EUROGRAPHICS 2019 / P. Alliez and F. Pellacini
(Guest Editors)

Volume 38 (2019), Number 2

Practical Person-Specific Eye Rigging

P. Bérard[1,2], D. Bradley[1], M. Gross[1,2], and T. Beeler[1]

[1]Disney Research [2]ETH Zurich

Figure 1: *We present a new person-specific eye rigging method based on accurate measurements from a multi-view imaging system.*

Abstract

We present a novel parametric eye rig for eye animation, including a new multi-view imaging system that can reconstruct eye poses at submillimeter accuracy to which we fit our new rig. This allows us to accurately estimate person-specific eyeball shape, rotation center, interocular distance, visual axis, and other rig parameters resulting in an animation-ready eye rig. We demonstrate the importance of several aspects of eye modeling that are often overlooked, for example that the visual axis is not identical to the optical axis, that it is important to model rotation about the optical axis, and that the rotation center of the eye should be measured accurately for each person. Since accurate rig fitting requires hand annotation of multi-view imagery for several eye gazes, we additionally propose a more user-friendly "lightweight" fitting approach, which leverages an average rig created from several pre-captured accurate rigs. Our lightweight rig fitting method allows for the estimation of eyeball shape and eyeball position given only a single pose with a known look-at point (e.g. looking into a camera) and few manual annotations.

1. Introduction

Eyes are amongst the most important facial features when it comes to creating believable digital characters. Humans have been primed by evolution to scrutinize the eye region, spending about 40% of our attention to that area when looking at a face [JWGD78]. One of the main reasons to do so is to estimate where others are looking in order to anticipate their actions. Once vital to survival, nowadays this is paramount for social interaction and hence it is important to faithfully model the way eyes move in digital characters.

When creating eye rigs, animators have traditionally thought of the eyeball as a sphere, which is rotated in place such that its optical axis points to where the character should be looking (Fig. 2 (a)). These assumptions are still predominant even in current state-of-the-art gaze estimation and manipulation research [WBM*16a, WXY16, WSXC16, WXLJH17, WBM*18]. The results reported by Bérard et al. [BBN*14], however, demonstrate that the eye shape is not a sphere, and is even asymmetric around the optical axis. But shape is only one aspect, and when building an animatable eye rig one must also carefully consider eye motion. While eye shape and motion have been long studied in ophthalmological communities, the wider field of computer graphics and vision have, for the most part, relied on simplified eye rigs and motion models. In this paper we investigate the creation and necessity of high-quality eye rigs in the context of computer vision and computer graphics applications. To be relevant for these fields, a phenomenon must be visible outside of ophthalmologic equipment, i.e. in imagery captured by ordinary cameras. Hence we employ a passive multi-view acquisition system to reconstruct high-quality eye poses over time, complete with accurate high-resolution eye geometry, and explore how the creation process can be reduced to even a single gaze input while still creating rigs of high quality.

A very important aspect that is not captured in naïve eye rigs is the fact that the gaze direction does not align with the optical axis of the eye but rather with its visual axis. The visual axis is the ray going through the center of the pupil starting from the fovea at the back of the eye, which is the location where the eye has the highest resolution. As depicted in Fig. 3 (b), the fovea is slightly shifted away from the nose, causing the visual axis to be tilted towards the nose (Fig. 2 (b)), on average around 6 degrees for adults [LE13, AA11]. This is an extremely important detail that cannot be neglected as otherwise the digital character will appear slightly cross-eyed, causing uncanny gazes.

In addition to the gaze direction, other phenomena of eye motion are often overlooked. For example, we demonstrate that torsion (i.e.

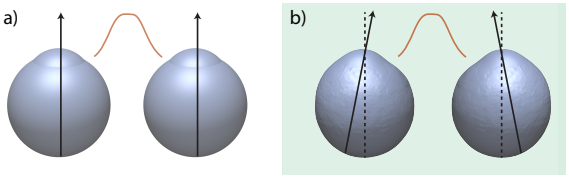

Figure 2: ***Eye Model:*** *a) Traditional eye models assume the eye to be roughly spherical and rotating around its center horizontally and vertically. The gaze direction is assumed to correspond to the optical axis of the eye (black arrows). b) The proposed eye model takes into account that the eye is not perfectly spherical and does exhibit rotation around all axes. Furthermore it respects the fact that the gaze direction is tilted towards the nose (see also Fig. 3 (b)).*

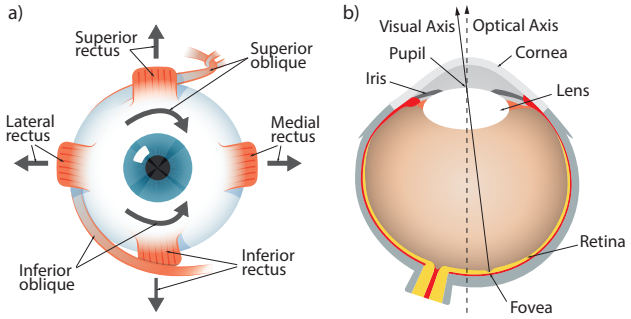

Figure 3: ***Anatomy:*** *a) The eye is controlled by six muscles (two per degree of freedom), which operate in an complex orchestrated way to rotate the eye. b) The gaze direction is not aligned with the optical axis of the eye (dashed line) but corresponds to the visual axis (solid line), which is formed by the ray passing through the center of the pupil originating from the fovea at the back of the eye, which is the area where the retina has the highest sensitivity.*

rotation about the optical axis) is clearly visible in the acquired data, as well as small amounts of eye translation during rotation due to the complex muscle system pulling on the eye (see Fig. 3 (a)). In this work we investigate the importance of modeling all these eye motion phenomena in the context of computer graphics applications. Our results indicate that an accurate gaze direction is crucial, along with properly computing the rotation center of the eye and modeling the torsion during rotation, while small translations during rotation can be neglected with essentially imperceptible visual consequences.

2. Related Work

Our work is related to eye tracking and gaze estimation in images, capturing and modeling 3D eye geometry and appearance, and rigging and animating eyes for virtual characters. In the following we will discuss related work in each area.

2.1. Eye Tracking and Gaze Estimation

The first methods for photographic eye tracking date back over 100 years [DC01, JMS05], and since then dozens of tracking techniques have emerged, including the introduction of head-mounted eye trackers [HT48, MT62]. We refer to detailed surveys on historical and more modern eye recording devices [Col99, Egg07]. Such devices have been widely utilized in human-computer interaction applications. Some examples were to study the usability of new interfaces [BOH91], to use gaze as a means to reduce rendering costs [LW90], or as a direct input pointing device [ZMI99]. These types of eye trackers typically involve specialized hardware and dedicated calibration procedures.

Nowadays, people are interested in computing 3D gaze from images in the wild. Gaze estimation is a fairly mature field (see [HJ10] for a survey), but a recent trend is to employ appearance-based gaze estimators. Popular among these approaches are machine learning techniques that attempt to learn eye position and gaze from a single image given a large amount of labeled training data [SMS14, ZSFB15], which can be created synthetically through realistic rendering [WBZ*15, WBM*16b]. Another approach is model-fitting, for example Wood et al. [WBM*16a, WBM*18] create a parametric eyeball model and a 3D morphable model of the eye region and then fit the models to images using analysis-by-synthesis. Other authors propose real-time 3D eye capture methods that couple eye gaze estimation with facial performance capture from video input [WSXC16] or from RGBD camera input [WXY16] including an extension to eyelids [WXLJH17]. However, these techniques use rather simple eye rigs and do not consider ophthalmological studies for modeling the true configuration of eyes, which is the focus of our work, and we believe that these methods could benefit from incorporating the knowledge compiled in this work.

2.2. Capturing and Modeling Eyes

Capturing and modeling eye geometry and appearance has also been a topic of interest in the computer graphics community. The eye consists of several different components with different appearance properties, including the semi-opaque sclera with veins, the transparent cornea, and the colored fibrous iris (see Fig. 3 (b)). Different approaches are sometimes used for different components, for example François et al. [FGBB09] synthesize the iris geometry from an input photograph using a dark-is-deep approach, and Lefohn et al. [LBS*03] take an oculariste's approach to create irises. Sagar et al. [SBMH94] propose a procedural eye creation model for surgery simulation. Recently Bérard et al. [BBN*14] presented a high resolution eye scanning method for all the visible parts of the eye, based on a complex multi-view acquisition setup. While tedious to utilize, the acquired data helped inform artists about realistic eye shapes, including eyeball asymmetry and the color-structure coupling of irises. Furthermore, Bérard et al. [BBGB16] showed how to build a high-quality parametric eye shape model from the captured high-resolution scans and how to fit the model to single eye images. These methods focus on modeling static shape and appearance of eyes, which is complementary to the topic of our work - rigging eyes for realistic animation.

2.3. Eye Rigging and Animation

Eye animation is of central importance for the creation of realistic virtual characters, and many researchers have studied this topic [RAB*14]. On the one hand, some of the research explores the coupling of eye animation and head motion [PRMG16, MD09] or speech [ZFI11, LMD12, MXL*13], where other work focuses

on gaze patterns [CKB01, VSvdVN01], statistical movement models for saccades [LBB02], or synthesizing new eye motion from examples [DLN05]. These studies focus on properties like saccade direction, duration, and velocity, and do not consider the 3D rigging and animation required to perform the saccades. When it comes to rigging eye animations, simplifications are often made, as mentioned earlier, for example modeling eyes as a rotating sphere with no distinction between visual and optical axis [IDP03, PM09, WLO10, WBM*16a, PRMG16] (Fig. 2 (a)). While easy to construct and animate, this simple eye rig is not anatomically accurate and, as we will show, can lead to uncanny eye gazes. In this work, we show that several of the basic assumptions of 3D eye rigging do not hold when fitting eyes to imagery of real humans, and we demonstrate that incorporating the knowledge from the field of ophthalmology can improve the realism of eye animation in computer graphics.

3. Overview

In this paper we present a novel parametric eye rig and two methods (high-quality and lightweight) to estimate its person-specific parameters from images. We define the eye rig in Section 4. In Section 5 we describe the image capture setup that we need for estimating the rig parameters. The high-quality parameter estimation has two phases. First, for a pair of eyes we fit the eyeball shape and position for a number of poses given annotated multi-view data (Section 6) and second we fit the actual rig given the reconstructed poses (Section 7.1). Based on these high-quality rigs we compute a data driven parametric rig prior that allows to estimate rig parameters from just a single pose requiring only a few manual annotations (Section 7.2).

4. Eye Rig

Our eye rig consists of several parameters that define the rig configuration. We differentiate between *fixed* and *variable* parameters, where fixed parameters are person-specific but do not change during animation, and variable parameters can change over time. The fixed configuration describes the geometry of the rig, such as, for example the interocular distance or the shape of the eyeballs. We attribute the fixed parameters with a bar ($\bar{x}$). The variable configuration defines the motion of the eyes, and we attribute variable parameters with a hat ($\hat{x}$). The entire configuration containing both fixed and variable parameters is denoted as $\hat{\bar{\mathcal{P}}}$.

In the following we describe the individual rig parameters. Without loss of generality, we will consistently refer to a right-handed coordinate system where the x-axis points left, the y-axis points up, and the z-axis points forward, all with respect to the character.

4.1. Eye Shape

Fig. 3 (b) shows a cross-section of the eye and labels the most important features in our context, which we will discuss in more detail below.

Eyeball shape For the eyeball shape we use the parametric eye model provided by Bérard et al. [BBGB16]. This model represents the eyeball shape with a PCA model with six modes plus a global scale. Since the two eyes of an individual are similar in shape, we employ a set of six symmetric coefficients coupled with a set of six antisymmetric coefficients that model the difference and are regularized to be small. It will become convenient to model certain parameters as splines on the eyeball surface (e.g. the limbus, as described next). In order to allow for efficient evaluation of splines on the eyeball surface, we transition from the irregular mesh domain to the regular image domain and store the mean shape and difference vectors as texture maps. The texture parameterization is based on spherical coordinates and chosen such that the poles are on the top and bottom of the eye, and the texture resolution is 2048x1024 pixels. Given the rig configuration $\hat{\bar{\mathcal{P}}}$, any point $\boldsymbol{x}_{uv} \in \mathbb{R}^2$ in texture space can be transformed to a point $\boldsymbol{x}_{world} \in \mathbb{R}^3$ in world space via

$$\boldsymbol{x}_{world} = Eyeball(\boldsymbol{x}_{uv}, \hat{\bar{\mathcal{P}}}), \tag{1}$$

which applies the inverse texture parameterization at $\boldsymbol{x}_{uv}$ followed by a forward evaluation of the rig configuration $\hat{\bar{\mathcal{P}}}$.

Limbus The limbus refers to the boundary between the cornea and sclera. Its shape and position is tightly coupled with the shape of the eyeball and has no additional degrees of freedom. We represent the limbus in texture space as a closed B-spline that is directly mapped to the eyeball surface. We define the mapping of points $x_{ctr} \in \mathbb{R}^1$ on the spline to points $\boldsymbol{x}_{uv} \in \mathbb{R}^2$ in texture space as

$$\boldsymbol{x}_{uv} = Limbus(x_{ctr}). \tag{2}$$

Pupil The parametric eye of Bérard et al. [BBGB16] also contains a pupil. However, it is the mean pupil of a captured dataset and does not account for any person-dependent excentricity of the pupil. To address this we add three translation parameters that are fixed and common to both eyes, which describe the offset from the mean pupil. Analogous to the eyeball shape coefficients we control the radius of the two pupils via a symmetric parameter and an antisymmetric one that accounts for the fact that the two pupils will be similar in radius but not exactly the same. The pupil radius parameters change per pose and are thus variable.

Visual axis The visual axis of the eye is defined as the ray passing through the center of the pupil and originating at the point on the retina with the sharpest vision, the fovea. Since we do not know the location of the fovea, we model the visual axis by a ray originating at the center of the pupil. The direction of the ray is defined in spherical coordinates, as the inclination relative to the z-axis. The pair of visual axes for the two eyes is given by four fixed parameters, a symmetric polar angle and antisymmetric azimuth that provide the main directions, coupled with an antisymmetric polar and symmetric azimuth that model slight deviations between the left and right eyes.

Eyelid interface The eyelid interface defines the location where the skin of the eyelid touches the eyeball. We extend the parametric eye model of Bérard et al. with a parametric model of the eyelid interface. Similar to the limbus, this interface is represented by curves in texture space, one for the upper and one for the lower eyelid interface. The shape of the curves is based on two fourth order B-splines whose six middle control points are constrained as shown in Fig. 4. The control points are constrained to lie on equidistant lines perpendicular to the horizontal line connecting

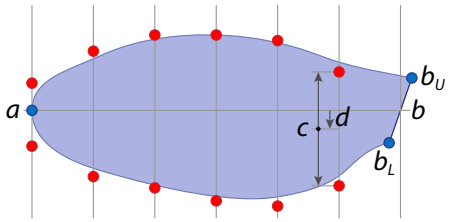

Figure 4: *The eyelid interface consists of two B-spline curves (from a to b_U and a to b_L) defined by their control points (red and blue). The blue control points can move freely. The middle control points (red) are equally distributed on the middle line connecting the eye corner (a) and the tear duct (b) and are constrained to move perpendicularly to this middle line. The two control points on each of these lines are parameterized by the eye opening distance (c) and their joint vertical shift from the middle line (d).*

the two corners of the eye. Each perpendicular line contains two control points that are parametrized by the opening of the eyelid (computed as the signed distance between the two points) and the vertical offset of the points (parameterized by the signed distance between their mean and the horizontal line). The opening parameter is constrained to positive values which prevents the upper curve from crossing over the lower curve. The eye rotation relative to the eyelid interface is accounted for by warping the eyelid curves in texture space. Since the texture coordinates are based on spherical coordinates, the warp can be computed analytically. Given the rig configuration $\hat{\bar{\mathcal{P}}}$, we define the mapping of points $x_{ctr} \in \mathbb{R}^1$ on the spline to points $\boldsymbol{x}_{uv} \in \mathbb{R}^2$ in texture space as

$$\boldsymbol{x}_{uv} = Eyelid(x_{ctr}, \hat{\bar{\mathcal{P}}}). \tag{3}$$

Tear duct We model the tear duct as a line segment between the last point on the upper eyelid interface curve and the last point on the lower eyelid interface curve.

4.2. Eye Motion

As depicted in Fig. 3 (a), the eye is driven by a set of muscles that exert translational forces on the eyeball in order to rotate it. Two muscles are responsible for one rotational degree of freedom (one for each direction), but for any actual motion there is always several of these muscles being activated in a complex and orchestrated way. An in-depth discussion of the muscular eye actuation is beyond the scope of this paper and we refer the interested reader to medical textbooks [Car88]. To name just one example, when the eye is rotated horizontally away from the nose *(abducted)*, most of the work to rotate the eye upwards *(elevation)* will be done by the *superior rectus* muscle. On the other hand, when the eye is rotated horizontally towards the nose *(adducted)*, it will be the *inferior oblique* muscle that is responsible for elevating the eye. As a consequence, the typical assumption that the eye rotates only horizontally and vertically around a fixed pivot is only an approximation. In reality the eye not only exhibits rotation around all axes, but also translates within its socket during rotation [FH62]. In the following we discuss how rotation and translation is handled within our rig.

Rotation We model the eye rotation $\hat{\boldsymbol{\Theta}}$ based on a Helmholz gimbal with three degrees of freedom (up/down=$\hat{\Theta}_x$, right/left=$\hat{\Theta}_y$, torsion=$\hat{\Theta}_z$). According to Donders' law, for a given gaze direction $\left(\hat{\Theta}_x, \hat{\Theta}_y\right)$ the torsion angle $\hat{\Theta}_z$ is unique and independent of how the eye reached that gaze direction. To determine the corresponding z-axis rotation for a given gaze direction we apply Listing's law following the work of Van Run et al. [VRVdB93]. Listing's law states that all feasible eye orientations are reached by starting from a single reference gaze direction and then rotating about an axis that lies within the plane orthogonal to this gaze direction. This plane is known as the Listing's plane, which we parameterize by ($\bar{\Theta}_x$,$\bar{\Theta}_y$)

$$\hat{\Theta}_z = \mathcal{L}\left(\hat{\Theta}_x - \bar{\Theta}_x, \hat{\Theta}_y - \bar{\Theta}_y\right). \tag{4}$$

Translation While Listing's model is well understood in ophthalmology, only very little is known about the translation of the rotation center. Fry and Hill [FH62, FH63] reported that the rotation center of the eye is not a single point, but that it lies on a fixed arc called the centrode. For the left-right motion of the eye, they report that the rotational center of the eye orbits around the center of its socket at an average distance of 0.79mm. For the up-down motion they report an inverted orbit, i.e. the eye moves forward when rotating up and down. Their measurements were limited to central left-right and up-down motions. Our measurements exhibit translations of the eyes in the same sub-millimiter range (see Fig. 12). Since these translations are extremely small and barely perceptible even in our close-up data, we have concluded that inaccuracies from ignoring eye translations during rotation are negligible for computer graphics applications, and employ a person-specific rotational pivot $\bar{\boldsymbol{p}}$ that is pose-independent.

4.3. Eye Positioning

The eyes are positioned inside the head via a series of transformations. The most direct way would be to place each eye independently in the world coordinate frame, but this would require two full rigid transformations per frame, and hence be highly overdetermined. The aim is thus to reduce the degrees of freedom as much as possible without sacrificing the required flexibility. For an overview of the chosen coordinate frames please refer to Fig. 5.

World → Skull A first step is to model the head motion. This will require one rigid transformation per frame $\hat{M}_{world \to skull}$, which can be given by animation curves or estimated from captured data (e.g. [BB14]).

Skull → Pair Relative to the skull we create an eye pair coordinate frame, defined via the reduced rigid transformation $\bar{M}_{skull \to pair}$. This coordinate frame is chosen such that its origin is in the middle between the left and right eyes, with the x-axis going through their rotational pivots $\bar{\boldsymbol{p}}$, and the x-axis rotation is kept identical with the x-axis rotation of the skull. The pair coordinate frame is person-specific but fixed as it does not change during animation.

Pair → Socket The left and right eye sockets are defined relative to the eye pair coordinate frame via a fixed transform $\bar{M}_{pair \to socket}$. The sockets are translated by plus/minus half the interocular distance along the x-axis and plus/minus half the vertical eye offset along the y-axis.

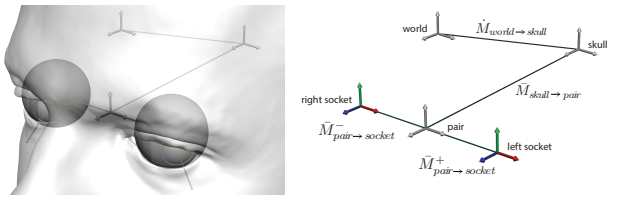

Figure 5: *The proposed rig rotates and offsets the eye relative to its socket. The left and right sockets are defined via antisymmetric transformations relative to the joint pair coordinate frame, which in turn is relative to the coordinate frame of the skull. While all of these transformations are fixed, the skull moves relative to the world coordinate frame over time.*

World → Socket The ultimate socket transformation per eye is given by the concatenation of the individual transformations. The total number of degrees of freedom is $6n$ (World → Skull) + 5 (Skull → Pair) + 2 (Pair → Socket) = $6n+7$, where n is the number of frames, versus the $12n$ of the most naïve model.

4.4. Eye Control

Once fit to a person (Section 7.1), the proposed rig exposes the eye gazes as control parameters for the eye pose. Consistent with industry grade eye rigs, an animator may animate the eye gazes of the left and right eyes individually, or couple them via a controllable look-at point of the character. In the former case the rig exposes four degrees of freedom (one 2D gaze per eye), which are reduced to three in the latter case (one 3D look-at point). Furthermore, the opening of the pupil can be controlled by a single user parameter.

5. Data Acquisition

In order to develop our eye rig we depend on high-quality data of real eye motion. We employ a multi-view capture setup consisting of 12 DSLR cameras (Canon 1200D) for taking photographs of static eye poses, from which we can reconstruct the shape of the skin surface using the system proposed by Beeler et al. [BBB*10]. In the following, we describe all the data capture required for the high-quality rig fitting. The lightweight rig estimation requires only a small subset of the data which we describe later.

For a given subject, we record approximately 60 different eye positions, corresponding to one set of gaze points approximately 1 meter from the subject, which span three horizontal rows at various heights, as well as a second set of gaze points that increase in distance from the subject along a single viewing ray, in the range of 0.25 to 3 meters. For the entire capture session the subject maintains a fixed head position. As a result, there is only little motion between frames and we can track a face mesh template to all frames [BHB*11] and compute the underlying skull pose using a rigid stabilization technique [BB14]. Our setup is shown in Fig. 6.

We further record the 3D look-at point for each pose using an HTC Vive tracking system[†]. We modified one of the Vive controllers by adding a small light bulb, which the subject is instructed to fixate on during acquisition. For calibrating the Vive to the camera coordinate frame we record a series of points and triangulate the light position from the camera views. To add robustness outside the working volume of the cameras, we also record the position of the cameras with the tracked controller by measuring a point at the back of each camera.

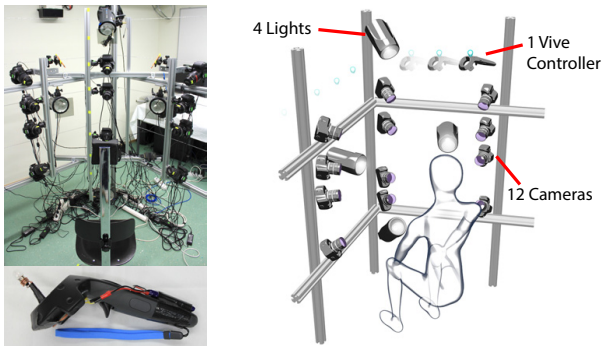

Figure 6: *Our capture setup consists of 12 DSLR cameras and 4 industrial light flashes, providing synchronized multi-view imagery of static eye poses. We modified an HTC Vive Controller by adding a small light bulb, which the subject fixates on during acquisition, giving ground truth 3D look-at points.*

The final result of our data acquisition stage is a multi-view image dataset of approximately 60 eye poses, complete with facial geometry that has known rigid head transformations between poses, and known 3D look-at points. We captured and evaluated our method on seven different subjects.

6. Eye Configuration Reconstruction

One of the core components of this work is to empirically design an eye rig that is capable of faithfully representing real eye motions while being compact and robust to noise. We aim to construct a person-specific rig from the captured data described in Section 5. Thus far, however, the dataset contains only reconstructed face meshes and skull transformations, but no per frame eyeball geometry to fit the rig to. In this section we describe how we obtain the eye configurations (shape and per frame pose) for the captured data. Once we have accurately reconstructed the eye configurations, we fit the person-specific eye rig parameters as described in Section 7.1.

We wish to reconstruct eye configurations with as little regularization as possible in order to remain faithful to the data. For the shape, fortunately we can rely on the parametric eye model of Bérard et al. [BBGB16], which was itself generated from measured data [BBN*14]. This alleviates the problem considerably and leaves us only with the need to recover the six degrees of freedom of the eye pose, which we denote $\hat{M} \in \mathbb{R}^6$, for each pose of each eye. As with most applications of parametric model fitting to real world data, our rig will only explain the captured imagery up to a certain error. In order to improve the fit we introduce two slack variables in the eye pose computation. First, we add a per pose torsion residual $\hat{\Theta}^{\varepsilon}_z$ to Eq. 4, yielding

$$\hat{\Theta}_z = \mathcal{L}\left(\hat{\Theta}_x - \bar{\Theta}_x, \hat{\Theta}_y - \bar{\Theta}_y\right) + \hat{\Theta}^{\varepsilon}_z. \tag{5}$$

[†] www.vive.com

Secondly, we add a per-pose residual $\hat{\boldsymbol{p}}^{\varepsilon}$ for the rotational pivot point $\bar{\boldsymbol{p}}$, yielding

$$\hat{\boldsymbol{p}} = \bar{\boldsymbol{p}} + \hat{\boldsymbol{p}}^{\varepsilon}. \tag{6}$$

Together with the gaze direction $(\hat{\Theta}_x, \hat{\Theta}_y)$, this amounts to six dynamic degrees of freedom per eye and allows us to accurately reconstruct eye poses.

We obtain the eye configurations in two stages. First we fit to manual annotations, which makes fitting very robust, since automatic labeling is challenging due to the complex geometry and appearance in the eye region. Manual annotations, however, are not pixel-perfect and therefore the fits contain errors. Thus, we refine the positions with photometric constraints in a second stage. For implementation details on the two fitting stages we refer to Appendix A.

The algorithm presented in this section allows us to reconstruct eye poses from multi-view imagery at submillimeter precision. Fig. 7 shows the measurements for one of the test subjects. We captured the person doing three horizontal sweeps followed by a single vertical sweep from neutral gaze upwards. The look-at points were distributed on the capture gantry (Fig. 6) and as a consequence the elevation of the eye changes during the horizontal sweeps. The gaze directions are clearly visible and while the vertical gaze is the same for both eyes, the horizontal gaze differs by a constant offset, due to the discrepancy between the optical and visual axis (Fig. 3 (b)).

The reconstruction accuracy is demonstrated in Fig. 8 by overlaying the fitted eyeball, limbus, eyelid interface, and tear duct on top of the input views. By modeling all the components of the eye region our method can robustly handle occlusions by the eyelashes and the eyelids. Given a fitted pose we can furthermore compute an eyeball texture from any view. If the fitted eyeballs have the correct shape and position these textures should align and be identical up to lighting differences. Fig. 9 shows how these textures align nicely by interleaving textures computed from different views. The eyeball is coated by a protective, mostly transparent tissue layer called the conjunctiva, which is not firmly attached to the eyeball but slides over it during rotation. As a consequence, the veins in the conjunctiva deform relative to the sclera, which complicates alignment of eye poses considerably. By computing textures from the same view but for different frames we can visualize the stabilized sclera in texture space and the sliding conjunctiva becomes apparent as shown in Fig. 10.

In the next section we describe how eye rigs may be fitted to the computed per frame eye poses and in Section 7.2 we describe how to estimate a rig from a single frame with limited annotations only.

7. Rig Fitting

Our fitting framework can fit an eye rig to a spectrum of poses, from a single pose to dozens of poses. The more poses available, the more parameters we can estimate and the more accurate the estimations become. In this section, we show the extreme cases of high-quality rig fitting to about 50 poses and lightweight rig fitting to a single pose only. In the lightweight case we leverage an average rig (computed from 7 high-quality fit rigs) for the parameters that cannot be estimated like the rotation center and the visual axis.

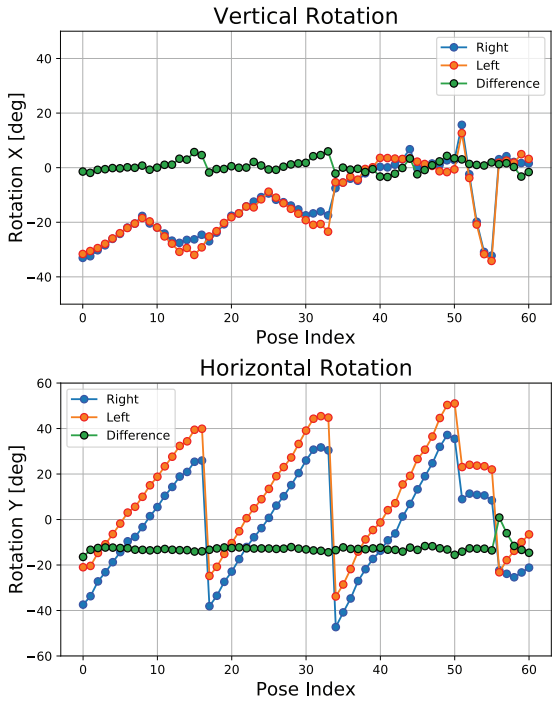

Figure 7: *This figure shows the raw measurements of gaze angles for one subject, for both the left (orange) and right (blue) eyes. The subject did three sweeps left to right at different eye elevations (frames 0-15, 16-33, and 34-50) and finally a vertical sweep from neutral upwards (51-55). For a single horizontal sweep, the vertical rotation first increases and then decreases, instead of staying constant. This is due to the look-at points being distributed on the capture gantry running over a corner.*

7.1. High-Quality Rig Fitting

In the previous section we introduced residual variables that add additional degrees of freedom to the rig and allow us to accurately reconstruct the eyeball poses for all frames independently. Unfortunately, we cannot interpolate these poses without a model. In this section we show how the individual components of the proposed rig can be fit to the reconstructed per frame eye configurations to create a model that faithfully reproduces human eye motion.

To determine the optimal rig parameters based on the per frame eye configurations we uniformly sample the eyeball to produce a set of texture coordinates for which we have corresponding 3D positions in each frame. Using all these positions as constraints we solve for the optimal rig parameters while enforcing the residual variables from the previous section to be zero. In addition to the per frame eye configuration reconstructed in the previous section we also record the look-at point for every frame (Section 5), which allows us to compute the visual axis per eye. It is well known from ophthalmology that the visual and optical axes of the human eye are shifted by approximately 6 degrees [AA11], and our measurements confirm this as shown in Fig. 17.

Due to inaccuracies in eye configuration reconstruction and facial stabilization our rig does not predict the eye and face positions

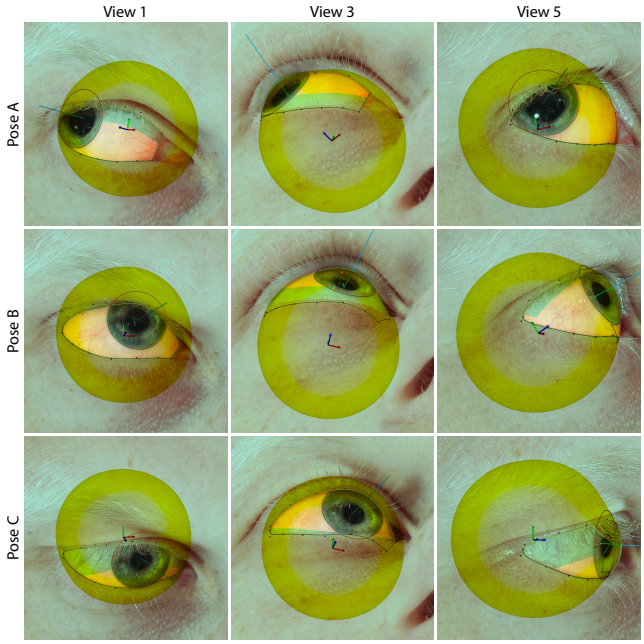

Figure 8: *The reconstruction accuracy is demonstrated by overlaying the limbus (red), the eyelid interface (green), and the visual axis (blue) on top of several input views (1,3,5). The eyeball geometry is shown in yellow.*

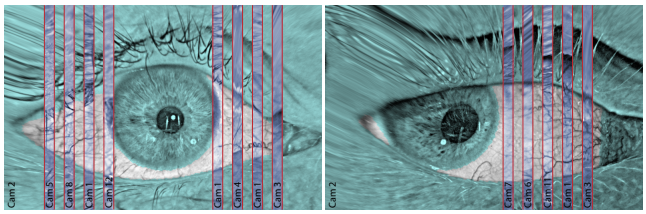

Figure 9: *Here accuracy is demonstrated by computing eyeball textures from different views. Texture slices from different views (blue) are overlaid on a reference texture (gray/turquoise). Accurate reconstruction leads to an alignment of the texture veins in the gray area (the turquoise skin and iris areas should not be compared).*

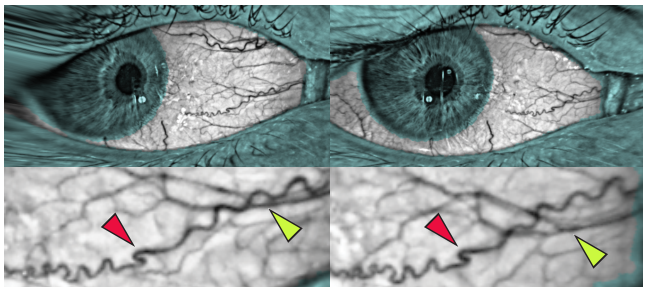

Figure 10: *The eyeball (shown here in texture space, with the sclera masked) is coated by a protective, mostly transparent tissue layer called the conjunctiva, which is not firmly attached to the eyeball but slides over it during rotation. As a consequence, the veins in the conjunctiva (green arrow) deform relative to the sclera (red arrows). This complicates alignment of eye poses considerably.*

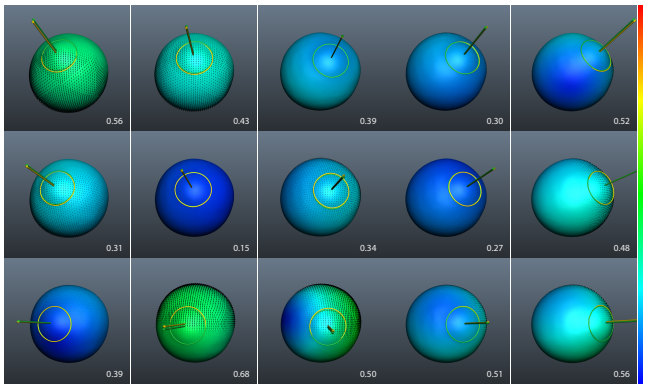

Figure 11: *The goal of the fitted rig (green curves) is to perfectly predict the reconstructed eyeball positions (yellow curves). The heat map encodes the difference between reconstructed positions and the positions predicted by the rig. The average vertex error in millimeters is indicated for each pose in white. The black arrows show the correspondences. The scale bar goes from zero (blue) to two millimeters (red).*

perfectly. Fig. 11 compares the reconstructed eye configurations and the eye poses predicted by the rig. The errors are measured between corresponding points and include the torsion of the eyeballs. All in all the error does not exceed one millimeter, which is in line with the residual variables shown in Fig. 12.

Translations After fitting we can relax the system by adding back the residual variables which allows us to match the per frame eye configurations perfectly. Fig. 12 shows these residual variables and one can see that they are within the range of +/- one millimeter. These offsets are small considering that the poses cover the full range of motion of the eyes. And since the range of the residual variables correspond to roughly the accuracy of our facial stabilization system we cannot reason about the patterns that we observe in these plots.

Listing's Model Listing's model predicts the per frame torsion $\hat{\Theta}_z$ based on the eye gaze $(\hat{\Theta}_x,\hat{\Theta}_y)$. Key to the Listing's model is the orientation of the Listing's plane $(\bar{\Theta}_x,\bar{\Theta}_y)$ which we fit based on the measured per frame orientations. As shown in Fig. 13 the model predicts the torsion well in the central field of view but degrades with more extreme gazes, where the physiology of the eye motion appears to disagree with the theoretical model. Many traditional eye models neglect the rotation around the z-axis (torsion) and use a simple Helmholtz gimbal. This results in a mismatch of up to 15 degrees as shown in Fig. 14. Using the torsion predicted by the Listing's model alleviates the mismatch considerably.

Gaze The fact that these measurements have been computed from ordinary cameras is a strong indicator that phenomena such as torsion and eye positioning can be important for computer vision applications, such as accurate eye gaze estimation. Eye gaze is also central for computer animation, where incorrect gaze can lead to cross-eyed characters (Fig. 15 and Fig. 21). As the visual axis is rotated towards the nose, the optical axis is actually pointing outwards when a person is looking at infinity. Given this it is even more important to model the eyeball torsion properly as torsion rotates the visual axis and contributes also to cross-eyed characters.

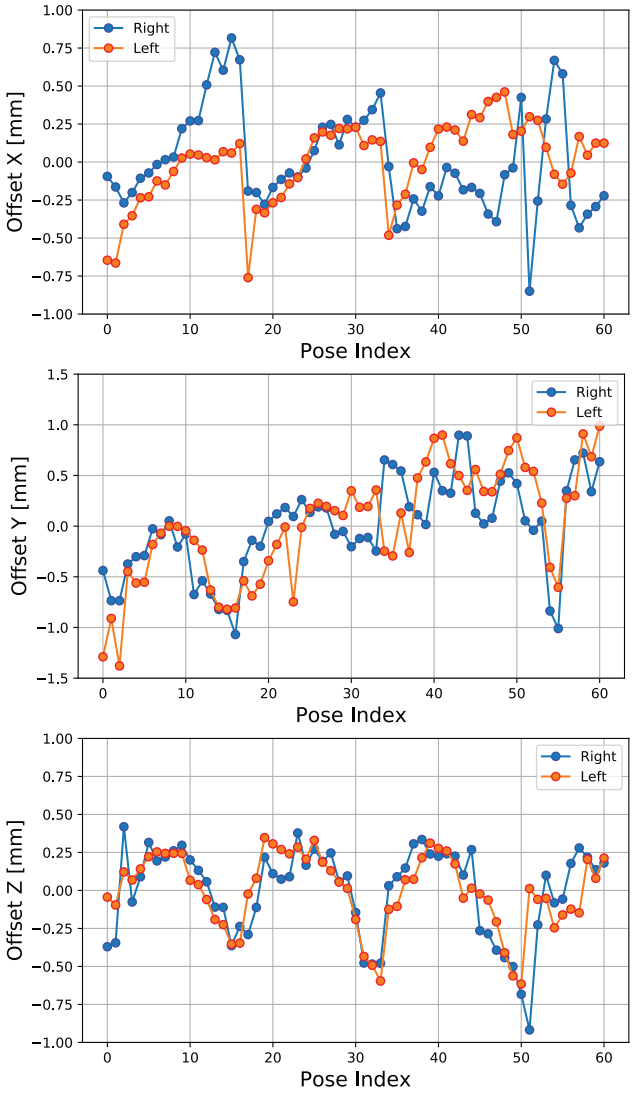

Figure 12: *The plots show the residual socket offsets of one subject. Their amplitude is within +/- one millimeter, which is within the error range of the used facial stabilization system.*

7.2. Lightweight Rig Fitting

The fitting method from Section 7.1 produces a high-fidelity and person-specific eye rig. It requires, however, the acquisition of dozens of eye poses including look-at points. For some applications this is not feasible and some trade-offs with quality might be acceptable. We thus propose a lightweight fitting method that leverages a small collection of high-quality rigs to compute an average rig that makes reasonable assumptions about parameters that cannot be estimated from the available data.

Average Rig We compute an average rig from seven subjects for which a high-quality rig has been computed as described in Section 7.1. Simple averaging of the rotation center parameter $\bar{\boldsymbol{p}}$ and the visual axis parameters leads to the desired average rig. For the z-component of the rotation center $\bar{\boldsymbol{p}}_z$ we observe a dependency on

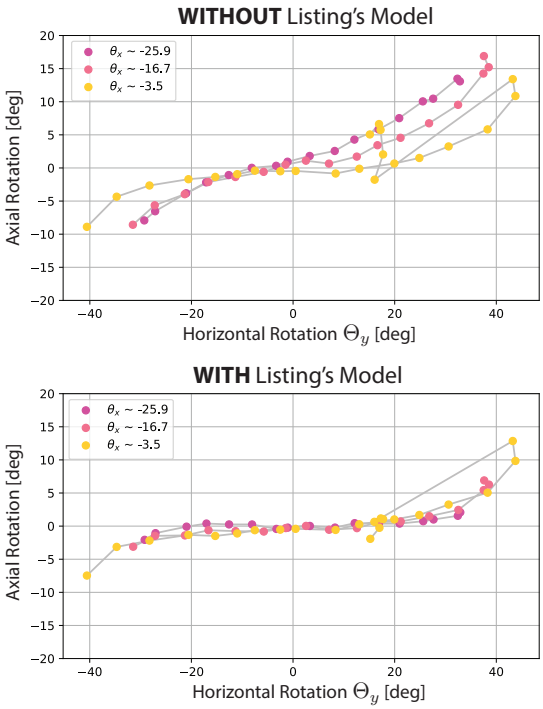

Figure 13: *Not predicting rotation around the optical axis amounts in large residuals across the entire range of motion. Listing's model predicts the torsion reliably for the largest part but fails to explain the extremes where it appears to deviate from the true physiology of the eye. Note that the model correctly predicts the dependency on elevation of the eye (Θ_x).*

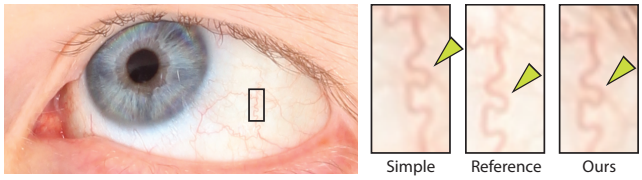

Figure 14: *Many traditional eye models neglect the rotation around the z-axis (torsion). This results in a mismatch of up to 15 degrees, which is visible in the close-up on the right, where the vein position predicted by the simple model differs from the true position. Using the torsion predicted by the Listing's model alleviates the mismatch. While visually subtle, torsion strongly influences the gaze since it rotates the visual axis around the optical axis of the eye.*

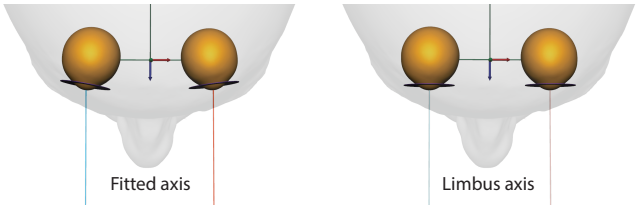

Figure 15: *The visual axis is tilted towards the nose and is not perpendicular to the limbus. This results in the limbus planes (violet) being oriented away from the nose if the subject's gaze is at infinity.*

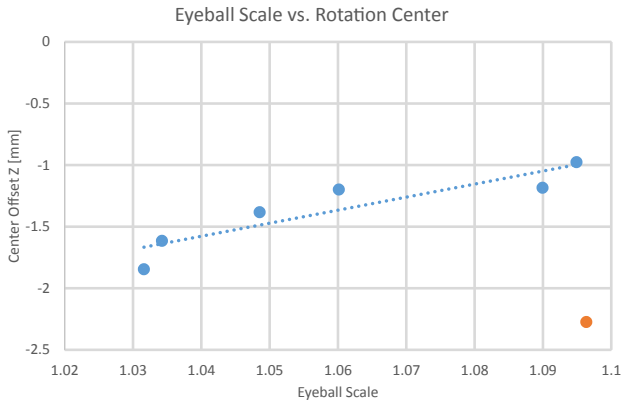

Figure 16: *This figure shows the dependence of the rotation center on the eyeball size. The horizontal axis shows the eyeball scale and the vertical axis shows the offset from the canonical rotation center along the z-direction. The linear regression models the blue data points. The orange outlier is excluded from the regression.*

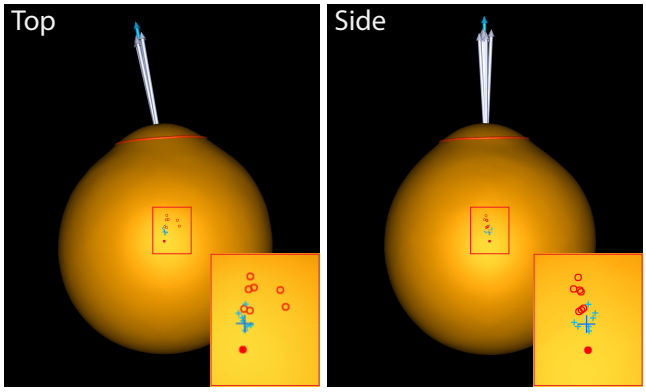

Figure 17: *The average eye rig is computed by averaging seven fitted eye rigs. The figure shows the average visual axis (blue) and the average rotation center (big blue cross). For all seven subjects we also visualize the visual axis (gray), the rotation center (small blue cross), the center of gravity (filled red circle), and the center of a sphere fitted to the visible part of the sclera (red circle).*

the eyeball scale. We model this dependency with a linear regression that predicts $\bar{\boldsymbol{p}}_z$ based on the current scale as shown in Fig. 16. From our seven subjects one does not seem to follow the same trend and we treat it as an outlier and exclude it from our model.

For the rotation center, an artist might be tempted to use the center of mass of the eye model or a fitted sphere. As shown in Fig. 17 we observe that the center of mass of our model is further back (-1.27 mm) and a fitted sphere with radius 12.5mm is closer towards the front (1.33 mm) compared to the average rotation center, which leads to substantial errors when the eye is rotating as shown in Fig. 19.

Fitting Given only a single pose we need to fit ten degrees of freedom. Six degrees of freedom for the $\bar{M}_{skull \to socket}$ transformation and two degrees of freedom for the orientation of each eye. To fit

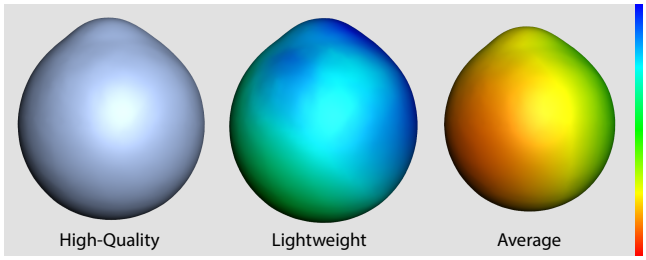

Figure 18: *If multiple eyeball poses are not available to estimate a high-quality eyeball shape, a lightweight estimate can be made given eyelid interface and eye corner annotations for a single pose. The average eyeball shape is shown as a comparison. The heat map encodes the distance between the high-quality and the estimated eyeball shape. The scalebar goes from 0 (blue) to 2 (red) millimeters.*

the position of the eyeball we use the most salient eyeball feature: the limbus. We annotate the limbus in three different views, which robustly constrains the limbus position in space. This also defines the orientation of the eyeball, but not very accurately.

By choosing the pose such that the subject is looking into the camera or is looking at a known calibrated point we can use a visual axis constraint that leverages the average visual axis and strongly constrains the orientation of the eyeball. This constraint minimizes the distance between a look-at point (e.g. the camera) and the average visual axis. Together with the limbus constraint and the average rotation center we can accurately fit the eyeball position.

Unfortunately, the eyeball shape cannot be estimated from these annotations. If we desire better eyeball shapes, we can further annotate the upper and lower eyelid interfaces and eye corners in two to three views. Since the shape is only partially defined by the given eyelid interface annotations we add an eyeball shape regularization constraint. With these constraints we fit the rig including the shape and scale of the eyeball as shown in Fig. 18.

8. Results

To evaluate the accuracy of our lightweight fitting approach we compare the fitting result to our multi-pose rig as shown in Fig. 19. More specifically, we compare the predicted eye positions of the fitted pose, and of three retargeted poses. Furthermore, we compare three different lightweight rigs among which two have suboptimal parameters, which might be the case when placing the eyeballs manually in the face.

The reference is the high-quality rig (Section 7.1) fitted to the reconstructed eye configurations (Section 6). Given the reference rig we triangulate a look-at point that we use to retarget the lightweight rigs, i.e. rotate the eyes such that their visual axes intersect at this look-at point. This reference rig is compared to our lightweight plus two modified lightweight rigs. One of the modified rigs has an eyeball center shifted by three millimeters to the left and the other modified rig has a visual axis offset by five degrees. These are common scenarios when an artist defines the rotation center and visual axis manually. With these erroneous parameters, one single pose may fit well, however when the rig is retargeted to another

pose, the position and/or the gaze will be incorrect. A bad rotation center for example does not affect the fitted pose as there are enough degrees of freedom to position the eyes for the single pose. However, in the retargeted poses it will yield incorrect eye positions, which is clearly visible in Fig. 19.

We further illustrate the lightweight fitting results with high-quality renders of several subjects in Fig. 1 and Fig. 20. In these renders the eyelids of the reconstructed eye meshes are deformed with Laplacian deformation to match the fitted eyelid interfaces, and an artistic beauty pass is performed in order to clean the face meshes and add eyelashes. Thanks to our eye rig fits, none of the subjects and gazes look cross-eyed, however incorrectly ignoring the difference between optical and visual axis will generate uncanny gazes, as shown in Fig. 21.

9. Conclusion

We present a novel eye rig based on accurate measurements from a multi-view imaging system that can reconstruct eye poses at submillimeter accuracy. Based on these high-quality eye rigs we introduce an average eye rig that can be used as prior information and allows us to do lightweight rig fitting requiring just a single pose and few manual annotations. We show that it is important to fit the visual axis of the eye and model torsion in order to avoid uncanny gazes.

Limitations and Future Work While the lightweight rig fitting aims to reduce the amount of manual annotation, still a few annotations are required. We have not investigated how the annotation and fitting process could be fully automated. This would, of course, be highly desirable for high-quality rigs, since building the high-quality rig requires a dedicated capture session and a lot of manual annotations, which is realistically only feasible for hero characters in larger applications such as film and video game productions.

The conjunctiva slides over the eyeball and deforms some of the veins while the sclera veins remain fixed. This phenomenon has not previously been modelled in computer graphics. Given the amplitude of these shifts, we believe that modeling this phenomenon could be valuable in order to further increase the realism of close-ups of modelled eye animations.

While we propose a rig that allows to pose eyes, we have not investigated the intricate patterns that govern eye motion, such as saccades or tremor. Capturing and quantifying these is a research area on its own, and we believe the proposed rig could help inform such research.

Finally, we have modelled the eyes in isolation of the surrounding skin. However, these two substantially influence each other. The eyelid is deformed as the eye moves underneath it. Future work should thus look at ways to couple these two models and provide a rig for the entire eye region.

Acknowledgements

We would like to thank our capture subjects: Michael Bernauer, Marlen Gaeumann, Sebastian Jaberg, Severin Klingler, Yeara Kozlov, and Mauritio Nitti, as well as Dr. med. Peter Maloca for advice and Prashanth Chandran for manual annotations.

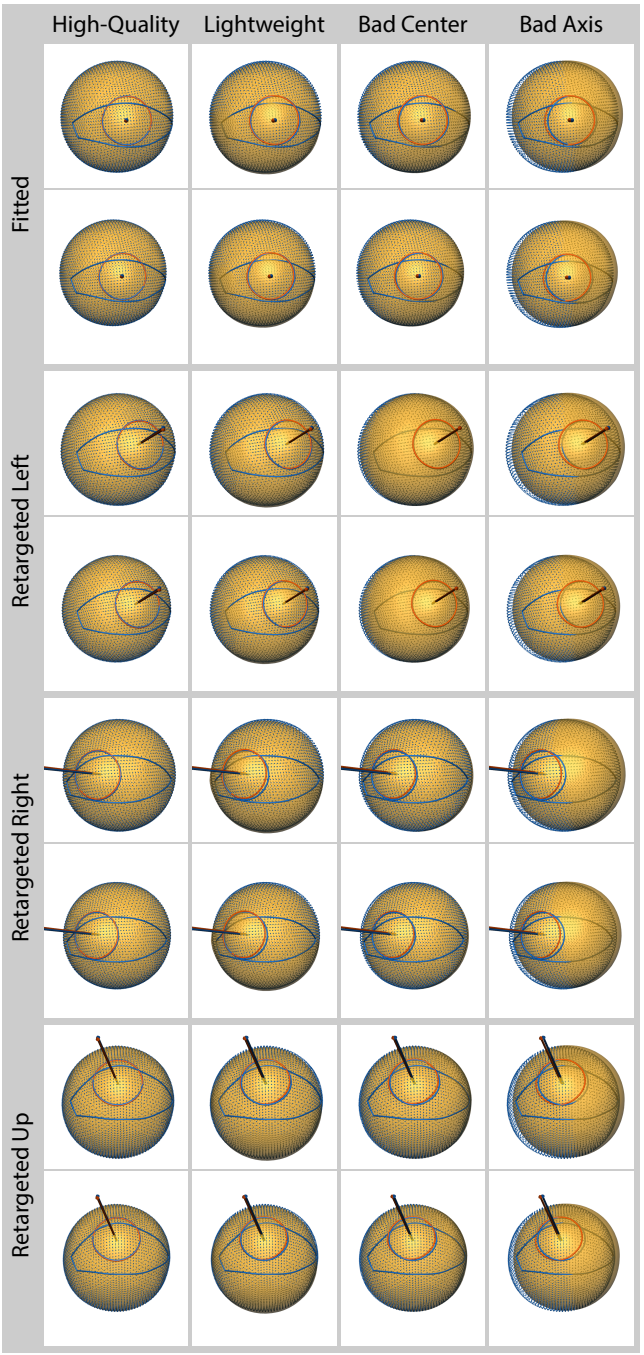

Figure 19: *This figure shows a comparison of different rigs and how well they can be retargeted to different poses. The dots (black), the eyelid interface (gray), and the limbus (gray) show the high-quality rig. The first column shows our high-quality rig (green) retargeted to the look-at point corresponding to this pose. The second column shows our lightweight rig (orange) based on the average rig and fitted to a single pose. The third column shows the effect of the rotation center offset to the left by three millimeters (red) and the forth column shows the visual axis offset to the left by five degrees (red). The pose on the top is the one used to fit the lightweight rig.*

Figure 20: *High-quality renders of different subjects (left column: four poses of one subject, right column: four different subjects). In all the examples eye shape and eye rigs are computed with our lightweight fitting method.*

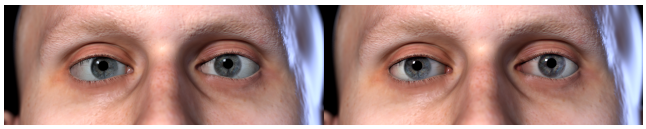

Figure 21: *Naïvely using the optical axis as the visual axis can lead to cross-eyed gazes (left). Correctly fitting the person-specific visual axis with our method helps to overcome the uncanny gaze (right).*

References

[AA11] AVUDAINAYAGAM K. V., AVUDAINAYAGAM C. S.: Simple method to measure the visual axis of the human eye. *Optics letters 36*, 10 (2011), 1803–1805. 1, 6

[AMO18] AGARWAL S., MIERLE K., OTHERS: Ceres Solver. http://ceres-solver.org, 2018. 12

[BB14] BEELER T., BRADLEY D.: Rigid stabilization of facial expressions. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 44. 4, 5

[BBB*10] BEELER T., BICKEL B., BEARDSLEY P., SUMNER B., GROSS M.: High-quality single-shot capture of facial geometry. In *ACM Transactions on Graphics (ToG)* (2010), vol. 29, ACM, p. 40. 5, 14

[BBGB16] BÉRARD P., BRADLEY D., GROSS M., BEELER T.: Lightweight eye capture using a parametric model. *ACM Trans. Graphics (Proc. SIGGRAPH) 35*, 4 (2016), 117:1–117:12. 2, 3, 5

[BBN*14] BÉRARD P., BRADLEY D., NITTI M., BEELER T., GROSS M.: High-quality capture of eyes. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 33*, 6 (2014), 223:1–223:12. 1, 2, 5

[BHB*11] BEELER T., HAHN F., BRADLEY D., BICKEL B., BEARDSLEY P., GOTSMAN C., SUMNER R. W., GROSS M.: High-quality passive facial performance capture using anchor frames. In *ACM Transactions on Graphics (TOG)* (2011), vol. 30, ACM, p. 75. 5

[BOH91] BENEL D. C. R., OTTENS D., HORST R.: Use of an eye tracking system in the usability laboratory. In *Proc. of the Human Factors Society 35th Annual Meeting* (1991), pp. 461–465. 2

[Car88] CARPENTER R. H.: *Movements of the Eyes, 2nd Rev.* Pion Limited, 1988. 4

[CKB01] CHOPRA-KHULLAR S., BADLER N. I.: Where to look? automating attending behaviors of virtual human characters. *Autonomous Agents and Multi-Agent Systems 4*, 1 (2001), 9–23. 3

[Col99] COLLEWIJN H.: Eye movement recording. *Vision Research: A Practical Guide to Laboratory Methods* (1999), 245–285. 2

[DC01] DODGE R., CLINE T. S.: The angle velocity of eye movements. *Psychological Review 8* (1901), 145–157. 2

[DLN05] DENG Z., LEWIS J. P., NEUMANN U.: Automated eye motion using texture synthesis. *IEEE CG&A 25*, 2 (2005), 24–30. 3

[Egg07] EGGERT T.: Eye movement recordings: Methods. *Neuro-Ophthalmology 40* (2007), 15–34. 2

[FB81] FISCHLER M., BOLLES R.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM 24*, 6 (1981), 381–395. 14

[FGBB09] FRANÇOIS G., GAUTRON P., BRETON G., BOUATOUCH K.: Image-based modeling of the human eye. *IEEE TVCG 15*, 5 (2009), 815–827. 2

[FH62] FRY G., HILL W.: The center of rotation of the eye*. *Optometry & Vision Science 39*, 11 (1962), 581–595. 4

[FH63] FRY G. A., HILL W.: The mechanics of elevating the eye*. *Optometry & Vision Science 40*, 12 (1963), 707–716. 4

[HJ10] HANSEN D. W., JI Q.: In the eye of the beholder: A survey of models for eyes and gaze. *IEEE PAMI 32*, 3 (2010), 478–500. 2

[HT48] HARTRIDGE H., THOMPSON L. C.: Methods of investigating eye movements. *British Journal of Ophthalmology 32* (1948), 581–591. 2

[IDP03] ITTI L., DHAVALE N., PIGHIN F.: Realistic Avatar Eye and Head Animation Using a Neurobiological Model of Visual Attention. In *Proceedings of SPIE 48th Annual International Symposium on Optical Science and Technology* (2003). 3

[JMS05] JUDD C. H., MCALLISTER C. N., STEEL W. M.: General introduction to a series of studies of eye movements by means of kinetoscopic photographs. *Psychological Review, Monograph Supplements 7* (1905), 1–16. 2

[JWGD78] JANIK S. W., WELLENS A. R., GOLDBERG M. L., DELL'OSSO L. F.: Eyes as the center of focus in the visual examination of human faces. *Perceptual and Motor Skills 47*, 3 (1978), 857–858. 1

[LBB02] LEE S. P., BADLER J. B., BADLER N. I.: Eyes alive. *ACM Trans. Graphics (Proc. SIGGRAPH) 21*, 3 (2002), 637–644. 3

[LBS*03] LEFOHN A., BUDGE B., SHIRLEY P., CARUSO R., REINHARD E.: An ocularist's approach to human iris synthesis. *IEEE CG&A 23*, 6 (2003), 70–75. 2

[LE13] LEGRAND Y., ELHAGE S. G.: *Physiological optics*, vol. 13. Springer, 2013. 1

[LMD12] LE B. H., MA X., DENG Z.: Live speech driven head-and-eye motion generators. *IEEE TVCG 18*, 11 (2012). 2

[LW90] LEVOY M., WHITAKER R.: Gaze-directed volume rendering. In *Symposium on Interactive 3D Graphics* (1990), pp. 217–223. 2

[MD09] MA X., DENG Z.: Natural eye motion synthesis by modeling gaze-head coupling. In *Proc. IEEE VR* (2009), pp. 143–150. 2

[MT62] MACKWORTH N. H., THOMAS E. L.: Head-mounted eye-marker camera. *Journal of the Optical Society of America 52* (1962), 713–716. 2

[MXL*13] MARSELLA S., XU Y., LHOMMET M., FENG A., SCHERER S., SHAPIRO A.: Virtual character performance from speech. In *Proc. SCA* (2013), pp. 25–35. 2

[PM09] PINSKIY D., MILLER E.: Realistic eye motion using procedural geometric methods. In *SIGGRAPH 2009: Talks* (2009), ACM, p. 75. 3

[PRMG16] PEJSA T., RAKITA D., MUTLU B., GLEICHER M.: Authoring directed gaze for full-body motion capture. *ACM Trans. Graphics (Proc. SIGGRAPH Asia 35*, 6 (2016). 2, 3

[RAB*14] RUHLAND K., ANDRIST S., BADLER J., PETERS C., BADLER N., GLEICHER M., MUTLU B., MCDONNELL R.: Look me in the eyes: A survey of eye and gaze animation for virtual agents and artificial systems. In *Eurographics State of the Art Reports* (2014), pp. 69–91. 2

[SBMH94] SAGAR M. A., BULLIVANT D., MALLINSON G. D., HUNTER P. J.: A virtual environment and model of the eye for surgical simulation. In *Proceedings of Computer Graphics and Interactive Techniques* (1994), pp. 205–212. 2

[SMS14] SUGANO Y., MATSUSHITA Y., SATO Y.: Learning-by-synthesis for appearance-based 3d gaze estimation. In *IEEE CVPR* (2014). 2

[VRVdB93] VAN RUN L., VAN DEN BERG A.: Binocular eye orientation during fixations: Listing's law extended to include eye vergence. *Vision research 33*, 5 (1993), 691–708. 4

[VSvdVN01] VERTEGAAL R., SLAGTER R., VAN DER VEER G., NIJHOLT A.: Eye gaze patterns in conversations: There is more to conversational agents than meets the eyes. In *Proc. Human Factors in Computing Systems* (2001), pp. 301–308. 3

[WBM*16a] WOOD E., BALTRUSAITIS T., MORENCY L. P., ROBINSON P., BULLING A.: A 3d morphable eye region model for gaze estimation. In *ECCV* (2016). 1, 2, 3

[WBM*16b] WOOD E., BALTRUSAITIS T., MORENCY L. P., ROBINSON P., BULLING A.: Learning an appearance-based gaze estimator from one million synthesized images. In *ETRA* (2016). 2

[WBM*18] WOOD E., BALTRUSAITIS T., MORENCY L.-P., ROBINSON P., BULLING A.: Gazedirector: Fully articulated eye gaze redirection in video. *Eurographics* (2018). 1, 2

[WBZ*15] WOOD E., BALTRUSAITIS T., ZHANG X., SUGANO Y., ROBINSON P., BULLING A.: Rendering of eyes for eye-shape registration and gaze estimation. In *IEEE ICCV* (2015). 2

[WLO10] WEISSENFELD A., LIU K., OSTERMANN J.: Video-realistic image-based eye animation via statistically driven state machines. *Vis. Comput. 26*, 9 (2010), 1201–1216. 3

[WSXC16] WANG C., SHI F., XIA S., CHAI J.: Realtime 3d eye gaze animation using a single rgb camera. *ACM Trans. Graphics (Proc. SIGGRAPH) 35*, 4 (2016). 1, 2

[WXLJH17] WEN Q., XU F., LU M., JUN-HAI Y.: Real-time 3d eyelids tracking from semantic edges. *ACM Transactions on Graphics (TOG)* (2017). 1, 2

[WXY16] WEN Q., XU F., YONG J.-H.: Real-time 3d eye performance reconstruction for rgbd cameras. *IEEE Transactions on Visualization and Computer Graphics* (2016). 1, 2

[ZFI11] ZORIC G., FORCHHEIMER R., I. P.: On creating multimodal virtual humans-real time speech driven facial gesturing. *Multimedia Tools and Applications 54*, 1 (2011), 165–179. 2

[ZMI99] ZHAI S., MORIMOTO C., IHDE S.: Manual and gaze input cascaded (magic) pointing. In *Proc. of the ACM CHI Human Factors in Computing Systems Conference* (1999), pp. 246–253. 2

[ZSFB15] ZHANG X., SUGANO Y., FRITZ M., BULLING A.: Appearance-based gaze estimation in the wild. In *IEEE CVPR* (2015). 2

Appendix A: Fitting: Implementation Details

We treat the task to find the optimal eye configurations as an energy minimization problem, which we solve using the Ceres solver [AMO18]. In this section we will describe the chosen constraints and resulting energy functionals in detail.

Figure 22: *Image annotations: limbus (yellow/turquoise), lower eyelid (red/blue), tear duct (orange/cyan), and pupil (brown/gray).*

Annotation Fitting

The eyeball positions are first fitted to image annotations. As shown in Fig. 22, we manually annotate the limbus, the eyelid interfaces, the pupils, and the eye corners. The features are annotated in approximately three camera views each, selecting vantage points where the feature is best visible. For each of these annotations we formulate a constraint, which together form the following optimization problem

$$E_{annotation} = E_{limbus} + E_{eyelid} + E_{shape} + E_{corners} + E_{pupil}. \quad (7)$$

Limbus constraint The limbus constraint forces the projection of the model limbus contour to be close to the annotated limbus contour in the image. The similarity of the two contours is computed in image space by sampling the annotated contour every millimeter. For each sample point $\boldsymbol{a}_i^{lim}$, the distance to the closest point on the model limbus contour is computed. This corresponding point is defined by a single curve parameter c_i^{lim}, which is part of the optimization to allow the correspondence to slide along the limbus contour. Via Eq. 2 and Eq. 1 the curve parameter c_i^{lim} is mapped to world space and then projected via $Camera(\cdot)$ into the image plane

$$\boldsymbol{x}_i^{lim} = Camera(\,Eyeball(\,Limbus(c_i^{lim}),\hat{\mathcal{P}})\,)$$
$$E_{limbus} = w_{limbus} \cdot \frac{1}{n^{lim}} \cdot \sum_{i=1}^{n^{lim}} \left\| \boldsymbol{x}_i^{lim} - \boldsymbol{a}_i^{lim} \right\|^2, \quad (8)$$

where we compute the weighted L_2 norm. The weights for this and the other energy terms are tabulated in Table 1.

w_{limbus}	$= 1$		w^a_{pupil}	$= 1$
w_{eyelid}	$= 1$		w^b_{pupil}	$= 10$
$w_{corners}$	$= 10$		$w_{inter-camera}$	$= 4000$
w_{shape}	$= 10$		$w_{inter-frame}$	$= 4$
			$w_{reference-frame}$	$= 4$

Table 1: *Weights used to balance the individual energy terms.*

Eyelid interface constraint Conceptually, the eyelid interface constraints are identical to the limbus constraints. They force the projection of the model eyelid interface to be close to the corresponding annotations. These annotations are sampled every millimeter and each sample has a sliding correspondence on the model defined by a curve parameters c_i^{lid}. This parameter is part of the optimization and is initialized with the closest point. Analogous to the limbus constraint the residuals are computed in camera space as the weighted

L_2 difference of the annotation samples $\boldsymbol{a}_i^{lid}$ and their projected correspondences $\boldsymbol{x}_i^{lid}$:

$$\boldsymbol{x}_i^{lid} = Camera(\,Eyeball(\,Eyelid(c_i^{lid}, \hat{\mathcal{P}}), \hat{\mathcal{P}})\,)$$
$$E_{eyelid} = w_{eyelid} \cdot \frac{1}{n^{lid}} \cdot \sum_{i=1}^{n^{lid}} \left\| \boldsymbol{x}_i^{lid} - \boldsymbol{a}_i^{lid} \right\|^2. \quad (9)$$

The eyelid interface is oftentimes only partially visible in a camera due to occlusion by the eyeball, and hence we have to take into account visibility when computing correspondences. As visibility computation is costly and not easily differentiable, we precompute it and keep it fixed during optimization. After convergence we recompute visibility and continue to optimize with updated constraints. We found two such alternating iterations to be sufficient.

Eyelid interface shape constraint The chosen eyelid interface model can represent shapes that are not realistic. To prevent the optimization to get stuck in such a configuration we add an eyelid interface shape constraint. This term penalizes angles α_i between three successive control points $\boldsymbol{c}_{i-1}$, $\boldsymbol{c}_i$, and $\boldsymbol{c}_{i+1}$ of the upper and lower eyelid interface curves. If the angle is smaller than $\alpha_{concave} = 10^\circ$ or bigger than $\alpha_{convex} = 30^\circ$ the curve is penalized with

$$E_{shape} = w_{shape} \cdot \frac{1}{n^{shp}} \cdot \sum_{i=1}^{n^{shp}} \left\| d_i^{shp} \right\|^2 \quad (10)$$

$$d_i^{shp} = \begin{cases} \alpha_i - \alpha_{convex}, & \alpha_i > \alpha_{convex} \\ \alpha_i + \alpha_{concave}, & \alpha_i < -\alpha_{concave} \\ 0, & otherwise \end{cases}$$

$$\alpha_i = angle(\boldsymbol{c}_{i-1}, \boldsymbol{c}_i, \boldsymbol{c}_{i+1}). \quad (11)$$

Eye corner constraint The eye corner constraint is a special case of the eyelid interface constraint and minimizes the weighted L_2 distance between the projection $\boldsymbol{x}_i^{cor}$ of the eyelid interface end points $c_i^{cor} \in 0, 1$ and their corresponding corner annotations $\boldsymbol{a}_i^{cor}$

$$\boldsymbol{x}_i^{cor} = Camera(\,Eyeball(\,Eyelid(c_i^{cor}, \hat{\mathcal{P}}), \hat{\mathcal{P}})\,)$$
$$E_{corners} = w_{corners} \cdot \frac{1}{n^{cor}} \sum_{i=1}^{n^{cor}} \left\| \boldsymbol{x}_i^{cor} - \boldsymbol{a}_i^{cor} \right\|^2. \quad (12)$$

Pupil constraint The pupil constraint forces the projection of the pupil model to be close to the pupil annotations. Conceptually, this is very similar to the limbus constraint but with the major difference that we have to take into account refraction at the cornea, for which no closed form solution exists. So instead we do not compute the residual in the image plane but in world space. We intersect the camera ray from the annotation $\boldsymbol{a}_i^{pup}$ with the cornea, providing the intersection point $\boldsymbol{y}_i^{pup}$ in texture space. We then refract the ray at this point and compute the distance between the refracted ray $\boldsymbol{r}_i^{pup}$ and the model pupil circle $Pupil(\hat{\mathcal{P}})$

$$\boldsymbol{r}_i^{pup} = Refract(\,Camera^{-1}(\boldsymbol{a}_i^{pup}), Eyeball(\boldsymbol{y}_i^{pup}, \hat{\mathcal{P}})\,)$$
$$E_{pupil}^a = w_{pupil}^a \cdot \frac{1}{n^{pup}} \cdot \sum_{i=1}^{n^{pup}} \left\| \boldsymbol{r}_i^{pup}, Pupil(\hat{\mathcal{P}}) \right\|_{ray-circle}^2. \quad (13)$$

However, since the shape of the cornea changes during the optimization, we cannot keep $\boldsymbol{y}_i^{pup}$ fixed but allow it to slide on the surface of the eyeball, such that its projection back into the image plane always coincides with the sample $\boldsymbol{a}_i^{pup}$

$$\boldsymbol{x}_i^{pup} = Camera(\,Eyeball(\boldsymbol{y}_i^{pup}, \hat{\mathcal{P}})\,)$$
$$E_{pupil}^b = w_{pupil}^b \cdot \frac{1}{n^{pup}} \cdot \sum_{i=1}^{n^{pup}} \left\| \boldsymbol{x}_i^{pup} - \boldsymbol{a}_i^{pup} \right\|^2. \quad (14)$$

The final pupil energy is given by the sum of Eq. 13 and Eq. 14.

Photometric Refinement

The annotation-based fitting produces a first estimate of the eye positions, but manual annotations are not pixel-perfect and lead to inaccuracies. To overcome these we introduce an image-based refinement term that does not depend on manual annotations, but can highly benefit from the close initial guess they provide. The idea is to incorporate additional constraints that enforce photoconsistency across cameras and across frames by projecting 3D patches of the eye into the different images to compute the discrepancy. These constraints are defined only on the unobstructed parts of the sclera and we first describe how we mask out occluders, such as skin or eyelashes, and introduce the photometric inter-camera and the inter-frame constraints subsequently. The constraints are formulated in the same framework and are integrated with the $E_{annotation}$ energy

$$\begin{aligned} E_{refinement} = {} & E_{annotation} \\ & + w_{inter-camera} \cdot E_{inter-camera} \\ & + w_{inter-frame} \cdot E_{inter-frame} \\ & + w_{reference-frame} \cdot E_{reference-frame}. \end{aligned} \quad (15)$$

Mask computation We compute a sclera mask by projecting the fitted eyelid interface and limbus contour from the current estimate into the camera. Unfortunately, for oblique views the sclera part might still be occluded by eyelashes, the nose or other skin parts. The nose and skin parts are masked using the face scan geometry, but eyelashes are not present in the face scan. Therefore, we create an eyelash geometry proxy as shown in Fig. 23. This proxy follows the fitted eyelid interfaces and consists of two parts: the eyelid margin and the actual eyelashes. The margin is a six millimeters wide section perpendicular to the eyeball surface. The eyelashes are connected at the end of the eye margin and extend the proxy further out by 7 millimeters but are bent down at a 45 degrees angle. This proxy is then used together with the face geometry to render sclera masks for both eyes in all cameras and all frames.

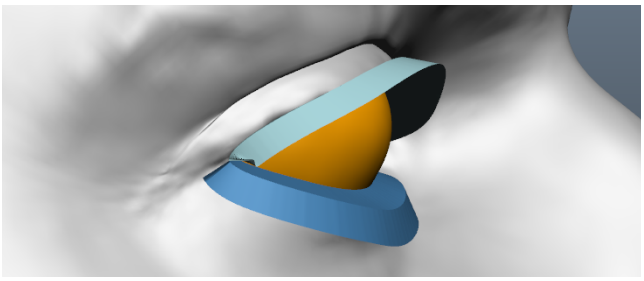

Figure 23: *The proxy eyelid geometry used to compute the sclera masks. The figure shows the eyeball (orange) and skin (gray) geometries. The eyelash proxies (blue) consist of an eyelid margin (perpendicular to the eyeball surface) and an eyelash part.*

Inter-camera constraint The inter-camera constraint tries to maximize the similarity of a 3D patch from one frame projected into all cameras. The approach is to sample the space along patch normals to find better positions. These positions are then added as constraints to the optimization problem. We select points on the sclera on a regular grid in texture space so that they are separated by about 0.5 millimeters. We prune points that are not seen by at least two cameras. Inspired by Beeler et al. [BBB*10] we create a 25x25 pixel 3D patch for each sample point that is offset forwards and backwards in steps of 0.1 millimeters up to ± 1.5 millimeters. These patches are not planar but have the local shape of the eyeball. At each offset the algorithm computes the normalized cross-correlation between a reference camera and all the other cameras and weights the correlations by the foreshortening angle. We use the masks to evaluate the visibility of the patches in each camera. The algorithm chooses the reference camera based on a structure measure, which is the sum of neighbor pixel differences. This is required since we cannot solely rely on foreshortening as some cameras might be out-of-focus due to the shallow depth of field of the cameras. The optimization residual is formed by the offset position with the smallest photometric error $\boldsymbol{x}_i^{opt}$ and the corresponding closest point on the eyeball surface. The closest point is defined by a texture coordinate $\boldsymbol{y}_i^{inter-camera}$ and is part of the optimization parameters.

$$\begin{aligned} \boldsymbol{x}_i^{inter-camera} &= Eyeball(\boldsymbol{y}_i^{inter-camera}, \hat{\mathcal{P}}) \\ E_{inter-camera} &= \left\| \boldsymbol{x}_i^{inter-camera} - \boldsymbol{x}_i^{opt} \right\|^2. \end{aligned} \tag{16}$$

Inter-frame constraint For a given camera the inter-frame constraint tracks and links the same features of two adjacent frames, for which the gaze direction differs by no more than 20 degrees. To compute correspondences between the frames for a given camera, we compute a texture for both frames. The veins are the features which are the easiest to track. Thus, we band-pass filter one of the textures and pick only a small percentage (0.05%) of the pixels with the highest response as samples. Then, the feature density is reduced such that features are separated by at least one millimeter using a non-maxima suppression strategy. For the remaining features we compute a correspondence in the other texture with a brute force search. The search window is 21 pixels wide and we search up to a maximum distance of 30 pixels. To speed up the search we use an image pyramid with three levels and initialize the next layer with the result of the coarser one. We filter the correspondences using RANSAC [FB81] as follows. For every two features in one texture we compute the similarity transform that transforms the features into the corresponding features of the other texture. Given this transformation we measure how well all the features map onto their corresponding features. Features with a distance bigger than 0.25 millimeters to their correspondences are considered to be outliers and ignored. Ultimately, the transformation with the overall highest score is used to filter outliers. If there are less than six correspondences we completely ignore the frame. When these features in texture space between two frames match up, then the eye configurations are reconstructed correctly. To constrain the optimization towards that configuration, we project for every feature j the texture locations $\boldsymbol{f}_i^j$ and $\boldsymbol{f}_k^j$ into the camera yielding $\boldsymbol{a}_i^j$ and $\boldsymbol{a}_k^j$ for frames i and k, respectively:

$$\boldsymbol{a}_i^j = Camera(\, Eyeball(\boldsymbol{f}_i^j, \hat{\mathcal{P}}_i)\,). \tag{17}$$

We now add a free variable $\boldsymbol{f}^j$ to the optimization, with the intent that this represents the true feature location on the eyeball, and hence projects onto all $\boldsymbol{a}_i^j$ in the respective frames.

$$\begin{aligned} \boldsymbol{x}_i^j &= Camera(\, Eyeball(\boldsymbol{f}^j, \hat{\mathcal{P}}_i)\,) \\ E_{inter-frame} &= \sum_{i,j} \left\| \boldsymbol{x}_i^j - \boldsymbol{a}_i^j \right\|^2. \end{aligned} \tag{18}$$

Reference-frame rotation constraint The sclera is covered by a protective, mostly transparent skin called the conjunctiva. This skin is not firmly attached to the eyeball, but actually slides over it, stretching and folding during eye rotations. Since both sclera and conjunctiva contain veins and other features, these features move relative to each other (Fig. 10) which poses a challenge for the inter-frame constraints and might cause drift as the inter-frame constraints are only concerned with neighboring poses. To prevent this drift we add a rotation constraint that constrains the axial rotation of a pose with respect to the pose in the reference frame. Since every pose is constrained to the same reference pose the drift can be eliminated. The relative motion of conjunctiva and sclera is minimal at the limbus, where the conjunctiva is thinnest and more firmly connected to the sclera. We compute a photometric residual from this area inside the texture map, which will constrain the torsion $\hat{\Theta}_z$ to align the two poses. With this final refinement step we can accurately compute the poses of the eyes in all frames individually.