ICCV #****

Rendering Photorealistic Training Images for Eye Tracking

Anonymous ICCV submission

Paper ID ****

Abstract

The ABSTRACT is to be in fully-justified italicized text, at the top of the left-hand column, below the author and affiliation information. Use the word "Abstract" as the title, in 12-point Times, boldface type, centered relative to the column, initially capitalized. The abstract is to be in 10-point, single-spaced type. Leave two blank lines after the Abstract, then begin the main text. Look at previous ICCV abstracts to get a feel for style and length.

1. Introduction

Machine learning approaches that leverage large amounts of image data are currently the best solutions to many problems in computer vision **[cite]** . However, capturing or collecting images can be extremely time consuming, especially for new areas of research without pre-existing datasets. Supervised learning approaches then require that the images are labelled. This annotation process can be expensive and tedious, and there is no guarantee the labels will be correct.

In this paper we describe our approach for generating photorealistic training data, and then present and evaluate two systems trained on SynthesEyes: an eye-region specific deformable model and an appearance-based gaze estimator. These systems are case studies that show how we leverage the degrees of control made available by rendering our training data to easily and quickly generate high quality training datasets.

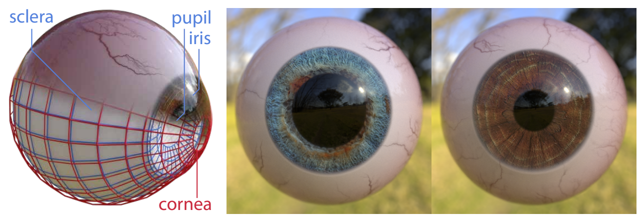
2. Related work

2.1. Synthetic data

[1] – uses rendered videos of eyes to evaluate eye tracking algorithms.

[2] – relit 3d face scans to study the effect of illumination on automatic expression recognition.

[3] – train head pose estimator on only synthetic depth data.



(a) 3D eye model (b) Pupil dilation and iris color variation

Figure 1: Our realistic eye model is capable of expressing degrees of variability seen in real life.

2.2. Deformable eye model

[4] – trained a detailed deformable eye region model on in-the-wild images.

2.3. Gaze estimation

[5] – regression with features of 3d pupil centers and eye-contours (the eyelids) for gaze estimation. Use multiple cameras and IR lights.

3. Synthetic data generation

In this section we first present our anatomically inspired CG eyeball model, and then explain our novel procedure for preparing a suite of 3D head scans for dynamic photorealistic labelled data generation. We then briefly describe how we use image-based lighting [6] to model a wide range of realistic lighting conditions, and finally discuss the details of our rendering setup.

3.1. Eye model

Eyeballs are complex organs comprised of multiple layers of tissue, each with different reflectance properties and levels of transparency. Fortunately, as realistic eyes are so important for many areas of CG, there is already a large body of previous work on modelling and rendering eyes **Erroll: cite**.

As shown in Figure 1a, our eye model consists of two parts. The outer part (red wireframe) approximates the eye's overall shape with two spheres ($r_1 = 12$mm, $r_2 = 8$mm [7]),

ICCV #****

the latter representing the corneal bulge. To avoid a discontinuous seam between spheres, the meshes were joined and then smoothed. It is transparent, refractive ($n = 1.376$), and partially reflective. The eye's bumpy surface variation is modelled by a displacement map generated with noise functions. The inner part (blue wireframe) is a flattened sphere with Lambertian material. 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 outer and inner parts which accounts for the thickness of the cornea. **Erroll: compare with recent Disney work**

Eyes exhibit variations in both shape (pupillary dilation) and texture (iris color and scleral veins). To model shape variation we use *shape keys* – a CG animation technique where different versions of a mesh are stored, modified, and interpolated between [8]. **Erroll: more on shape keys** We have shape keys representing 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 appearance of the eye by compositing textures in three separate layers: *i*) a *sclera* layer representing the tint of the sclera (white, pink, or yellow); *ii*) an *iris* layer with four photo-textures of different colored irises (amber, blue, brown, grey); and *iii*) a *veins* layer which varies between blood-shot and clear . We matched the sclera tint to each separate face model, but uniformably randomly varied iris color. Previous research on iris-synthesis **Erroll: cite** would have allowed continually different iris textures, but we decided this added complexity would not make a worthwhile improvement in overall appearance variation, especially when rendered at lower resolutions.

3.2. Preparing a suite of 3D eye-region models

As can be seen in Figure 3a, the cornea has been incorrectly reconstructed in the head scan. This is because transparent surfaces are not directly visible, so cannot be reconstructed in the same way as diffuse surfaces like skin. Recent work uses a hybrid reconstruction method to reconstruct the corneal surface seprately, but requires additional hardware [9] – this level of detail was deemed unneccesary for our purposes. As we need full control of where the eye looks, we remove the original scanned eyeball from the mesh using boolean operations and place our own eyeball approximation in its place.

As can be seen in Figure 3a, the cornea has been incorrectly reconstructed in the head scan. This is because transparent surfaces are not directly visible, so cannot be reconstructed in the same way as diffuse surfaces like skin. Recent work uses a hybrid reconstruction method to reconstruct the corneal surface seprately, but requires additional hardware [9] – this level of detail was deemed unneccesary for our purposes. As we need full control of where the eye looks, we remove the original scanned eyeball from the mesh using boolean operations and place our own eyeball approximation in its place.

Preparing the head geometry

While the original head scan geometry is suitable for being rendered as a static model, its topology cannot easily represent dynamic changes in eye-region shape. Vertical saccades are always accompanied by eyelid motion [10], so we need to be able to pose the eyelids according to the gaze vector. When preparing a mesh for facial animation, edge loops should flow along and around the natural contours of facial muscles. This leads to a more efficient (lower-resolution) geometric representation of the face, and more realistic animation as mesh deformation mathes that of actual muscles.

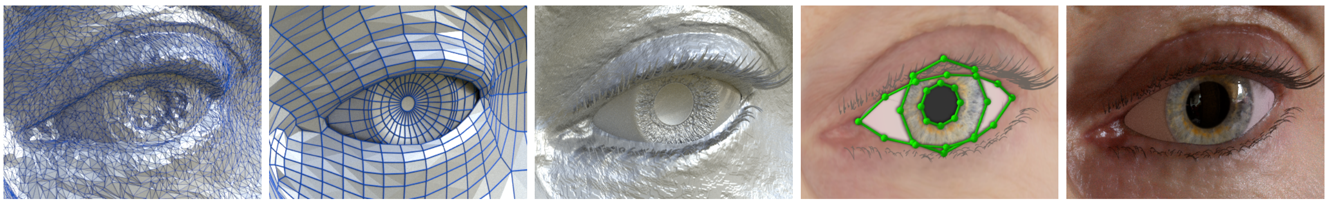
We therefore *retopologize* the face geometry into a more optimal form using a commercial semi-automatic system [11]. **Erroll: Reference some other options, e.g automatic methods in research** As can be seen in Figure 3b, edge loops now follow the *Orbicularis Oculi* muscle, allowing for realistic eye-region deformations. This retopologized low-poly mesh now lacks the detail of the original scan (e.g. the crease above the eye), and has visible sharp edges. We therefore use it as the control mesh for a displaced subdivision surface [12], with displacement map computed from the scanned geometry. As can be seen in Figure 3c, detail is restored.

Although they are two seperate 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 geometry of our eyeball model (Figure 3b). To compensate, the face mesh's eyelid vertices are displaced along their normals to their respective closest positions on the eyeball geometry (Figure 3c). This automatic operation ensures the models are joined, even after changes in pose [13].

Eyelashes

Eyelashes are short curved hairs that grow from the outer edges of the eyelids. These can occlude parts of the eye and affect eye-tracking algorithms, so it is important that we model them. We follow the approach of Świrski and Dodgson [1], and model them using directed hair particle effects. The hair particles are generated from a smoothed control surface below the eyelid edges and directed away from the face. To make them curl, the eyelash particles experience a slight amount of gravity during growth (negative gravity for the upper eyelash).

Figure 2: Our suite of female and male head models for rendering.



(a) Original 3D head scan data: 1.4 million polys (b) Retopologized head model: 9 thousand polys (c) Surface detail is stored in displacement maps (d) 3D iris and eyelid landmarks are annotated (e) The final render

Figure 3: Model preparation process

Figure 4: Eyelids.

Figure 5: Appearance variation from lighting is modelled with poseable high-dynamic-range environment maps [6].

3D Landmarks

Eyelid motion

Create face, eyelash, and landmark blend shapes for eyelids looking up and down.

3.3. Lighting

4. Experiments

4.1. Deformable model

- Evaluate eyelid landmark accuracy on LFW and M-PIE data, compare against several state-of-the-art CLM methods.

- Evaluate eyelid and iris landmarks on hand-annotated MPII data, compare against a baseline method: majority vote for iris position.

(Maybe) Plot landmark accuracy on LFW against number of training participants. Show that even with just a few participants (e.g. 4) we get good results for eyelid positions compared to state-of-the-art face trackers.

4.2. Gaze estimation

We render a targeted dataset that matches MPII's gaze and pose distribution, with added 3D laptop screen emitting light. This shows how we can target specific scenarios like laptop-based gaze estimation, and render a suitable dataset within a day rather than take 3 months of data collection.

Using Xucong's CNN sytem, we train on targeted version of SynthesEyes, test on MPII. Show results are better than training on UT and testing on MPII. This shows that the range of lighting in SynthesEyes is important for better results.

5. Conclusion

References

[1] L. Świrski and N. Dodgson, "Rendering synthetic ground truth images for eye tracker evaluation," in *Proceedings of the Symposium on Eye Tracking Research and Applications*. ACM, 2014, pp. 219–222. 1, 2

[2] G. Stratou, A. Ghosh, P. Debevec, and L. Morency, "Effect of illumination on automatic expression recognition: a novel 3d relightable facial database," in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. IEEE, 2011, pp. 611–618. 1

[3] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 617–624. 1

[4] J. Alabort-i Medina, B. Qu, and S. Zafeiriou, "Statistically learned deformable eye models," in *Computer Vision-ECCV 2014 Workshops*. Springer, 2014, pp. 285–295. 1

[5] C. Xiong, L. Huang, and C. Liu, "Gaze estimation based on 3d face structure and pupil centers," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 1156–1161. 1

[6] P. Debevec, "Image-based lighting," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 26–34, 2002. 1, 3

[7] 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 State-of-the-Art Report*, 2014, pp. 69–91. 1

[8] V. Orvalho, P. Bastos, F. Parke, B. Oliveira, and X. Alvarez, "A facial rigging survey," in *in Proc. of the 33rd Annual Conference of the European Association for Computer Graphics-Eurographics*, 2012, pp. 10–32. 2

[9] P. Bérard, D. Bradley, M. Nitti, T. Beeler, and M. Gross, "Highquality capture of eyes," *ACM Transactions on Graphics*, vol. 33, no. 6, p. 223, 2014. 2

[10] S. Liversedge, I. Gilchrist, and S. Everling, *The Oxford handbook of eye movements*. Oxford University Press, 2011. 2

[11] Pixologic, *ZBrush ZRemesher 2.0 Automatic retopology taken to a new level*, 2015, http://docs.pixologic.com/user-guide/3d-modeling/topology/zremesher/. 2

[12] A. Lee, H. Moreton, and H. Hoppe, "Displaced subdivision surfaces," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 85–94. 2

[13] The Blender Foundation, *The Shrinkwrap Modifier*, 2015, http://wiki.blender.org/index.php/Doc:2.6/Manual/Modifiers/Deform/Shrinkwrap. 2