Rendering of Eyes for Eye-Shape Registration and Gaze Estimation

Anonymous ICCV submission

Paper ID 1113

Abstract

Images of the eye are key in several computer vision problems, such as facial feature localization 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 demonstrate the benefits of our synthesized training data (SynthesEyes) by out-performing state-of-the-art methods for eye-shape registration in the wild, and achieving competitive performance on appearance-based gaze estimation. Furthermore, we show that its important to include realistic illumination and shape variation in training data.

1. Introduction

Machine learning methods that leverage large amounts of training data currently perform best for many problems in computer vision, such as object detection, scene recognition, or gaze estimation [1, 2, 3]. 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.

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 eyes and their movements convey our attention and play a role in communicating social and emotional information [4]. Therefore they are important for a range of applications including gaze-based human-computer interaction [5], visual behaviour monitoring [6], and – more recently – collaborative human-computer vision systems [7, 8, 9]. 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).

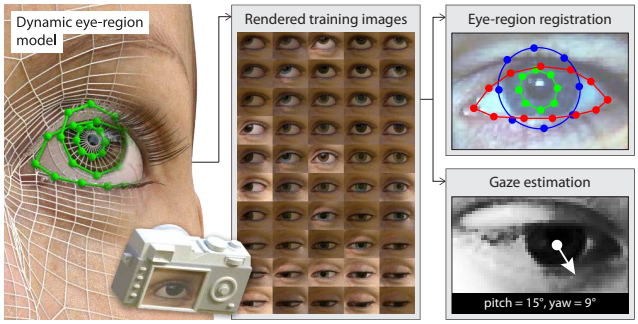


Figure 1: We render a large number of photorealistic images of eyes using a dynamic eye region model. These are used as training data for eye-shape registration and appearance-based gaze estimation.

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 modelling illumination changes or accounting for the varying material properties of the face [10, 11]. 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 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 pro-

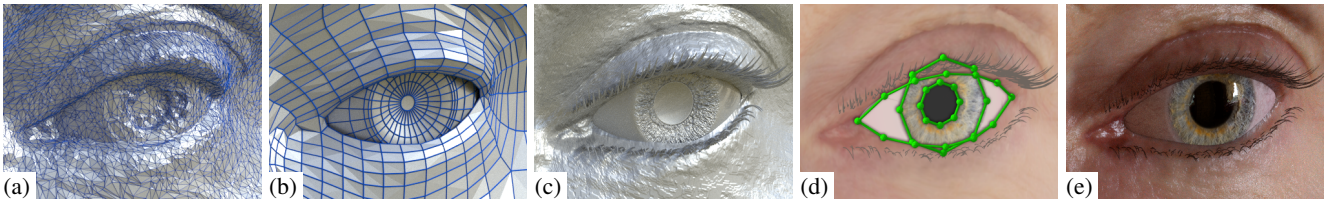


Figure 2: An overview of our model preparation process: Dense 3D head scans (1.4 million polygons) (a) are first retopologised into an optimal form for animation (9,005 polygons) (b). High resolution skin surface details are restored by displacement maps (c), and 3D iris and eyelid landmarks are annotated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

cess 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. evaluation of iris-biometrics or gaze correction.

The specific contributions of this work are threefold. We 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 for eye-shape registration, where we outperform the state-of-the-art, and perform competitively in challenging cross-dataset gaze estimation experiments. 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 works on 1) learning using synthetic data and 2) computational modelling of the eyes.

2.1. Learning Using Synthetic Data

Learning-based approaches have been recognised as a promising solution for various problems in computer vision. But it remains challenging for such approaches to handle unknown test data and their performance often depends on how well the test data distribution is covered by the training set. Since recording training data that covers the whole range of potential test data variation is challenging, synthesized training data has been used instead. Previous work demonstrates that synthetic training data is beneficial for many tasks in computer vision including body pose estimation [12, 13], object detection/recognition [14, 15], and facial landmark localizaion [16, 17, 18]. Since faces exhibit a huge degree of color and texture variability, some previous approaches have side-stepped this by relying on depth images [19, 17], and synthesizing depth images of the head using existing datasets or a deformable head-shape model. Recent work has also synthesised combined color and geometry data by sampling labelled depth videos for training a dense 3D facial landmark detector [18].

As discussed by Kaneva et al. [16], one of the most important factors to consider is the realism of synthesised 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. [3] revealed that estimation accuracy significantly drops when the test data is obtained from a different environment. Similarly to facial expression recognition [20], illumination effects are a critical factor for computer vision, and the change of viewpoints is not enough to cover the test data variability. In contrast, our model allows the synthesis of realistic lighting effects – an important degree of variation for performance improvements in eye-shape registration.

Most similar to this work, Sugano et al. [10] used 3D reconstructions of eye regions to synthesise multi-view training data for appearance-based gaze estimation. One limitation of that work is that they do not provide a parametric model. Their data is essentially 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. The scope of learning-by-synthesis with a realistic eye model is not limited to appearance-based gaze estimation; the model can be also applied to other problems, e.g. eye shape registration. Since our model is fully controllable, it can be used to synthesise 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 Modelling 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 modelling and rendering eyes (see [21] for a recent survey).

Eyes are important for the entertainment industry, who want to model them with potentially dramatic appearance. Bérard et al. [22] 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 detil, and recorded deformations of the eyeball's interior structures. Visually-appealing eyes are also important for the video-game industry. Jimenez et al. [23] recently developed techniques for modelling eye wetness, refraction, and ambient occlusion in a standard rasterisation 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. [24] built a virtual environment of the eye and surrounding face for mechanically simulating surgery with finite element analysis. Priamikov and Triesch [25] built a 3D biomechanical model of the eye and its interior muscles to understand the underlying problems of visual perception and motor control. ye 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 [26] 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. Our aim therefore was 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. In this section we first present our anatomically inspired computer graphics eyeball model, and then explain our procedure for converting a collection of static 3D head scans into dynamic eye-region models that can assume a range of realistic poses.

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 [21]), 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 modelled with smoothed solid noise functions,

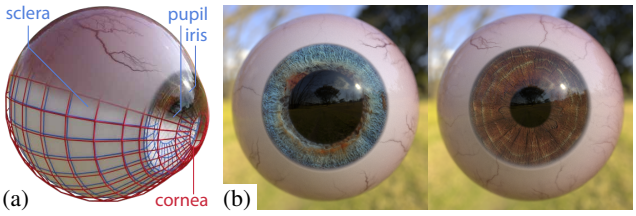


Figure 4: Our eye model includes the sclera, pupil, iris, and cornea (a) and can exhibit realistic variation in both shape (pupillary dilation) and texture (iris color, scleral veins) (b).

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 exhibit variation in both shape (pupillary dilation) and texture (iris color and scleral veins). To model shape variation we use *blend shapes* – an animation technique where several different poses are created for the same topological mesh, and then interpolated between [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 face detail. While previous approaches used lifelike artist-created models, for example [26], we instead 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 both genders with a variety of ethnicities and ages.

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. We also wanted images representing a wide range of eye-gaze directions, so we needed to be able to pose the eyeball separately from the face geometry. We therefore removed the scanned eyeball from the mesh, and placed our own eyeball approximation

[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.

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 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 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 into a more optimal form using a commercial semi-automatic system[2]. As can be seen in Figure 2b, edge loops now follow the exterior eye muscles, allowing for realistic eye-region deformations. This retopologized low-poly mesh has now lost the skin surface detail of the original scan, like wrinkles and creases (see Figure 2c). These were restored with a displacement map computed from the scanned geometry [27]. Although they are two separate organs, 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 will not necessarily meet the eyeball geometry (see Figure 2b). To compensate for this, the face mesh's eyelid vertices are automatically displaced along their normals to their respective closest positions on the eyeball geometry (see Figure 2c). This prevents unwanted gaps between the models, even after changes in pose. The face geometry is 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. Modelling 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 [10], 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.

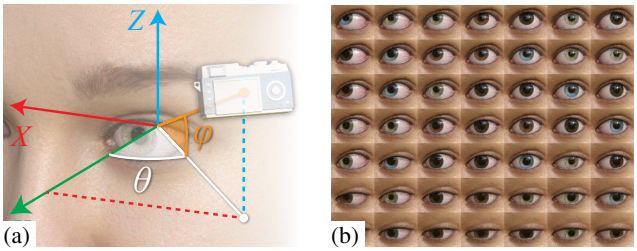
[2]ZBrush ZRemesher 2.0, Pixologic, 2015

Figure 5: Eyelids are posed by interpolating between blend shapes based on gaze direction. Note how we simulate the folding of the skin above and below the eye.

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 [26], and modelled 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) (b)

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).

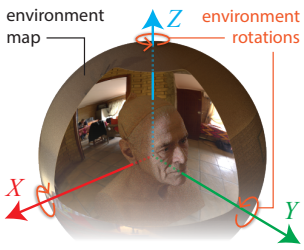
4.1. Posing the Model

For a chosen eye-region model configuration, each rendered image is determined by parameters $(\mathbf{c}, \mathbf{g}, L)$: 3D camera position $\mathbf{c}$; 3D gaze vector $\mathbf{g}$; and lighting environment L. Camera positions $\mathbf{c}$ were chosen by iterating over spherical coordinates (r, 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, so we set $r = 1$ for convenience. 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. 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


(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

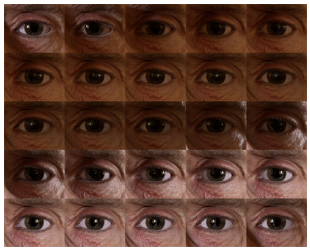

(c) Renders using a single environment, rotated about Z

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

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 need 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 eye corners (2), eyelids ($5+5$), 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 and eye corner 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

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

physically-based materials in the scene until they reach illuminants. A GPU implementation is available for processing large numbers of rays simultaneously (150/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 color 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 [3].

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 edge information [31, 32] or by using statistically learnt deformable models [33]. As our method can reliably generate consistent landmark location training data, we use it for Constrained Local Neural Field (CLNF) [34] deformable model training.

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 [35, 36, 10, 3]. 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. [10], 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

Eye-Shape Registration In the Wild We performed an experiment to see how our our system generalises on unseen and unconstrained images from the 300 Faces In-the-Wild (300-W) challenge [37] validation datasets 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 in some of the baselines. This lead to 1660 eye images for evaluation.

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

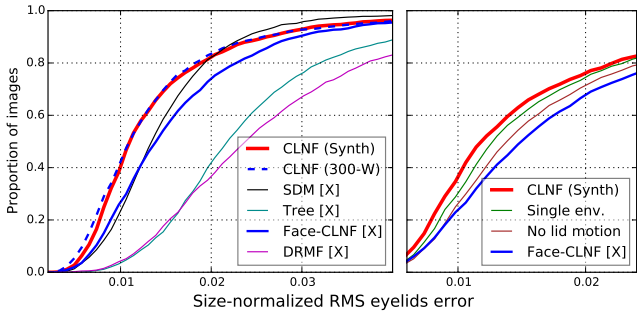


Figure 8: We outperform the state-of-the-art for eyelid-registration in the wild. The right plot shows how performance degrades for training data without important degrees of variation: realistic lighting and eyelid movement.

labels by moving the upper eyelid down to lower one or meeting both eyelids halfway. We initialised our approach by using the face-CLNF [34] 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 real world in-the-wild data: CLNF [34], 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 9a. 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, the results show the eye-CLNF (both synthetic ($\mathrm{Mdn} = 0.0110$) and real data ($\mathrm{Mdn} = 0.0110$) outperforming all other systems in eye-lid localization: SDM ($\mathrm{Mdn} = 0.0134$), face-CLNF ($\mathrm{Mdn} = 0.0139$), DRMF ($\mathrm{Mdn} = 0.0238$), and Tree based ($\mathrm{Mdn} = 0.0217$). Second, our system (CLNF Synth) trained on only ten participants in four lighting conditions results in very similar performance to a system trained on unconstrained in-the-wild images (CLNF 300-W). This suggests the importance of high-quality consistent labels.

Our data synthesis system also allows us to examine what steps of the synthesis approach are important for generating good training data. We trained two further eye-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 ($\mathrm{Mdn} = 0.0129$) and using basic lighting ($\mathrm{Mdn} = 0.0120$) leads to worse performance due to missing degrees of variability.

Eye-Shape Registration for Webcams While the 300-W images represent challenging conditions for eyelid registration, they are not representative of typical webcam-style

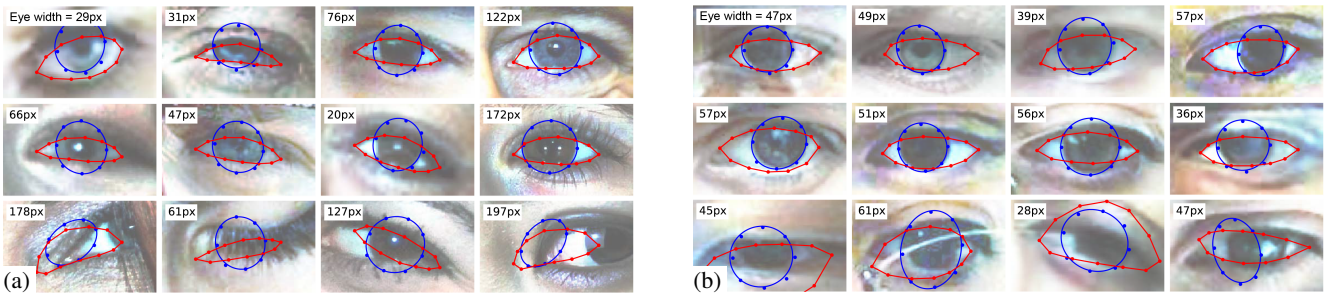


Figure 9: Example fits of our SynthesEyes eye-CLNF on in-the-wild images (a) and webcam images (b). The top two rows illustrate successful eye-shape registrations, while the bottom row illustrates failure cases, including unmodelled occlusions (hair), unmodelled poses (fully closed eye), glasses, and incorrect model initialization. Note our algorithm generalizes well to eye images of different sizes.

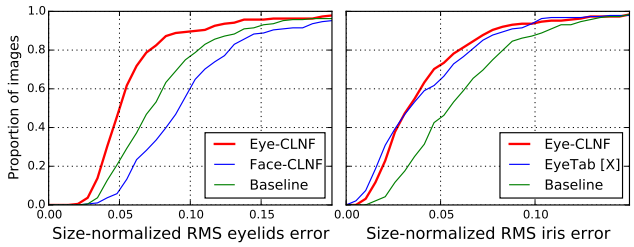


Figure 10: We perform comparably with state-of-the-art for iris-registration on in-the-wild webcam images.

images and do not feature iris labels. We therefore annotated sub-pixel eyelid and iris boundaries onto a subset of MPIIGaze (188 images), a recent large-scale dataset of face images and corresponding on-screen gaze locations collected during everyday laptop use over several months [3]. Pupil accuracy was not evaluated as it was impossible to discern in in most images. We compared our eye-CLNF 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]. We used a modified version of the author's implementation with improved eyelid localization using CLNF [34]. As a baseline, we used the mean position of all 28 eye-landmarks following model initialization. Eyelid errors were calculated as RMS distances from eyelid landmarks to the ground truth eyelid boudnary. Iris errors were calculated by first least-squares fitting an ellipse to the tracked iris landmarks, discretizing it, removing points outside ground truth and tracked eyelid boundaries, and then measuring RMS distances to the ground truth iris. This avoided calculating errors for parts of the iris which were occluded by the eyelid. Errors were normalized by the eye-width, and are reported using average eye-width (44.4px) as reference.

As shown in Figure 10, our approach (Mdn = 1.48px) demonstrates comparable iris-fitting accuracy with EyeTab (Mdn = 1.44px), a state-of-the-art algorithm for fitting ellipses to irises in low-quality images. However, our eye-

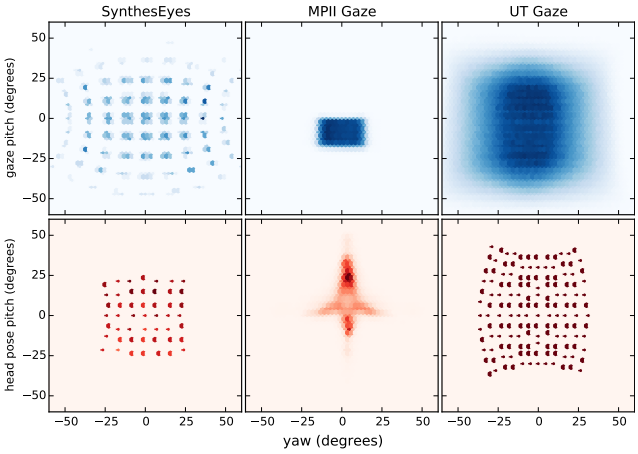


Figure 11: The gaze direction (first row, blue) and head pose (second row, red) distributions of different datasets.

CLNF is more robust, with EyeTab failing to terminate in 2% of test cases. As also shown by the 300-W experiment, our eye-CLNF localizes eyelids better than the face-CLNF. See Figure 9b for example model fits.

5.2. Appearance-Based Gaze Estimation

To evaluate the usage of our method on appearance gaze estimation, we perform the cross-dataset validation as described in Zhang et al. [3], where they train and test the model on different datasets. We synthesized our dataset using the same camera setting as UT dataset [10], and the same normalization scheme can be applied to the test data. The training data is fully compatible with UT dataset, and we can directly compare our SynthesEyes using the same Convolutional Neural Network (CNN) model [3].

As shown with the two bars at the far left of Figure 12, compared to the median gaze direction prediction error 12.7 degrees that trained with UT dataset, the model trained with our generic SynthesEyes dataset can achieve similar performance performance. This confirms that our method can generate a equivalent data for the appearance-based gaze

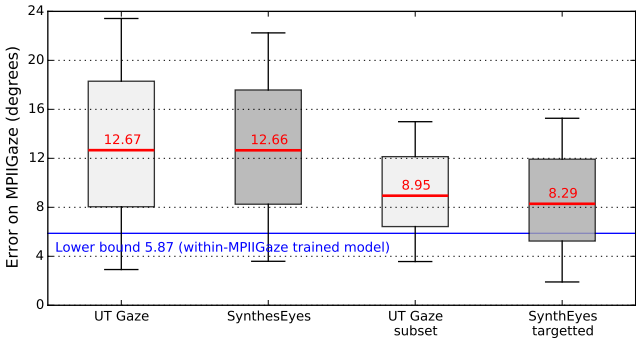


Figure 12: Gaze estimation performance on MPIIGaze using different training datasets. Red bar is median, and blue line represents a practical lower-bound (within-dataset cross-validation score).

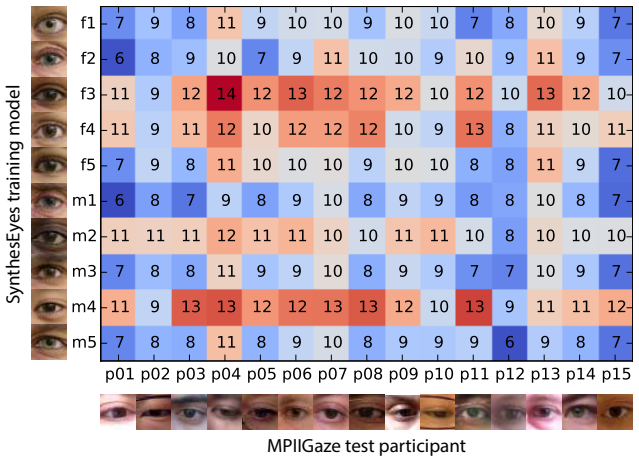


Figure 13: Gaze estimation performance on MPIIGaze using different training datasets. Red bar is median, and blue line represents a practical lower-bound (within-dataset cross-validation score).

estimation training. The blue background line of Figure 12 shows the leave-one-person-out cross validation score within the MPIIGaze dataset to show the lower bound of this scenario. These result indicates that both SynthesEyes and UT datasets still have their own error sources which is causing the performance gap.

One of the important factors related to the above gap is the head pose and gaze ranges. While it is important to cover a wide range of head poses to handle arbitrary camera settings, this can make the training task more difficult. If the target setting can be preliminary specified, as the laptop interaction case in Zhang et al. [3], it is practically possible to limit the synthesis of training data with minimum required head pose and gaze ranges. In order to analyze the effect of different head pose ranges, we rendered an additional targetted dataset guided by typical laptop use (10° pose variation, 20° gaze variation). For comparison, we also re-sample a subset of UT dataset as described in [3] that has the same gaze and head pose distribution as MPIIGaze. The head pose and gaze distributions are shown in Figure 11.

Another important difference between two datasets is the number of subjects in the dataset. In order to compare two approaches with the same number of subjects, we divide the UT Multiview subset into 5 groups with 10 subjects. Each group of this UT subset has 15,000 samples as with our SynthesEyes subset. We then average the performance of the 5 groups for the final result. As shown in the third and forth bars of Figure 12, in general having the similar head pose and gaze direction ranges of target domain can significantly improve the performance: Mdn $= 8.95°$ and Mdn $= 8.29°$ for UT and SynthesEyes respectively.

Person Specific Appearance Appearance-based gaze estimation performs best when trained and tested on the same person, as the training data includes the exact same eye-shape appearances that can be occur during testing. However, eye-region images from SynthesEyes and MPIIGaze can appear very different due to eye-shape, skin-texture, and skin color. We explored what this meant for gaze estimation accuracy. We conducted an experiment where we trained 10 separate systems (one for each SynthesEyes eye model) and tested them on MPIIGaze, recording average error for each MPIIGaze participant. The results can be seen in Figure 13 where we plot errors along with representative images of eye-regions seen during training and testing.

This plot reveals which SynthesEyes eye-region models have been useful for training, and which ones have not. As we can see, the test set contains few dark-skinned participants, so training with dark-skinned eye models (f3,m2) leads to poor test results. Similarly, learning with an asian-shaped eye-region model (m4) leads poor performance for non-asian-shape eye-region test subjects. Also, it can be said that some test participants have "easier" eye-regions than others, e.g. ones with a plain appearance (p15). Though intuitive, our experiments further confirm the importance of correctly covering appearance variations with training data.

6. Conclusion

We presented a novel method to synthesise 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 can outperform state-of-the-art methods for eye-shape registration in the wild, and achieve competitive performance on appearance-based gaze estimation. These results are promising and underline the significant potential of such learning-by-synthesis approaches particularly in combination with large-scale supervised methods.

References

[1] 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.

[2] 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.

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

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

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

[6] A. Bulling, J. A. Ward, H. Gellersen, and G. Tröster, "Eye movement analysis for activity recognition using electrooculography," *IEEE TPAMI*, 2011.

[7] K. Yun, Y. Peng, D. Samaras, G. J. Zelinsky, and T. L. Berg, "Studying relationships between human gaze, description, and computer vision," in *CVPR*, 2013, pp. 739–746.

[8] 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.

[9] H. Sattar, S. Müller, M. Fritz, and A. Bulling, "Prediction of search targets from fixations in open-world settings," in *Proc. CVPR*, 2015.

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

[11] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Head pose-free appearance-based gaze sensing via eye image synthesis," in *ICPR*, 2012, pp. 1008–1011.

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

[13] 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.

[14] J. Yu, D. Farin, C. Krger, 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] B. Kaneva, A. Torralba, and W. Freeman, "Evaluation of image features using a photorealistic virtual world," in *ICCV*, 2011, pp. 2282–2289.

[17] T. Baltrusaitis, P. Robinson, and L. Morency, "3d constrained local model for rigid and non-rigid facial tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2610–2617.

[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] 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.

[21] 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.

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

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

[24] 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.

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

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

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

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

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

[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.