eyelid down to lower one or meeting both eyelids halfway. We initialized our approach by using the face-CLNF [36] facial landmark detector. To compare using synthetic or real training images, we trained an eyelid CLNF model on 300-W images, but used the same PDM used for synthetic data (CLNF 300-W). We also compared our approach with the following state-of-the-art facial landmark detectors trained on in-the-wild data: CLNF [36], Supervised Descent Method (SDM) [38], Discriminative Response Map Fitting (DRMF) [39], and tree based face and landmark detector [40].

The results of our experiments can be seen in Figure 8, and example model fits are shown in Figure 10a. Errors were recorded as the RMS point-to-boundary distance from tracked eyelid landmarks to ground truth eyelid boundary, and were normalized by inter-ocular distance. First, our system CLNF Synth (Mdn $= 0.0110$px) trained on only 10 participants in four lighting conditions results in very similar performance to a system trained on unconstrained in-the-wild images, CLNF 300-W (Mdn $= 0.0110$px). Second, the results show the eye-specific CLNF outperformed all other systems in eye-lid localization: SDM (Mdn $= 0.0134$px), face-CLNF (Mdn $= 0.0139$px), DRMF (Mdn $= 0.0238$px), and Tree based (Mdn $= 0.0217$px). The first result suggests the importance of high-quality consistent labels. In addition, we perform well despite the fact our models do not exhibit emotion-related shape deformation, such as brow-furrowing, squinting, and eye-widening.

Our approach also allow us to examine what steps of the synthesis are important for generating good training data. We trained two further eye-specific CLNFs on different versions of SynthesEyes, one without eyelid motion and one with

only one fixed lighting condition. As can be seen in Figure 8, not using shape variation (Mdn $= 0.0129$px) and using basic lighting (Mdn $= 0.0120$px) lead to worse performance due to missing degrees of variability in training sets.

Eye-Shape Registration for Webcams While the 300-W images represent challenging conditions for eyelid registration they do not feature iris labels and are not representative of conditions encountered during everyday human-computer interaction. We therefore annotated sub-pixel eyelid and iris boundaries for a subset of MPIIGaze [8] (188 images), a recent large-scale dataset of face images and corresponding on-screen gaze locations collected during everyday laptop use over several months [8]. Pupil accuracy was not evaluated as it was impossible to discern in most images.

We compared our eye-specific CLNF (CLNF Synth) with EyeTab [31], a state-of-the-art shape-based approach for webcam gaze estimation that robustly fits ellipses to the iris boundary using image-aware RANSAC [32]. Note we did not compare with other systems from the previous experiment as they do not detect irises. We used a modified version of the author's implementation with improved eyelid localization using CLNF [36]. As a baseline, we used the mean position of all 28 eye-landmarks following model initialization. Eyelid errors were calculated as RMS distances from predicted landmarks to the eyelid boundary. Iris errors were calculated by least-squares fitting an ellipse to the tracked iris landmarks, and measuring distances only to visible parts of the iris. Errors were normalized by the eye-width, and are reported using average eye-width (44.4px) as reference.

As shown in Figure 9, our approach (Mdn $= 1.48$px) demonstrates comparable iris-fitting accuracy with EyeTab (Mdn $= 1.44$px). However, CLNF Synth is more robust, with EyeTab failing to terminate in 2% of test cases. As also shown by the 300-W experiment, the eye-specific CLNF Synth localizes eyelids better than the face-CLNF. See Figure 10b for example model fits.

5.2. Appearance-Based Gaze Estimation

To evaluate the suitability of our synthesis method for appearance-based gaze estimation we performed a cross-dataset experiment as described by Zhang et al. [8]. We synthesized training images using the same camera settings as in the UT dataset [9]. The head pose and gaze distributions for the three datasets are shown in Figure 11. We then trained the same convolutional neural network (CNN) model as in [8] on both synthetic datasets and evaluated their performance on MPIIGaze. As shown in Figure 12, the CNN model trained on our generic SynthesEyes dataset achieved similar performance ($\mu = 13.91°$) as the model trained on the UT dataset ($\mu = 13.55°$). This confirms that our approach can synthesize data that leads to comparable results with previous synthesis procedures [9]. Note from Figure 12 that

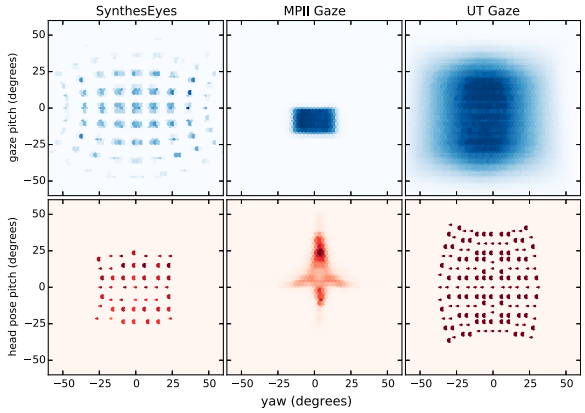


Figure 11: The gaze direction (first row) and head pose (second row) distributions of different datasets: SynthesEyes, MPIIGaze [8], and UT Multiview [9].

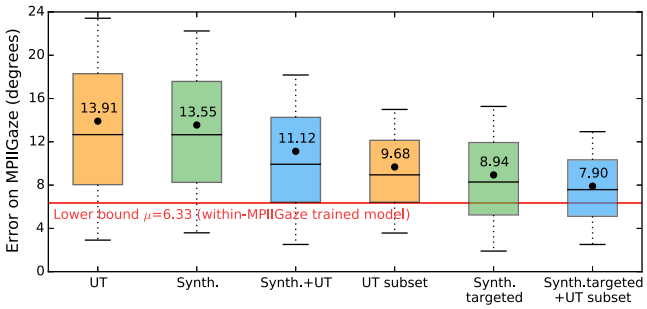


Figure 12: Test performance on MPIIGaze; x-axis represents training set used. Dots are mean errors, and red line represents a practical lower-bound (within-dataset cross-validation score). Note how combining synthetic datasets for training lead to improved performance (blue plots).

there is still a performance gap between this cross-dataset and the within-dataset training (red line).

While it is in general important to cover a wide range of head poses to handle arbitrary camera settings , if the target setting is known in advance, e.g. laptop gaze interaction as in case of MPIIGaze, it is possible to target data synthesis to the expected head pose and gaze ranges. To study the ability of our method to perform such a targeting, we rendered an additional dataset (SynthesEyes targeted) for a typical laptop setting ($10°$ pose and $20°$ gaze variation). For comparison, we also re-sampled the entire UT dataset to create a subset (UT subset) that has the same gaze and head pose distribution as MPIIGaze. To make a comparison assuming the same number of participants, we further divided the UT subset into five groups with 10 participants each, and averaged the performance of the five groups for the final result. As shown in the third and forth bars of Figure 12, having similar head pose and gaze ranges as the target domain improves performance compared to the generic datasets. Trained on our SynthesEyes dataset the CNN achieves a statistically

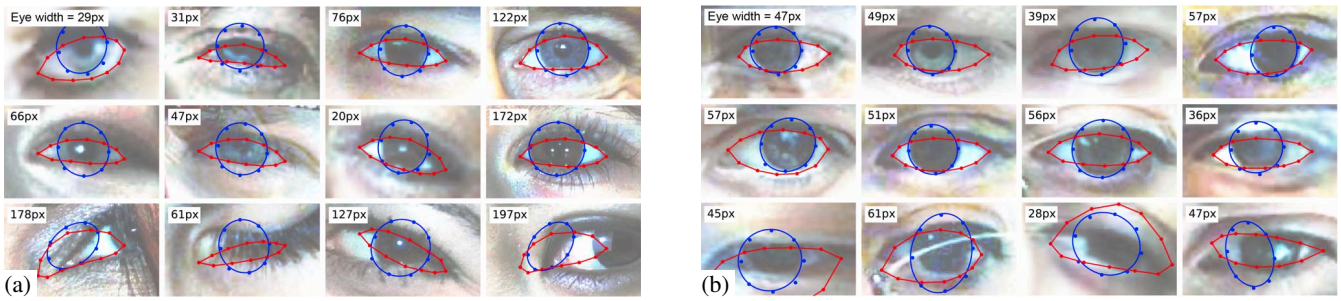


Figure 10: Example fits of our SynthesEyes eye-CLNF on in-the-wild images (a) and webcam images (b). The top two rows illustrate successful eye-shape registrations, while the bottom row illustrates failure cases, including unmodelled occlusions (hair), unmodelled poses (fully closed eye), glasses, and incorrect model initialization. Note our algorithm generalizes well to eye images of different sizes.

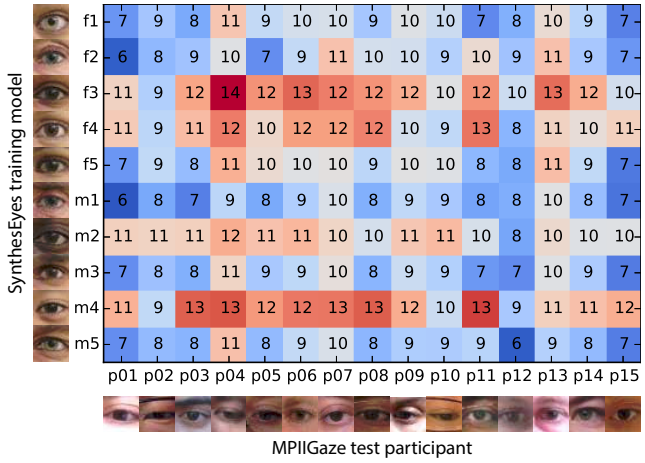


Figure 13: Per–eye-model gaze estimation mean errors on MPIIGaze. Red represents worst scores. Note how some eye-models have proved more useful than others for training.

significant performance improvement over the UT dataset of 0.74° (Wilcoxon signed-rank test: $p<0.0001$).

These results suggest that neither SynthesEyes nor the UT dataset alone capture all variations present in the test set, but different ones individually. For example, while we cover more variations in lighting and facial appearance, the UT dataset contains real eye movements captured from more participants. Recent works by Fu and Kara [13] and Peng et al. [16] demonstrated the importance of fine-tuning models initially trained on synthetic data on real data to increase performance. Finally, we therefore evaluated the performance by training and fine-tuning using both datasets (see Figure 12). We first trained the same CNN model on the SynthesEyes dataset and fine-tuned the model using the UT dataset. This fine-tuned model achieved better performances in both settings (untargeted $\mu = 11.12°$, targeted $\mu = 7.90°$). The performance of the untargeted case significantly outperformed the state-of-the-art result [8] (Wilcoxon signed-rank test: $p<0.0001$), and indicates a promising way for a future investigation to fill the performance gap.

Person-Specific Appearance Appearance-based gaze estimation performs best when trained and tested on the same person, as the training data includes the same eye appearances that occur during testing. However, eye images from SynthesEyes and MPIIGaze can appear different due to differences in eye-shape and skin color. To examine the effects of this we conducted a second experiment where we trained 10 separate systems (one trained on each SynthesEyes eye model) and tested on each participant in MPIIGaze. The results can be seen in Figure 13.

This plot illustrates which SynthesEyes models were useful for training and which ones were not. As we can see, training with certain eye models lead to poor generalization, for example `f3`, `m2`, and `m4`, perhaps due to differences in skin-tone and eye-shape. Also, total errors for some target participants are lower than for others, perhaps because of simpler eye-region shape that is matched to the training images. Although intuitive, these experiments further confirm the importance of correctly covering appearance variations in the training data. They also open up potential directions for future work, including person-specific adaptation of the renderings and gaze estimation systems.

6. Conclusion

We presented a novel method to synthesize perfectly labelled realistic close-up images of the human eye. At the core of our method is a computer graphics pipeline that uses a collection of dynamic eye-region models obtained from head scans to generate images for a wide range of head poses, gaze directions, and illumination conditions. We demonstrated that our method outperforms state-of-the-art methods for eye-shape registration and cross-dataset appearance-based gaze estimation in the wild. These results are promising and underline the significant potential of such learning-by-synthesis approaches particularly in combination with recent large-scale supervised methods.

References

[1] M. Argyle and J. Dean, "Eye-Contact, Distance and Affiliation." *Sociometry*, 1965.

[2] P. Majaranta and A. Bulling, *Eye Tracking and Eye-Based Human-Computer Interaction*, ser. Advances in Physiological Computing. Springer, 2014.

[3] A. Bulling, J. A. Ward, H. Gellersen, and G. Tröster, "Eye Movement Analysis for Activity Recognition Using Electrooculography," *IEEE TPAMI*, 2011.

[4] D. P. Papadopoulos, A. D. Clarke, F. Keller, and V. Ferrari, "Training object class detectors from eye tracking data," in *ECCV*, 2014, pp. 361–376.

[5] H. Sattar, S. Müller, M. Fritz, and A. Bulling, "Prediction of Search Targets From Fixations in Open-World Settings," in *Proc. CVPR*, 2015.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014, pp. 580–587.

[7] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *NIPS*, 2014, pp. 487–495.

[8] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-Based Gaze Estimation in the Wild," in *CVPR*, 2015.

[9] Y. Sugano, Y. Matsushita, and Y. Sato, "Learning-by-Synthesis for Appearance-based 3D Gaze Estimation," in *CVPR*, 2014.

[10] L. Świrski and N. Dodgson, "Rendering synthetic ground truth images for eye tracker evaluation," in *ETRA*, 2014.

[11] R. Okada and S. Soatto, "Relevant feature selection for human pose estimation and localization in cluttered images," in *ECCV*, 2008.

[12] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from a single depth image," in *CVPR*, 2011.

[13] L. Fu and L. B. Kara, "Neural network-based symbol recognition using a few labeled samples," *Computers & Graphics*, vol. 35, no. 5, pp. 955–966, 2011.

[14] J. Yu, D. Farin, C. Krüger, and B. Schiele, "Improving person detection using synthetic training data," in *ICIP*, 2010.

[15] J. Liebelt and C. Schmid, "Multi-view object class detection with a 3d geometric model," in *CVPR*, 2010, pp. 1688–1695.

[16] X. Peng, B. Sun, K. Ali, and K. Saenko, "Exploring invariances in deep convolutional neural networks using synthetic images," *arXiv preprint*, 2014.

[17] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "3D constrained local model for rigid and non-rigid facial tracking," in *CVPR*, 2012.

[18] L. A. Jeni, J. F. Cohn, and T. Kanade, "Dense 3D Face Alignment from 2D Videos in Real-Time," *FG*, 2015.

[19] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," in *CVPR*, 2011.

[20] B. Kaneva, A. Torralba, and W. Freeman, "Evaluation of image features using a photorealistic virtual world," in *ICCV*, 2011, pp. 2282–2289.

[21] G. Stratou, A. Ghosh, P. Debevec, and L.-P. Morency, "Effect of illumination on automatic expression recognition: a novel 3D relightable facial database," in *FG*, 2011.

[22] K. Ruhland, S. Andrist, J. Badler, C. Peters, N. Badler, M. Gleicher, B. Mutlu, and R. Mcdonnell, "Look me in the eyes: A survey of eye and gaze animation for virtual agents and artificial systems," in *Eurographics*, 2014, pp. 69–91.

[23] P. Bérard, D. Bradley, M. Nitti, T. Beeler, and M. Gross, "Highquality capture of eyes," *ACM TOG*, 2014.

[24] J. Jimenez, E. Danvoye, and J. von der Pahlen, "Photorealistic eyes rendering," SIGGRAPH Advances in Real-Time Rendering, 2012.

[25] M. Sagar, D. Bullivant, G. Mallinson, and P. Hunter, "A virtual environment and model of the eye for surgical simulation," in *Computer graphics and interactive techniques*, 1994.

[26] A. Priamikov and J. Triesch, "Openeyesim - a platform for biomechanical modeling of oculomotor control," in *ICDL-Epirob*, Oct 2014, pp. 394–395.

[27] A. Lee, H. Moreton, and H. Hoppe, "Displaced subdivision surfaces," in *SIGGRAPH*, 2000, pp. 85–94.

[28] V. Orvalho, P. Bastos, F. Parke, B. Oliveira, and X. Alvarez, "A facial rigging survey," in *Eurographics*, 2012, pp. 10–32.

[29] P. Debevec, "Image-based lighting," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 26–34, 2002.

[30] *MIL-STD-1472G Design Criteria Standard: Human Engineering*, Department of Defence, USA, January 2012.

[31] E. Wood and A. Bulling, "Eyetab: Model-based gaze estimation on unmodified tablet computers," in *ETRA*, 2014.

[32] L. Świrski, A. Bulling, and N. Dodgson, "Robust real-time pupil tracking in highly off-axis images," in *ETRA*, 2012.

[33] J. Alabort-i Medina, B. Qu, and S. Zafeiriou, "Statistically learned deformable eye models," in *ECCVW*, 2014.

[34] K. A. Funes Mora and J.-M. Odobez, "Person independent 3D gaze estimation from remote RGB-D cameras," in *ICIP*, 2013.

[35] T. Schneider, B. Schauerte, and R. Stiefelhagen, "Manifold alignment for person independent appearance-based gaze estimation," in *Proc. ICPR*. IEEE, 2014, pp. 1167–1172.

[36] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "Constrained local neural fields for robust facial landmark detection in the wild," in *ICCVW*, 2013.

[37] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *ICCVW*, 2013, pp. 397–403.

[38] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *CVPR*, 2013.

[39] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Robust discriminative response map fitting with constrained local models," in *CVPR*, 2013.

[40] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *CVPR*, 2012.