ETH zürich Research Collection

Doctoral Thesis

Eye Reconstruction and Modeling for Digital Humans

Author(s):
Bérard, Pascal

Publication Date:
2018

Permanent Link:

https://doi.org/10.3929/ethz-b-000271976 →

Rights / License:

In Copyright - Non-Commercial Use Permitted →

This page was generated automatically upon download from the ETH Zurich Research Collection. For more information please consult the Terms of use.

ETH Library

Diss. ETH No. 25064

Eye Reconstruction and Modeling for Digital Humans

A thesis submitted to attain the degree of
Doctor of Sciences of ETH Zurich
(Dr. sc. ETH Zurich)

Presented by
Pascal Bérard
MSc in Microengineering, EPFL, Switzerland
Born 24.11.1987
Citizen of Vollèges (VS), Switzerland

Accepted on the recommendation of
Prof. Dr. Markus Gross, examiner
Prof. Dr. Christian Theobalt, co-examiner
Dr. Thabo Beeler, co-examiner

2018

Abstract

The creation of digital humans is a long-standing challenge of computer graphics. Digital humans are tremendously important for applications in visual effects and virtual reality. The traditional way to generate digital humans is through scanning. Facial scanning in general has become ubiquitous in digital media, but most efforts have focused on reconstructing the skin only. The most important part of a digital human are arguably the eyes. Even though the human eye is one of the central features of an individual's appearance, its shape and motion have so far been mostly approximated in the computer graphics community with gross simplifications. To fill this gap, we investigate in this thesis methods for the creation of eyes for digital humans. We present algorithms for the reconstruction, the modeling, and the rigging of eyes for computer animation and tracking applications.

To faithfully reproduce all the intricacies of the human eye we propose a novel capture system that is capable of accurately reconstructing all the visible parts of the eye: the white *sclera*, the transparent *cornea* and the non-rigidly deforming colored *iris*. These components exhibit very different appearance properties and thus we propose a hybrid reconstruction method that addresses them individually, resulting in a complete model of both spatio-temporal shape and texture at an unprecedented level of detail.

This capture system is time-consuming to use and cumbersome for the actor making it impractical for general use. To address these constraints we present the first approach for high-quality *lightweight* eye capture, which leverages a database of pre-captured eyes to guide the reconstruction of new eyes from much less constrained inputs, such as traditional single-shot face scanners or even a single photo from the internet. This is accomplished with a new parametric model of the eye built from the database, and a novel image-based model fitting algorithm.

For eye animation we present a novel eye rig informed by ophthalmology findings and based on accurate measurements from a new multi-view imaging system that can reconstruct eye poses at submillimeter accuracy. Our goal is to raise the awareness in the computer graphics and vision communities that eye movement is more complex than typically assumed, and provide a new eye rig for animation that models this complexity.

Finally, we believe that the findings of this thesis will alter current assumptions in computer graphics regarding human eyes, and our work has the potential to significantly impact the way that eyes of digital humans will be modelled in the future.

Zusammenfassung

Das Erstellen von digitalen Doppelgängern ist eine Herausforderung, die das Gebiet der Computergrafik schon lange beschäftigt. Digitale Doppelgänger sind essentiell für Anwendungen in der virtuellen Realität oder in visuellen Effekten in Filmen und werden klassischerweise durch Scannen erstellt. Insbesondere Gesichtsscanning ist in digitalen Medien allgegenwärtig geworden. Die meisten Forschungsarbeiten haben sich jedoch auf die Rekonstruktion der Haut beschränkt. Obwohl das Auge vermutlich das wichtigste Gesichtsmerkmal ist und eine zentrale Rolle im Erscheinungsbild eines Individuums darstellt, wurde seine Form und Bewegung in der Computergrafik mit groben Vereinfachungen angenähert. Um diese Lücke zu schliessen, untersuchen wir in dieser Arbeit Methoden zum Erstellen von Augen für digitale Doppelgänger. Wir präsentieren Algorithmen für die Rekonstruktion, die Modellierung und das Rigging von Augen für Computeranimationen und Tracking-Anwendungen.

Um alle Feinheiten des menschlichen Auges originalgetreu wiederzugeben, schlagen wir ein neuartiges Erfassungssystem vor, das in der Lage ist alle sichtbaren Teile des Auges exakt zu rekonstruieren: die weisse *Lederhaut*, die transparente *Hornhaut* und die sich deformierende farbige *Iris*. Diese Teile weisen alle sehr unterschiedliche visuelle und optische Eigenschaften auf und deshalb schlagen wir eine hybride Rekonstruktionsmethode vor, die die verschiedenen Eigenschaften berücksichtigt. Daraus resultiert ein vollständiges Augenmodell, das die Form und die Deformation als auch die Textur in einem noch nie dagewesenen Detaillierungsgrad modelliert.

Dieses Erfassungssystem ist zeitaufwändig und umständlich in der Benützung und in der Anwendung für den Darsteller, womit es sich für den allgemeinen Gebrauch nicht eignet. Um diese Einschränkungen zu beheben, stellen wir einen neuen Ansatz für eine *benutzerfreundlichere* Augenerfassung vor, die weiterhin hochwertige Augen generiert. Dabei verwenden wir eine Datenbank mit hochqualitativen Augenscans, aus der neue Augen generiert werden. Dieser Prozess wird durch einfache Eingaben gelenkt. Dazu kann z.B. ein traditioneller Gesichtsscan oder sogar ein einziges Foto aus dem Internet verwendet werden. Die Robustheit vom System wird mit einem neuen parametrischen Augenmodell und einem neuartigen bildbasierten Algorithmus zum Anpassen der Modellparameter erreicht.

Für die Augenanimation stellen wir ein neuartiges Augen-Rig vor, das auf den Erkenntnissen der Ophthalmologie und auf genauen Messungen eines neuen Multikamerasystems basiert, mit dem sich Augenpositionen mit Submillimetergenauigkeit bestimmen lassen. Unser Ziel ist es, das Bewusstsein in den Computergrafik- und Computervision-Gemeinschaften zu schärfen, dass Augenbewegungen komplexer sind als üblicherweise angenommen. Dazu führen wir ein neues Augen-Rig für die Animation ein, das diese Komplexität modelliert.

Wir glauben, dass die Resultate dieser Arbeit die aktuellen Annahmen in der Computergrafik in Bezug auf die menschlichen Augen beeinflussen werden und wir glauben, dass unsere Arbeit das Potenzial hat signifikante Auswirkungen auf den Modellierungsprozess von Augen von digitalen Doppelgängern zu haben.

Acknowledgments

Since I was very little, my parents allowed me to experiment and tinker with everything, even if that meant leaving behind a gigantic mess. I am immensely grateful for their tolerance and I believe this allowed me to become who I am. I also want to thank my sister, Stéphanie, my brother, Michel, and my sister-in-law, Karima, for supporting me throughout this PhD.

I would like to sincerely thank my adviser Prof. Markus Gross. He gave me the opportunity to work in this exciting field and gave me the freedom to investigate my own ideas. He also believed that I can do a PhD in computer graphics without a strong background in computer science. I am immensely grateful for that.

The same is true for Thabo Beeler and Derek Bradley who supervised me since the very beginning of my PhD. Their advice and guidance was invaluable and helped me not to forget the big picture of what we want to achieve. There is nothing better than to learn from the best in the field, and this work wouldn't have been possible without their ideas and it wouldn't have been possible if I wasn't able to build on top of what they have built. I'm also grateful for the countless hours they spent rewording and polishing the papers that make up this thesis.

Thank you, Christian Theobalt, for refereeing my examination and reviewing my thesis.

I would have published nothing without my coauthors. I am grateful to have worked with all of them: Thabo Beeler, Amit Bermano, Derek Bradley, Alexandre Chapiro, Markus Gross, Maurizio Nitti, Mattia Ryffel, Stefan Schmid, Robert Sumner, and Fabio Zünd.

Many thanks go to my collaborators Maurizio Nitti and Alessia Marra for creating the illustrations and renders that made our papers so much nicer to look at. Special thanks go to Maurizio Nitti for his endless patience.

I wish to thank Dr. med. Peter Maloca and Prof. Dr. Dr. Jens Funk and his team at UniversitätsSpital Zürich for the helpful discussions and their eye-opening remarks.

I would like to thank Lewis Siegel and Michael Koperwas for their industry perspective.

Prof. Gaudenz Danuser introduced me to world of research. I am grateful that he encouraged me to do a PhD.

I would also like to thank all of our eye models, who spent countless hours in uncomfortable positions and made this work possible.

Thank you, Jan Wezel and Ronnie Gänsli, for your support in building the hardware required for these projects.

I am fortunate to have worked and spent time with people of the Computer Graphics Lab, Disney Research Zurich, as well as the members of the Interactive Geometry Lab who inspired me with their works. They all made the PhD so enjoyable and deadlines so much more human.

I was lucky to share an office during my first year at ETH with Fabian Hahn. He is an exceptional researcher and friend, who taught me how to code.

Thank you, Antoine Milliez, for making the dull days so much more enjoyable with your great jokes!

Special thanks for my collaborators and friends: Simone Meyer, Fabio Zünd, Tanja Käser, Yeara Kozlov, Virag Varga, Paulo Gothardo, Severin Klingler, Christian Schüller, Oliver Glauser, Leo Helminger, Christian Schumacher, Alexandre Chapiro, Loïc Ciccone, Vittorio Megaro, Riccardo Roveri, Ivan Ovinnikov, Endri Dibra, Romain Prévost, Kaan Yücer. I will miss the great time and the refreshing coffee breaks.

Finally, I would like to thank all my friends who supported and kept me going during my PhD.

"Discovery consists not in seeking new lands but in seeing with new eyes."

— Marcel Proust

Contents

Introduction **1**
1.1 Contributions 8
1.2 Publications 9

Related Work **11**
2.1 Reconstruction and Modeling 11
2.2 Iris Deformation 12
2.3 Medical Instruments 13
2.4 Facial Capture 13
2.5 Non-Rigid Alignment 14
2.6 Texture and Geometry Synthesis 14
2.7 Eye Tracking and Gaze Estimation 15
2.8 Eye Rigging and Animation 16

Eye Anatomy **17**
3.1 Eyeball 17
3.2 Sclera and Conjunctiva 19
3.3 Cornea 19
3.4 Limbus 20
3.5 Iris 21
3.6 Pupil 22
3.7 Muscles 22

Eye Reconstruction **23**
4.1 Data Acquisition 25
4.1.1 Capture Setup 25
4.1.2 Calibration 26
4.1.3 Image Acquisition 27
4.1.4 Initial Reconstruction 27
4.2 Sclera 27
4.2.1 Image Segmentation 28
4.2.2 Mesh Segmentation 29
4.2.3 Pose Registration 30
4.2.4 Sclera Merging 31

4.2.5 Sclera Texturing . . . 32
4.3 Cornea . . . 33
4.3.1 Theory . . . 34
4.3.2 Constraint Initalization . . . 34
4.3.3 Surface Reconstruction . . . 36
4.3.4 Cornea-Eyeball Merging . . . 37
4.4 Iris . . . 37
4.4.1 Pupil Reconstruction . . . 37
4.4.2 Iris mesh generation . . . 39
4.4.3 Mesh cleanup . . . 40
4.4.4 Mesh Propagation . . . 41
4.4.5 Temporal Smoothing and Interpolation . . . 41
4.4.6 Iris Texturing . . . 42
4.5 Results . . . 42

Parametric Eye Model **55**
5.1 Input Data . . . 56
5.2 Eyeball Model . . . 56
5.3 Iris Model . . . 60
5.3.1 Iris Texture Synthesis . . . 61
5.3.2 Iris Geometry Synthesis . . . 64
5.4 Sclera Vein Model . . . 65
5.4.1 Vein Model . . . 66
5.4.2 Vein Rendering . . . 68
5.5 Model Fitting . . . 70
5.5.1 Multi-View Fitting . . . 70
5.5.2 Single Image Fitting . . . 73
5.6 Results . . . 73

Eye Rigging **85**
6.1 Eye Rig . . . 86
6.1.1 Eye Shape . . . 86
6.1.2 Eye Motion . . . 88
6.1.3 Eye Positioning . . . 90
6.1.4 Eye Control . . . 91
6.2 Data Acquisition . . . 91
6.3 Eye Configuration Reconstruction . . . 93
6.3.1 Annotation Fitting . . . 95
6.3.2 Photometric Refinement . . . 98
6.4 Eye Rig Fitting . . . 102
6.4.1 Listing's Model . . . 102
6.4.2 Translation Model . . . 103

6.4.3 Eye Rig 103
6.4.4 Visual Axis 105
6.5 Results 105

Conclusion **113**
7.1 Limitations 114
7.2 Outlook 115

Appendix **117**

References **119**

CHAPTER 1

Introduction

Creating photo-realistic digital humans is a long-standing grand challenge in computer graphics. Applications for digital humans include video games, visual effects in films, medical applications and personalized figurines. One of the cornerstones of producing digital doubles is capturing an actor's face. Several decades of research have pushed facial capture technology to an incredible level of quality, where it is becoming difficult to distinguish the difference between digital faces and real ones. An example for such a digital human is Mike depicted in Fig. 1.1. The members of the *wikihuman.org* project created Mike to demonstrate state-of-the-art methods for the creation of digital humans.

A lot of research went into better models and simpler capture methods for digital humans. However, most research has focused on the facial skin, ignoring other important characteristics like the eyes. The eyes are arguably the most important part of the face, as this is where humans tend to focus when looking at someone. Eyes can convey emotions and foretell the actions of a person and subtle inaccuracies in the eyes of a character can make the difference between realistic and uncanny.

In this thesis we present methods for the entire digital eye creation pipeline. This includes reconstructing the visible parts of the eye, modeling the variability of human eyes with a parametric model, and rigging the position and motion for animation and tracking applications.

While a simple modeled or simulated eye may be sufficient for background characters, current industry practices spend significant effort to manually create eyes of hero characters. In this thesis, we argue that generic eye mod-

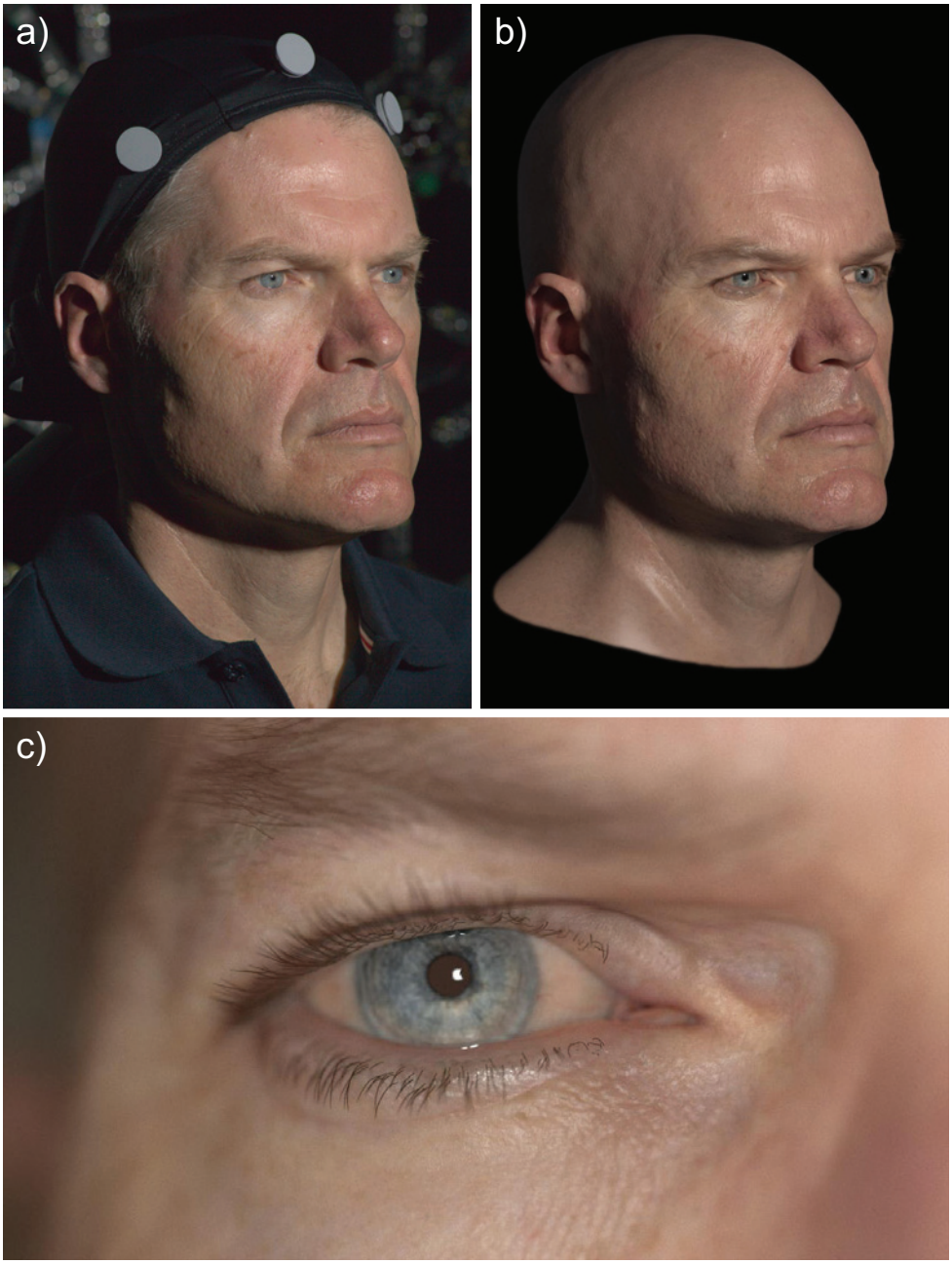

Figure 1.1: *Mike is a state-of-the-art digital human. He is the result of an industry-wide collaboration of researchers, visual effect specialists, and artists that came together in the wikihuman project with the goal to create an open, and publicly available data set of a digital human. The methods presented in this thesis have been used to scan Mike's eyes. The figure shows a reference photograph (a), a photo-realistic render from the same view (b) and a close-up render of the right eye (c) reconstructed with the methods presented in this thesis. Images courtesy of wikihuman.org.*

els typically used in computer graphics are insufficient for capturing the individual identity of a digital human. The shape of the eye is generally approximated by two spheres, a big one for the sclera and a smaller one for the cornea [Lefohn et al., 2003; Ruhland et al., 2014]. The iris is often thought of as a planar disc, or as a cone to fake the refraction of the cornea. The constriction and dilation of the pupil is typically modelled as planar, radial motion and the out-of-plane deformation of the iris is generally neglected [Ruhland et al., 2014]. Figure 1.2 shows such a generic CG eye.

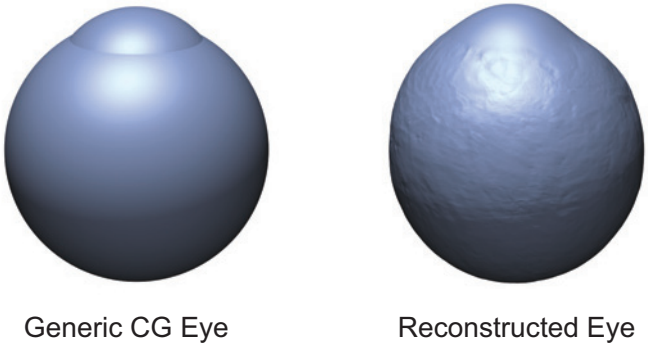

Figure 1.2: *The shape of a generic CG eye represents only a low order approximation of an individual eye, while the proposed method reconstructs all its intricacies.*

Our reconstruction method can greatly reduce the time spent and help increase the realism of the eye. As an example, Figure 1.2 presents an eye that is reconstructed by the method proposed in Chapter 4. Our reconstruction specifically captures the overall shape and spatial surface variation of the sclera including a detailed vein texture, the complex shape, texture and deformation of the iris, and even properties of the transparent cornea including the exact curvature along with the refractive index at the boundary. This example demonstrates that the aforementioned assumptions only roughly approximate the true physiology of the eye, and thus cannot represent actor-specific details that can greatly increase the realism of a digital double. Furthermore, the eyeball exhibits strong asymmetry, contains microscopic surface details and imperfections such as *Pingueculas*[1] - all of which are very person-specific. The micro-geometry of the iris is as unique to every person as a fingerprint, and its position and deformation depends on the accommodation of the underlying lens. These are just a few examples of eye details that cannot be captured with traditional models. Through the results of this thesis we will show several more examples, in particular when it comes to the dynamic deformation of the iris during *pupillary response*[2].

[1] A degeneration of the fibers of the sclera resulting in a small bump.
[2] Varying pupil size via relaxation/contraction of the iris dilator muscle.

To overcome the limitations of generic eye models and accurately reproduce the intricacies of a human eye, we argue that eyes should be captured and reconstructed from images of real actors, analogous to the established practice of skin reconstruction through facial scanning. The eye, however, is more complex than skin, which is often assumed to be a diffuse Lambertian surface in most reconstruction methods. The human eye is a heterogeneous compound of opaque and transparent surfaces with a continuous transition between the two, and even surfaces that are visually distorted due to refraction. This complexity makes capturing an eye very challenging, requiring a novel algorithm that combines several complementary techniques for image-based reconstruction. In this work, we propose the first system capable of reconstructing the spatio-temporal shape of all visible parts of the eye; the sclera, the cornea, and the iris, representing a large step forward in realistic eye modeling. Our approach not only allows us to create more realistic digital humans for visual effects and computer games by scanning actors, but it also provides the ability to capture the accurate spatio-temporal shape of an eye in-vivo.

While the results of our eye reconstruction system are compelling, the acquisition process is both time consuming and uncomfortable for the actors, as they must lie horizontally with a constraining neck brace while manually holding their eye open for dozens of photos over a 20 minute period for each eye. The physical burden of that approach is quite far from the single shot face scanners that exist today, which are as easy as taking a single photo in a comfortable setting, and thus the applicability of their method is largely limited.

In this thesis, we present a new lightweight approach to eye capture that achieves a comparable level of quality as our eye reconstructions but from input data that can be obtained using traditional single-shot face scanning methods or even just from a single image. Our key idea is to build a parametric model of the eye, given a training database of high-quality scans. Our model succinctly captures the unique variations present across the different components of the eye labeled in Fig. 1.3, including 1 - the overall size and shape of the eyeball and cornea, 2 - the detailed shape and color of the iris and its deformation under pupil dilation, and 3 - the detailed vein structure of the sclera which contributes to both its color and fine-scale surface details.

Given our model, new and unique human eyes can be created. Aspects like the shape or the color can be controlled without in-depth knowledge of the subtleties of real eyes. Furthermore, we propose a novel fitting algorithm to reconstruct eyes from sparse input data, namely multi-view images, i.e. from a single-shot multi-view face scanner. The results are very plausible

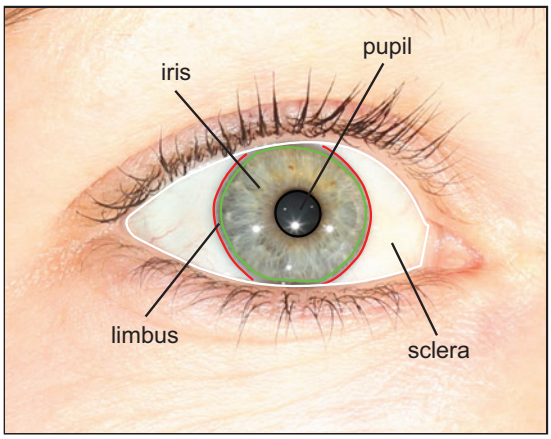

Figure 1.3: *The visually salient parts of a human eye include the black pupil, the colored iris, and the limbus that demarcates the transition from the white sclera to the transparent cornea.*

eye reconstructions with realistic details from a simple capture setup, which can be combined with the face scan to provide a more complete digital face model. In this work we demonstrate results using the face scanner of Beeler et al. [2010], however our fitting approach is flexible and can be applied to any traditional face capture setup. Furthermore, by reducing the complexity to a few intuitive parameters, we show that our model can be fit to just single images of eyes or even artistic renditions, providing an invaluable tool for fast eye modeling or reconstruction from internet photos. We demonstrate the versatility of our model and fitting approach by reconstructing several different eyes ranging in size, shape, iris color and vein structure.

Besides fitting single frames and poses the system can also be extended to fit entire sequences. This allows for the analysis of the three-dimensional position and orientation during gaze motion. An accurate eyeball pose is important since it directly affects the eye region through the interaction with surrounding tissues and muscles. 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 [Janik et al., 1978]. 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 and reproduce the way our eyes move for digital characters.

When creating eye rigs, animators traditionally think of the eyeball as a sphere, which is being rotated in place such that its optical axis points to where the character should be looking (Fig. 1.4 a). However, from our eye reconstruction work we know that the eye shape is not a sphere, and is even

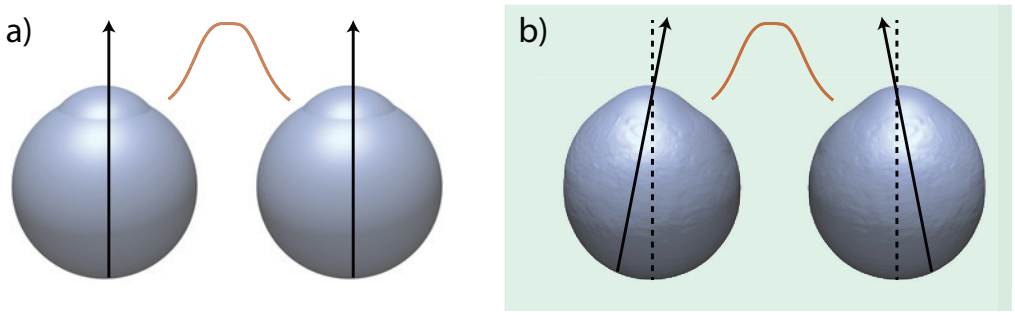

Figure 1.4: *Eye Model: a) Traditional eye models assume the eye to be roughly spherical and rotating around its center. The gaze direction is assumed to correspond to the optical axis of the eye (black arrows). b) The proposed eye model takes into account that the eye is not perfectly spherical and does not simply rotate around its center. Furthermore it respects the fact that the gaze direction is tilted towards the nose (see also Fig. 3.1 (b)).*

asymmetric around the optical axis. This, of course, begs the question of how correct these other assumptions are, and answering this question is the main focus of Chapter 6 of this thesis. We explore ophthalmological models for eye motion and assess their relevancy and applicability in the context of computer graphics.

The eye is not a rotational apparatus but instead is being pulled into place by six muscles, two for each degree of rotation (Fig. 3.1 a). These muscles are activated in an orchestrated manner to control the gaze of the eye. As a consequence, the eyeball actually does translate within its socket, meaning that its rotational pivot is not a single point but actually lies on a manifold. Furthermore, the eye is not simply rotated horizontally and vertically but also exhibits considerable rotation around its optical axis, called torsion. With the emersion of head mounted displays for augmented and virtual reality applications, modeling these phenomena may become central to allow for optimal foveal rendering.

A very important fact 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. 3.1 b, the fovea is slightly shifted away from the nose, causing the visual axis to be tilted towards the nose (Fig. 1.4 b), on average around five degrees for adults [LeGrand and ElHage, 2013]. This is an extremely important detail that cannot be neglected as otherwise the digital character will appear slightly cross-eyed, causing uncanny gazes.

To be relevant for computer vision and computer graphics applications, a phenomenon must be visible outside of ophthalmologic equipment, i.e. in imagery captured by ordinary cameras. We employ a passive multi-view acquisition system to reconstruct high-quality eye poses over time, complete with accurate high-resolution eye geometry. We demonstrate that both translation and torsion is clearly visible in the acquired data and hence investigate the importance of modeling these phenomena, along with the correct visual axis, in an eye rig for computer graphics applications.

We believe that the work presented in this thesis on eye reconstruction, eye modeling, and eye rigging has the potential to change how eyes are modeled in computer graphics applications.

1.1 Contributions

This thesis makes the following main contributions:

- *An eyeball reconstruction algorithm* for the coupled reconstructions of sclera, cornea, and iris including the deformation of the iris.
- *A parametric eyeball shape model* created from a database of eyes. This model allows us to generate a wide range of plausible human eyeball shapes by defining only a few shape parameters.
- *A parametric iris model* which generates iris shapes including its deformation. The method requires only a photo or an artist sketch of an iris as input.
- *A parametric vein model* that synthesizes realistic vein networks. The various synthesized vein properties are fed to a renderer that leverages vein samples from an eye database to render a sclera texture.
- *A parametric model fitting algorithm* that allow us to determine the best eye model parameters to match input images and scans. Fitting to a single image is possible.
- *A parametric eye rig* describing the positions and orientations of the eyeballs. The rig can be configured to match a specific person, including parameters for the interocular distance, the center of rotation, and the visual axis.
- *An eye rig fitting algorithm* that estimates the best person-specific rig parameters from a multi-view image sequence.

1.2 Publications

This thesis is based on the following peer-reviewed publications:

- P. BÉRARD, D. BRADLEY, M. NITTI, T. BEELER, and M. GROSS. High-Quality Capture of Eyes. In Proceedings of ACM SIGGRAPH Asia (Shenzhen, China, December 3-6, 2014). *ACM Transactions on Graphics, Volume 33, Issue 6, Pages 223:1–223:12*.
- P. BÉRARD, D. BRADLEY, M. GROSS, and T. BEELER. Lightweight Eye Capture Using a Parametric Model. In Proceedings of ACM SIGGRAPH (Anaheim, USA, July 24-28, 2016). *ACM Transactions on Graphics, Volume 35, Issue 4, Pages 117:1–117:12*.

The thesis is also based on the following submitted publication:

- P. BÉRARD, D. BRADLEY, M. GROSS, and T. BEELER. Physiologically Accurate Eye Rigging. Submitted to ACM SIGGRAPH (Vancouver, Canada, August 12-16, 2018).

During the course of this thesis, the following peer-reviewed papers were published, which are not directly related to the presented work:

- F. ZÜND, P. BÉRARD, A. CHAPIRO, S. SCHMID, M. RYFFEL, M. GROSS, A. BERMANO, and R. SUMNER. Unfolding the 8-bit era. In Proceedings of the 12th European Conference on Visual Media Production (CVMP) (London, UK, November 24-25, 2015). *Pages 9:1–9:10*.

CHAPTER 2

Related Work

The eye is an important part of the human and this is reflected by the wide spectrum of work related to eyes in various disciplines such as medicine, psychology, philosophy, and computer graphics. The requirements for applications in computer graphics are however very different from other fields. This chapter presents some of the related works that are relevant to digital humans and computer graphics in general.

The amount of work related to reconstructing and modeling the human eye is within limits. Our work is related to medical instruments, and facial capture methods, so we also provide a brief overview of these techniques, followed by a description of other methods that are related to our approach at a lower level. Specifically, the algorithms presented in this thesis touch on various fields including non-rigid alignment to find correspondences between eye meshes, data driven fitting to adjust an eye model to a given eye mesh, as well as constrained texture and geometry synthesis to create iris details.

These methods focus on modeling shape and appearance of eyes, which provides a great starting point to our rigging and tracking work, which 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 Reconstruction and Modeling

Reconstructing and modeling eye geometry and appearance have so far received only very little attention in the graphics community, as a recent sur-

vey of Ruhland et al. [2014] shows. Most research so far has focused solely on acquiring the iris, the most prominent part of an eye, typically only considering the color variation and neglecting its shape. An exception is the seminal work by François et al. [2009], which proposes to estimate the shape based on the color variation. Guided by the physiology of the iris, they develop a bright-is-deep model to hallucinate the microscopic details. While impressive and simple, the results are not physically correct and they have to manually remove spots from the iris, since these do not conform with their model. Lam et al. [2006] propose a biophysically-based light transport model to simulate the light scattering and absorption processes occurring within the iridal tissues for image synthesis applications, whereas Lefohn et al. [2003] mimic an ocularist's workflow, where different layers of paint are applied to reproduce the look of an iris from a photograph. Their method is tailored to manufacture eye prosthetics, and only considers the synthesis of the iris color, neglecting its shape.

One of the first to model the entire eye were Sagar et al. [1994], who model a complete eye including the surrounding face for use in a surgical simulator. However, the model is not based on captured data and only approximates the shape of a real eye. More recently, Wood et al. [2016a] presented a parametric eyeball model and a 3D morphable model of the eye region and then fit the models to images using analysis-by-synthesis.

While there has been a substantial amount of research regarding the reconstruction of shape of various materials [Seitz et al., 2006; Ihrke et al., 2008; Hernández et al., 2008], none of these methods seem particularly suited to reconstruct the heterogeneous combination of materials present in the eye. As the individual components of the eye are all coupled, they require a unified reconstruction framework, which is what we propose in this thesis.

2.2 Iris Deformation

Other authors have looked into the motion patterns of the iris, such as dilation or hippus[3] [Hachol et al., 2007]. Pamplona and colleagues study the deformation of the iris when the pupil dilates in 2D [Pamplona et al., 2009]. They manually annotate a sparse set of features on a sequence of images taken while the pupil dilates. The recovered tracks show that the individual structures present in the iris prevent it from dilating purely radially on linear trajectories. Our method tracks the deformation of the iris densely since we

[3] A rhythmic but irregular continuous change of pupil dilation.

do not require manual annotation and our measurements confirm these findings. More importantly, we capture the full three-dimensional deformation of the iris, which conveys the detailed shape changes during pupil dilation. In one of our proposed applications we complement our deformation model with the temporal model proposed by Pamplona et al. [2009].

More importantly, we do capture the full three dimensional motion, which not only conveys how the shape of the iris changes during dilation but also shows that the iris moves on the curved surface of the underlying lens. As we demonstrate in this thesis, the lens changes its shape for accommodation and as a consequence, the shape of the iris is a function of both dilation and accommodation - a feature not considered in our community so far.

2.3 Medical Instruments

In the medical community the situation is different. There, accurate eye measurements are fundamental, and thus several studies exist. These either analyze the eye ex-vivo [Eagle Jr, 1988] or employ dedicated devices such as MRI to acquire the eye shape [Atchison et al., 2004] and slit lamps or keratography for the cornea [Vivino et al., 1993]. Optical coherence tomography (OCT) [Huang et al., 1991], in ophthalmology mostly employed to image the retina, can also be used to acquire the shape of cornea and iris at high accuracy. An overview of the current corneal assessment methods can be found in recent surveys [Rio-Cristobal and Martin, 2014; Piñero, 2013]. Such devices however are not readily available and the data they produce is oftentimes less suited for graphics applications. We therefore chose to construct our own setup using commodity hardware and employ passive and active photogrammetry methods for the reconstruction.

2.4 Facial Capture

Unlike eye reconstruction, the area of facial performance capture has received a lot of attention over the past decades, with a clear trend towards more lightweight and less constrained acquisition setups. The use of passive multi-view stereo [Beeler et al., 2010; Bradley et al., 2010; Beeler et al., 2011] has greatly reduced the hardware complexity and acquisition time required by active systems [Ma et al., 2007; Ghosh et al., 2011; Fyffe et al., 2011]. The amount of cameras employed was subsequently further reduced to binocular [Valgaerts et al., 2012] and finally monocular

acquisition [Blanz and Vetter, 1999; Garrido et al., 2013; Cao et al., 2014; Suwajanakorn et al., 2014; Fyffe et al., 2014].

To overcome the inherent ill-posedness of these lightweight acquisition devices, people usually employ a strong parametric prior to regularize the problem. Following this trend to more lightweight acquisition using parametric priors, we propose to leverage data provided by our high-resolution capture technique and build up a parametric eye-model, which can then be fit to input images acquired from more lightweight setups, such as face scanners, monocular cameras or even from artistically created images.

2.5 Non-Rigid Alignment

A vast amount of work has been performed in the area of non-rigid alignment, ranging from alignment of rigid object scans with low-frequency warps, noise, and incomplete data [Ikemoto et al., 2003; Haehnel et al., 2003; Brown and Rusinkiewicz, 2004; Amberg et al., 2007; Li et al., 2008] to algorithms that find shape matches in a database [Kazhdan et al., 2004; Funkhouser et al., 2004]. Another class of algorithms registers a set of different meshes that all have the same overall structure, like a face or a human body, with a template-based approach [Blanz and Vetter, 1999; Allen et al., 2003; Anguelov et al., 2005; Vlasic et al., 2005]. In this work we use a variant of the non-rigid registration algorithm of Li et al. [2008] in order to align multiple reconstructed eyes and build a deformable eye model [Blanz and Vetter, 1999]. Although Li et al.'s method is designed for aligning a mesh to depth scans, we will show how to re-formulate the problem in the context of eyes, operating in a spherical domain rather than the 2D domain of depth scans.

2.6 Texture and Geometry Synthesis

In this work, texture synthesis is used to generate realistic and detailed iris textures and also geometry from low-resolution input images. A very broad overview of related work on texture synthesis is presented in the survey of Wei et al [2009]. Specific topics relevant for our work include constrained texture synthesis [Ramanarayanan and Bala, 2007] and example-based image super resolution [Tai et al., 2010], which both aim to produce a higher resolution output of an input image given exemplars. With patch-based synthesis methods [Praun et al., 2000; Liang et al., 2001; Efros and

Freeman, 2001], controlled upscaling can be achieved easily by constraining each output patch to a smaller patch from the low-resolution input. These algorithms sequentially copy patches from the exemplars to the output texture. They were further refined with graph cuts, blending, deformation, and optimization for improved patch-boundaries [Kwatra et al., 2003; Mohammed et al., 2009; Chen et al., 2013]. Dedicated geometry synthesis algorithms also exist [Wei et al., 2009], however geometry can often be expressed as a texture and conventional texture synthesis algorithms can be applied. In our work we take inspiration from Li et al. [2015], who propose to use gradient texture and height map pairs as exemplars where in their work the height map encodes facial wrinkles. We expand on their method and propose to encode color, geometry and also shape deformation in a planar parameterization, allowing us to jointly synthesize texture, shape and deformation to produce realistic irises that allow dynamic pupil dilation.

2.7 Eye Tracking and Gaze Estimation

The first methods for photographic eye tracking date back over 100 years [Dodge and Cline, 1901; Judd et al., 1905], and since then dozens of tracking techniques have emerged, including the introduction of head-mounted eye trackers [Hartridge and Thompson, 1948; Mackworth and Thomas, 1962]. We refer to detailed surveys on historical and more modern eye recording devices [Collewijn, 1999; Eggert, 2007]. Such devices have been widely utilized in human-computer interaction applications. Some examples were to study the usability of new interfaces [Benel et al., 1991], to use gaze as a means to reduce rendering costs [Levoy and Whitaker, 1990], or as a direct input pointing device [Zhai et al., 1999]. 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 [Hansen and Ji, 2010] 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 [Sugano et al., 2014; Zhang et al., 2015], which can be created synthetically through realistic rendering [Wood et al., 2015; Wood et al., 2016b]. Another approach is model-fitting, for example Wood et al. [2016a] 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 [Wang et al., 2016] or from RGBD camera input [Wen et al., 2017b] including an extension to eyelids [Wen et al., 2017a]. However, these techniques use rather simple eye rigs and do not consider ophthalmological studies for modeling the true motion patterns of eyes, which is the focus of our work.

2.8 Eye Rigging and Animation

Eye animation is of central importance for the creation of realistic virtual characters, and many researchers have studied this topic [Ruhland et al., 2014]. On the one hand, some of the research explores the coupling of eye animation and head motion [Pejsa et al., 2016; Ma and Deng, 2009] or speech [Zoric et al., 2011; Le et al., 2012; Marsella et al., 2013], where other work focuses on gaze patterns [Chopra-Khullar and Badler, 2001; Vertegaal et al., 2001], statistical movement models for saccades [Lee et al., 2002], or synthesizing new eye motion from examples [Deng et al., 2005]. 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. Generally speaking, a common assumption is that an eye is comprised of a spherical shape, rotating about its center, with the gaze direction corresponding to the optical axis, which is the vector from the sphere center through the pupil center [Itti et al., 2003; Pinskiy and Miller, 2009; Weissenfeld et al., 2010; Wood et al., 2016a; Pejsa et al., 2016] (Fig. 1.4 (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 several models from the field of ophthalmology can improve the realism of eye animation in computer graphics.

CHAPTER 3

Eye Anatomy

In this chapter we provide an overview of the anatomy of the human eye viewed through the lens of computer graphics. Medical books [Hogan et al., 1971] describe it in much greater detail, but in this chapter we want to summarize what is relevant to this thesis and to computer graphics in general.

The human eye consists of several different parts as shown in Fig. 3.1. The white sclera and the transparent cornea define the overall shape of the eyeball. The colored iris, located behind the cornea, acts like a diaphragm controlling the light going through the pupil at the center of the iris, and behind the iris is the lens. It focuses the light and forms an image at the back of the eyeball on the retina. The eyeball is connected to muscles that control its position and orientation.

In the following sections we will provide more details about each individual part of the eye.

3.1 Eyeball

The eyeball is the rigid and hard part of the eye. It is located inside the eye socket that holds the eye in place with muscles as shown in Fig. 3.1 a. The spherical eyeball shape allows for smooth rotations, however, its shape is not perfectly spherical. The transparent cornea protrudes from the spherical shape. Besides the cornea the front part of the eyeball is flatter towards the nose and rounder towards the outer side of the face as depicted in Fig. 4.11.

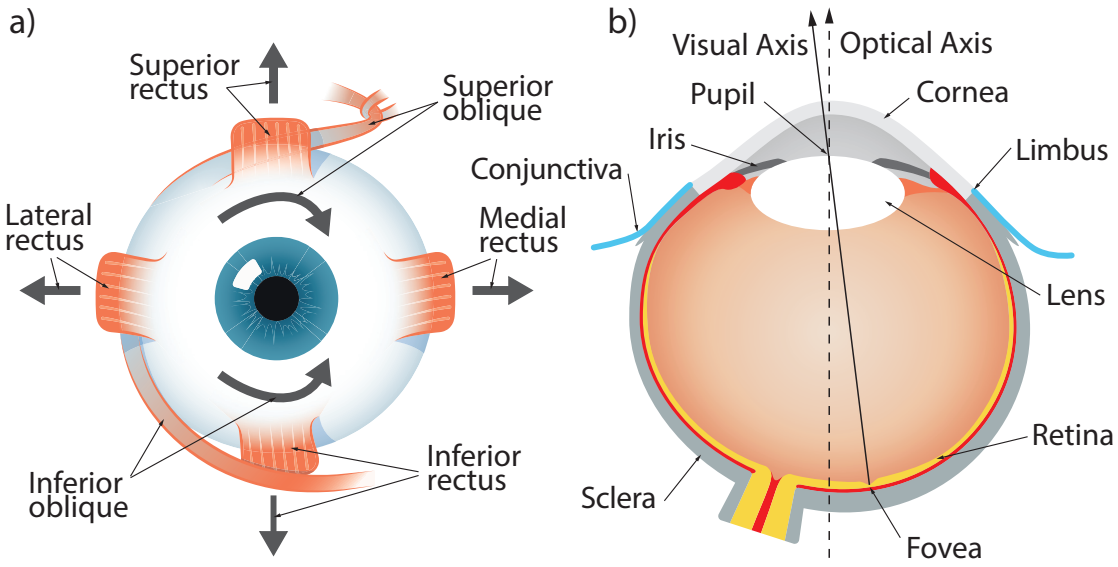

Figure 3.1: *Anatomy: a) The eye is controlled by six muscles (two per degree of freedom), which operate in an complex orchestrated way to rotate the eye. b) The eye consists of different parts with different visual and optical properties. The cornea, the limbus, and the sclera are rigid, whereas the iris, the pupil, the conjunctiva, and the lens can deform. 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.*

Nevertheless, the eyeball shape is often approximated with two spheres, one for the sclera and one for the cornea. The radius of the main sphere representing the sclera is about 11.5 mm and the cornea is modeled with a smaller sphere with a radius of about 7.8 mm. The mean axial length of a human eye is about 24 mm as reported by Hogan et al. [1971]. The axial length is also affected by medical conditions like *myopia* or *hyperopia*. This means that the axial length is either too long or too short to properly focus the light onto the retina, requiring the people with these conditions to wear glasses. Given theses spherical eyeball assumptions it is also very common to define the rotation center of the eyeball at the center of the sphere defining the sclera part of the eyeball.

The eyeball can be subdivided into different parts that all have different appearance and optical properties. The outer layer of the eyeball consists of two parts: the sclera and the cornea that are described in the following sections.

3.2 Sclera and Conjunctiva

The sclera and the conjunctiva (Fig. 3.1) make up the white part of the eyeball. The sclera is part of the rigid eyeball whereas the conjunctiva is connected to the eyeball near the limbus and to the eye socket. This thin layer covers thus the visible part of the sclera and moves freely on top of it as shown in Fig. 6.7. It can be stretched and compressed leading to folds in the conjunctiva that result in characteristic reflections following these folds.

Both the conjunctiva and the sclera contain blood vessels. These blood vessels are visible since the sclera and the conjunctiva are translucent and not fully opaque. This also means that light scatters inside the sclera and the conjunctiva and makes them visually very soft. If eyes are rendered without taking this scattering into account the rendered eyes will look very hard and unnatural.

The blood vessels can be at different depths and have different sizes and carry varying amounts of oxygen, all affecting the appearance of the blood vessel. Also, in general, the color of the vessels in the conjunctiva is more intense than the color of the vessels in the sclera since the latter are covered by the conjunctiva. Another factor affecting the color of these vessels is the emotional state of the person. A sad or an angry person might have more pronounced and redder vessels.

3.3 Cornea

The cornea (Fig. 3.1) is the transparent part of the eyeball and is surrounded by the sclera. The cornea is not perfectly transparent and reflects a part of the incident light. This leads to visible reflections of bright light sources like lamps and windows. The cornea is also not a homogeneous medium, but it consists of multiple layers and each layer reflects a fraction of the incident light, which results in one main and multiple weaker glints. In contrast to the conjunctiva, the cornea is completely smooth, which is important to guarantee the optical properties of the cornea. This also results in very sharp reflections on the cornea which can be leveraged by environment map creation [Nishino and Nayar, 2004] and eye tracking [Wang et al., 2015] algorithms.

Also, since each layer has a slightly different index of refraction, the light traversing the cornea will be refracted multiple times. Since the difference in index of refraction between the air and the first cornea layer is the biggest, the refraction is the strongest at this first interface and the refraction taking

place at the other interfaces can often be neglected. In this thesis we will simplify the cornea and approximate it with a single homogeneous medium.

Structurally, the cornea and the sclera are very similar. However, while one is transparent the other has an opaque white color. Even though they both consist of a similar composition of collagen fibers. The reason for the different optical properties lies in the arrangement of theses fibers. The regular alignment of the collagen fibers in the cornea leads to transparency. Whereas the random alignment of the fibers in the sclera scatters the light and makes the sclera white.

3.4 Limbus

The transition region from the sclera to the cornea is called the limbus (Fig. 3.1). Viewed from the front it is not a perfect circle, but it is usually a bit wider than high. Hogan et al. [1971] report mean dimensions of 11.7 mm for the width and 10.6 mm for the height.

The limbus is not an abrupt interface, but expands over a few millimeters due to a gradual internal change in structure. Besides the transition in composition the sclera geometrically clamps the cornea, further contribution to the smooth transition. In photographs the limbus can appear as a hard interface or it can expand over a larger region as shown in Fig. 3.2. The limbus also contains a blood vessel network that is well visible in the almost transparent part of the limbus.

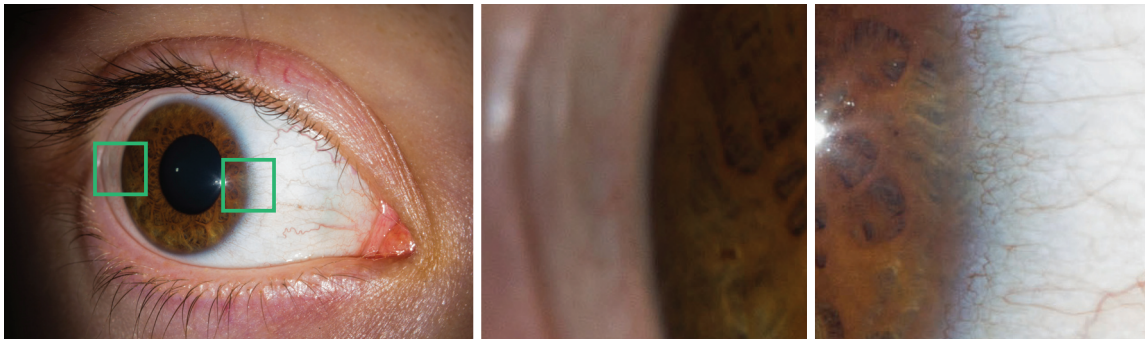

Figure 3.2: *The appearance of the limbus in a photograph depends on the width of the limbus and the viewing direction. The insets show the limbus as well as the limbal vessel network*

3.5 Iris

The iris is located behind the cornea and the limbus, but in front of the lens (Fig. 3.1). It is responsible to control the amount of light that hits the retina. It does so by contracting and dilating the pupil at its center.

The iris has a fibrous structure with craters called *crypts*. To contract and dilate the pupil the iris has a sphincter muscle (Fig. 3.3) around the pupil that contracts the iris and radial muscles that open the iris again. These deformations lead to radial and circular folds on the iris.

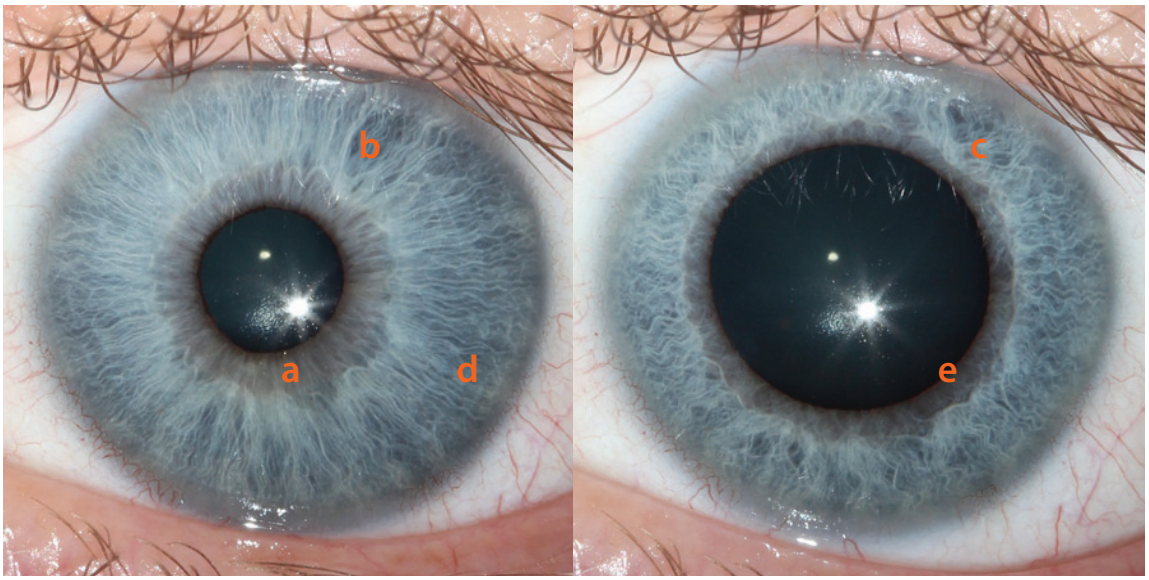

Figure 3.3: *A blue iris in contracted state (left) and dilated state (right) with visible sphincter muscle (a), radial folds (b,c), circular fold marks (d), and the dark rim (e).*

The color of the iris is a combination of blue, green, and brown hues. A strict classification of iris colors is difficult, but several authors define classification systems with about ten classes [Mackey et al., 2011]. The composition of the iris defines its color. For example the amount of melanin is responsible for the brown color of the iris. Another factor affecting the appearance is the environment light, which can make eyes very dull or make them stand out.

The edge of the iris usually has a fine brown or black pigmented rim. This rim makes the transition to the pupil visually very soft.

Also, the iris is not a rigid object and it wobbles due to its inertia if the eye moves very fast and then stops abruptly.

3.6 Pupil

The pupil (Fig. 3.1) is the opening at the center of the iris and controls the amount of light entering the eye. The pupil is not exactly at the center of the iris and this center can even shift during contraction and dilation.

Through contraction and dilation the pupil adjusts it size constantly to account for the amount of environment light (*direct response*). But there are other factors affecting its size. Due to the *accommodation reflex* the pupil contracts when looking at a close object to guarantee the best possible sharpness. Also, the pupil of the right and the left eye react in a coordinated way (*consensual response*). Thus, if light is shone into one eye, the pupil of the other eye will contract as well. This phenomenon is leveraged in Chapter 4 of this thesis.

Visually, the pupil is almost never pitch black in a photograph. Light is reflected on the back of the eye and makes the pupil appear in a shade of gray. If light is projected co-axially to the view axis the pupil becomes very bright, since the light is directly reflected off the back of the eyeball. This effect in combination with infrared light is employed by various pupil detection algorithms.

3.7 Muscles

The muscles are responsible to orient the eyeball within the eye socket. There are six muscles per eye (Fig. 3.1), which can be grouped in three pairs: *superior rectus/inferior rectus*, *lateral rectus/medial rectus*, and *superior oblique/inferior oblique*. These six muscles move the eye in an orchestrated way. The muscle have multiple functions depending on the current eyeball pose. If the eye is in the neutral position (looking straight ahead) the *superior rectus* is the muscle exerting the primary action responsible for looking up. If however the eye is adducted (eye moving nasally) the *inferior oblique* becomes the primary muscle for looking up. For a more detailed analysis of the functions of the individual muscles we refer to the medical literature [Hogan et al., 1971].

CHAPTER 4

Eye Reconstruction

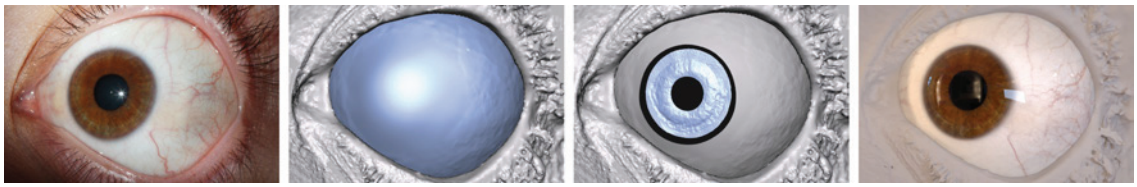

Figure 4.1: *We present a system to acquire the shape and texture of an eye at very high resolution. This figure shows one of the input images, the reconstructed eyeball and iris geometry, and a final render from a novel viewpoint under different illumination (left to right).*

The creation of digital humans for the use in animation requires a pipeline with several components inluding eye reconstruction, modeling , and rigging. In this chapter we introduce a system for the reconstruction of eyes for digital humans. In Chapter 5 and Chapter 6 we show how this eye reconstruction system can be leveraged to model and rig eyes.

The complexity of human eyes dictates a novel approach for capture and accurate reconstruction. We must pay particular attention to the appearance properties of the different components of the eye, and design different strategies for reconstructing each component. While it is possible to assume that the sclera is diffuse and Lambertian (such as often assumed for skin), the cornea is completely transparent, and the iris is viewed under unknown distortion due to refraction. Furthermore, there is a coupling of the eye components, for example the corneal shape should transition smoothly to the sclera, and the perceived iris position depends on both the corneal shape as

well as the exact index of refraction (both of which *do* vary from person to person).

The above observations lead to a progressive algorithm for eye reconstruction. We start by recovering the sclera shape, followed by the cornea, and finally the iris. Each stage of the reconstruction requires a different approach, relying on constraints from the previous stages but tuned to the appearance properties at hand. The various reconstruction methods also require different (but complementary) capture data, which we acquire through a novel hardware setup of cameras, flashes and LED lights.

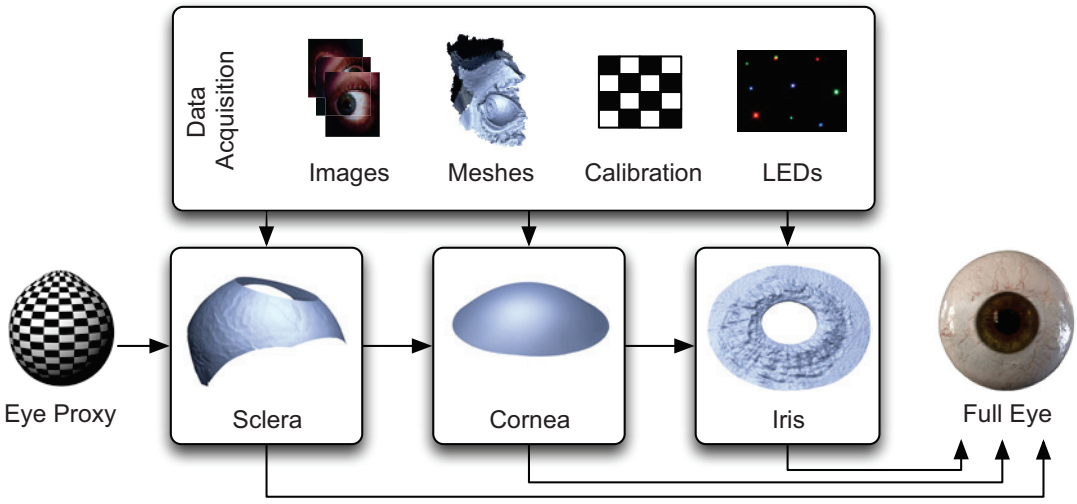

Figure 4.2: *This figure shows an overview of the system. First, several modalities of data are acquired (Section 4.1). From these plus a generic eye proxy, the system reconstructs the individual components of the eye, the sclera (Section 4.2), the cornea (Section 4.3), and the iris (Section 4.4) and combines them into a complete eye model.*

To describe our method in detail, we organize this chapter as illustrated in Fig. 4.2. Section 4.1 explains the data acquisition phase including the capture hardware. Section 4.2 discusses our passive multi-view, multi-pose reconstruction method for obtaining the sclera. Given the approximate sclera shape, we design a photometric approach for computing the corneal shape given a set of known LED lights in the scene and multiple views of the refracted iris (Section 4.3). The iris itself is then reconstructed using a novel multi-view stereo approach that traces light paths through the corneal interface (Section 4.4). Irises are reconstructed for a sequence of different pupil dilations and we recover a deformable model for iris animation, parameterized by pupil radius. Our results demonstrate that each individual eye is unique in many ways, and that our reconstruction algorithm is able to

capture the main characteristics required for rendering digital doubles (Section 4.5).

4.1 Data Acquisition

The first challenge in eye reconstruction is obtaining high-quality imagery of the eye. Human eyes are small, mostly occluded by the face, and have complex appearance properties. Additionally, it is difficult for a subject to keep their eye position fixed for extended periods of time. All of this makes capture challenging, and for these reasons we have designed a novel acquisition setup, and we image the eye with variation in gaze, focus and pupil dilation.

4.1.1 Capture Setup

Our capture setup consists of multiple cameras, a modified flash for primary illumination, and a variety of colored LEDs that will reflect off the cornea. To help the subject remain still during acquisition, we arrange the setup such that they can lie on the floor with their head in a headrest, situated under the camera array (Fig. 4.3).

To get the best coverage in the space available, we place six cameras (Canon 650D) in a 2 by 3 configuration, with 100mm macro lenses focused on the iris. The lens is stepped down to $f11$ and the camera is set to ISO100. The exposure is set to 1 second since we capture in a dark room and the flash provides the primary illumination. The main flash light consist of three elements: a conventional flash (Canon 600EX-RT), a cardboard aperture mask and a lens. This assembly allows us to intensify and control the shape of the light so that reflections of the face and the eyelashes can be prevented as much as possible. We use 9 RGB LEDs and arrange them in a 3x3 pattern, ensuring that similar colors are not adjacent in order to maximize our ability to uniquely detect their reflections on the cornea. The pupil dilation is controlled with a high-power LED with adjustable brightness. We place this LED close to the eye that is *not* being captured. Since the pupil dilation of both eyes is linked we can control the dilation of the captured eye indirectly, avoiding an extra specular highlight on the captured eye. In order to measure the eye focusing at different depths, a focus pole with specifically marked distances is placed in front of the subject. Finally, additional studio lamps are used during camera calibration.

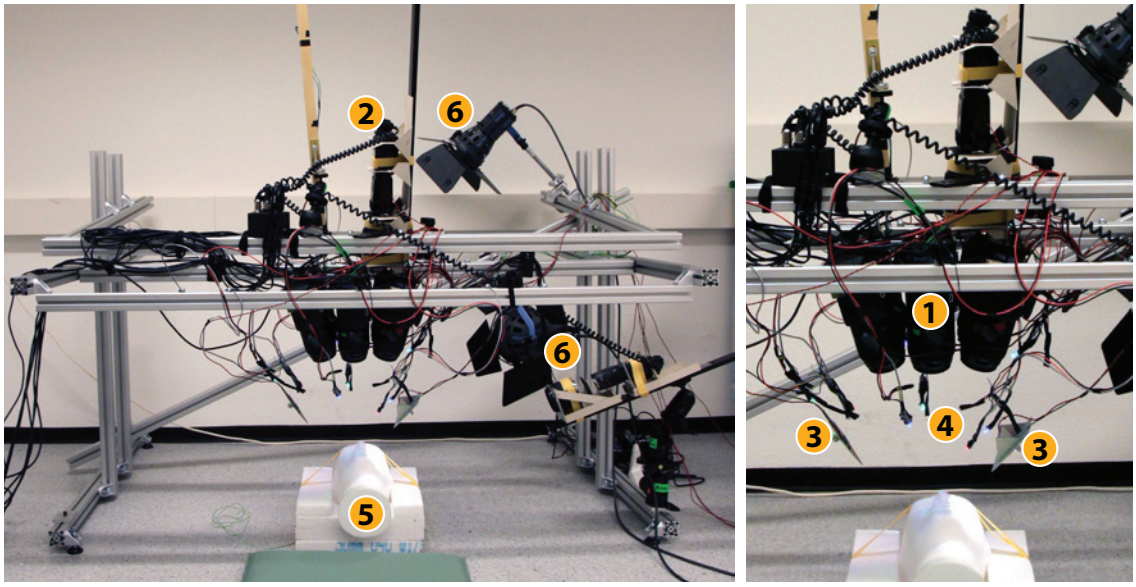

Figure 4.3: *Overview of the capture setup consisting of a camera array (1), a focused flash light (2), two high-power white LEDs (3) used to control the pupil dilation, and color LEDs (4) that produce highlights on the cornea. The subject is positioned in a headrest (5). The studio lamps (6) are used during camera calibration.*

4.1.2 Calibration

Cameras are calibrated using a checkerboard of CALTag markers [Atcheson et al., 2010], which is acquired in approximately 15 positions throughout the capture volume. We calibrate the positions of the LEDs by imaging a mirrored sphere, which is also placed at several locations in the scene, close to where the eyeball is during acquisition. The highlights of the LEDs on the sphere are detected in each image by first applying a Difference-of-Gaussian filter followed by a non-maximum suppression operator, resulting in single pixels marking the positions of the highlights. The detected highlight positions from a specific LED in the different cameras form rays that should all intersect at the 3D position of that LED after reflection on the sphere with known radius (15mm). Thus, we can formulate a nonlinear optimization problem where the residuals are the distances between the reflected rays and the position estimates of the LEDs. We solve for the unknown LED and sphere positions with the Levenberg-Marquardt algorithm.

4.1.3 Image Acquisition

We wish to reconstruct as much of the visible eye as possible, so the subject is asked to open their eyes very wide. Even then, much of the sclera is occluded in any single view, so we acquire a series of images that contain a variety of eye poses, covering the possible gaze directions. Specifically we used 11 poses: *straight*, *left*, *left-up*, *up*, *right-up*, *right*, *right-down*, *down*, *left-down*, *far-left*, and *far-right*. The *straight* pose will be used as reference pose, as it neighbors all other poses except *far-left* and *far-right*.

We then acquire a second series of images, this time varying the pupil dilation. The intricate geometry of the iris deforms non-rigidly as the iris dilator muscle contracts and expands to open and close the pupil. The dilation is very person-specific, so we explicitly capture different amounts of dilation for each actor by gradually increasing the brightness of the high-power LED. In practice, we found that a series of 10 images was sufficient to capture the iris deformation parametrized by pupil dilation.

The acquisition of a complete data set takes approximately 5 minutes for positioning the hardware, 10 minutes for image acquisition, and 5 minutes for calibration, during which time the subject lies comfortably on a cushion placed on the floor.

4.1.4 Initial Reconstruction

To initialize our eye capture method, we pre-compute partial reconstructions for each eye gaze using the facial scanning technique of Beeler et al. [2010]. Although this reconstruction method is designed for skin, the sclera region of the eye is similarly diffuse, and so partial sclera geometry is obtainable. These per-gaze reconstructions will be used in later stages of the pipeline. Additionally, the surrounding facial geometry that is visible will be used for providing context when rendering the eye in Section 4.5.

4.2 Sclera

Reconstructing the sclera is challenging because large parts are occluded by the eyelids and the eye socket at any given time. As indicated previously, the problem can be alleviated by acquiring the eye under multiple poses. In this section we explain our approach to register the different poses into a common frame and integrate the partial scans into a complete model of the eyeball. The individual steps are outlined in Fig. 4.4.

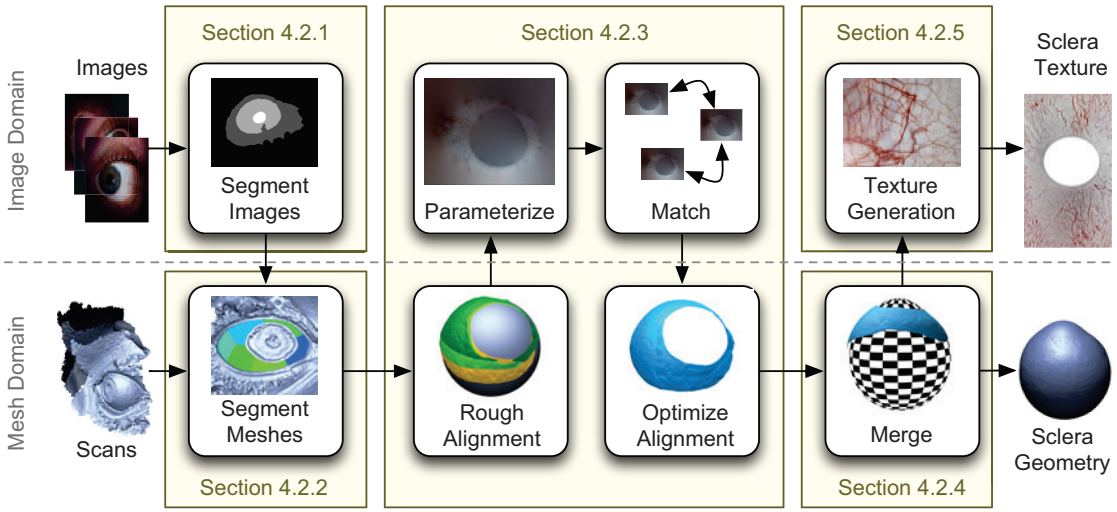

Figure 4.4: *The sclera reconstruction operates in both image and mesh domains. The input images and meshes are segmented (Section 4.2.1 and Section 4.2.2). The partial scans from several eye poses are registered (Section 4.2.3) and combined into a single model of the sclera using a generic proxy (Section 4.2.4). A high-resolution texture of the sclera is acquired and extended via texture synthesis (Section 4.2.5).*

4.2.1 Image Segmentation

The individual components of the eye require dedicated treatment, and thus the first step is to segment the input images to identify skin, sclera, iris, and pupil regions. We acquire approximately 140 images for a single eye dataset, considering all the poses, pupil dilations and multiple cameras, which would make manual segmentation tedious. Therefore, a semi-supervised method is proposed to automate the process. All images are captured under similar conditions, and thus the appearance of the individual classes can be expected to remain similar. We therefore employ a nearest-neighbor classification. We manually segment one of the images into skin, sclera, iris and pupil regions (Fig. 4.5a). These serve as examples, from which the algorithm labels the pixels of the other images automatically by assigning the label of the most similar example pixel. Similarity is computed in a lifted 21 dimensional feature space of 15 color and 6 Haralick texture features [Haralick, 1979], and has proven to provide sufficiently accurate and robust results. This classification is fast since every pixel is treated independently. We obtain high quality classification by employing a post-processing step that uses the following topological rules:

- The iris is the largest connected component of iris pixels.
- There is only a single pupil and the pupil is inside the iris.
- The sclera part(s) are directly adjacent to the iris.

Fig. 4.5b shows the final classification results for a subset of images, based on the manually annotated exemplar shown in (a).

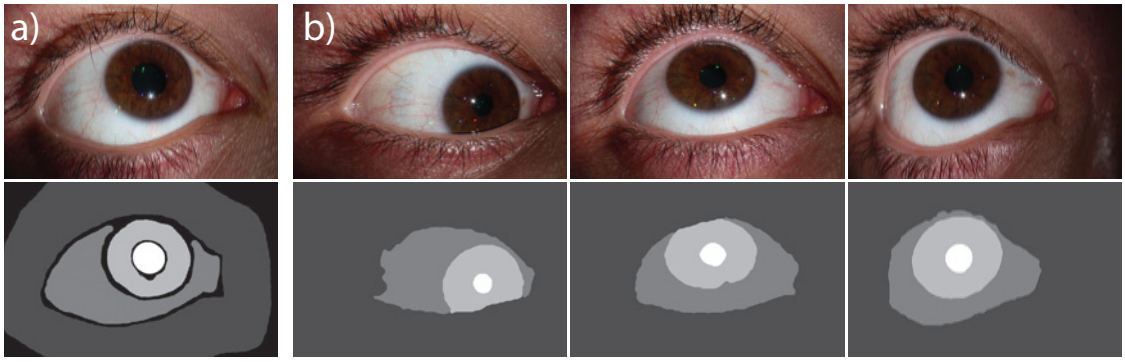

Figure 4.5: *Pupil, iris, sclera, and skin classification with manual labels (a) and examples of automatically labeled images (b).*

4.2.2 Mesh Segmentation

Given the image-based classification, we wish to extract the geometry of the sclera from the initial mesh reconstructions from Section 4.1.4. While the geometry is mostly accurate, the interface to the iris and skin may contain artifacts or exhibit over-smoothing, both of which are unwanted properties that we remove as follows.

While a single sphere only poorly approximates the shape of the eyeball globally (refer to Fig. 4.11 in the results), locally the surface of the sclera may be approximated sufficiently well. We thus over-segment the sclera mesh into clusters of about 50mm^2 using k-means and fit a sphere with a 12.5mm radius (radius of the average eye) to each cluster. We then prune vertices that do not conform with the estimated spheres, either in that they are too far off surface or their normal deviates strongly from the normal of the sphere. We found empirically that a distance threshold of 0.3mm and normal threshold of 10 degrees provide good results in practice and we use these values for all examples in this chapter. We iterate these steps of clustering, sphere fitting, and pruning until convergence, which is typically reached in less than 5 iterations. The result is a set of partial sclera meshes, one for each captured gaze direction.

4.2.3 Pose Registration

The poses are captured with different gaze directions *and* slightly different head positions, since it is difficult for the subject to remain perfectly still, even in the custom acquisition setup. To combine the partial sclera meshes into a single model, we must recover their rigid transformation with respect to the reference pose. ICP [Besl and McKay, 1992] or other mesh-based alignment methods perform poorly due to the lack of mid-frequency geometric detail of the sclera. Feature-based methods like SIFT, FAST, etc. fail to extract reliable feature correspondences because the image consists mainly of edge-like structures instead of point-like or corner-like structures required by the aforementioned algorithms. Instead, we rely on optical flow [Brox et al., 2004] to compute dense pairwise correspondences.

Optical flow is an image based technique and typically only reliable on small displacements. We therefore align the poses first using the gaze direction and then parameterize the individual meshes jointly to a uv-plane. The correspondences provided by the flow are then employed to compute the rigid transformations of the individual meshes with respect to the reference pose. These steps are iterated, and convergence is typically reached in 4-5 iterations. In the following we will explain the individual steps.

Initial Alignment: The gaze direction is estimated for every pose using the segmented pupil. Since the head does not remain still during acquisition, the pose transformations are estimated by fitting a sphere to the reference mesh and aligning all other meshes so that their gaze directions match.

Joint Parameterization: The aligned meshes are parameterized to a common uv-space using spherical coordinates. Given the uv-parameterization, we compute textures for the individual poses by projecting them onto the image of the camera that is closest to the line of sight of the original pose. This naive texturing approach is sufficient for pose registration, and reduces view-dependent effects that could adversely impact the matching.

Correspondence Matching: We compute optical flow [Brox et al., 2004] of the individual sclera textures using the blue channel only, since it offers the highest contrast between the veins and the white of the sclera. The resulting flow field is sub-sampled to extract 3D correspondence constraints between any two neighboring sclera meshes. We only extract constraints which are both well localized and well matched. Matching quality is assessed using

the normalized cross-correlation (NCC) within a $k \times k$ patch. Localization is directly related to the spatial frequency content present within this patch, quantified by the standard deviation (SD) of the intensity values. We set $k = 21$ pixels, NCC > 0, and SD < 0.015 in all our examples.

Optimization: The rigid transformations of all the poses are jointly optimized with a Levenberg-Marquardt optimizer so that the weighted squared distances between the correspondences are minimized. The weights reflect the local rigidity of the detected correspondences and are computed from the Euclidean residuals that remain when aligning a correspondence plus its 5 neighbors rigidly. The optimization is followed by a single ICP iteration to minimize the perpendicular distances between all the meshes.

4.2.4 Sclera Merging

After registering all partial scans of the sclera, they are combined into a single model of the eyeball. A generic eyeball proxy mesh, sculpted by an artist, is fit to the aligned meshes and the partial scans are merged into a single mesh, which is then combined with the proxy to complete the missing back of the eyeball.

Proxy Fitting: Due to the anatomy of the face, less of the sclera is recovered in the vertical direction and as a result the vertical shape is less constrained. We thus fit the proxy in a two step optimization. In the first step we optimize for uniform and in the second step for horizontal scaling only. In both steps we optimize for translation and rotation of the eyeball while keeping the rotation around the optical axis fixed.

Sclera Merging: The proxy geometry prescribes the topology of the eyeball. For every vertex of the proxy, a ray is cast along its normal and intersected with all sclera meshes. The weighted average position of all intersections along this ray is considered to be the target position for the vertex and the standard deviation of the intersections will serve as a confidence measure. The weights are a function of the distance of the intersection to the border of the mesh patch and provide continuity in the contributions.

Eyeball Merging: The previous step only deforms the proxy where scan data is available. To ensure a smooth eyeball, we propagate the deformation

to the back of the eyeball using a Laplacian deformation framework [Sorkine et al., 2004]. The target vertex positions and confidences found in the previous step are included as weighted soft-constraints. The result is a single eyeball mesh that fits the captured sclera regions including the fine scale details and surface variation, and also smoothly completes the back of the eye.

4.2.5 Sclera Texturing

As a final step, we compute a color for each point on the reconstructed sclera surface by following traditional texture mapping approaches that project the 3D object onto multiple camera images. In our case, we must consider all images for all eye poses and use the computed sclera segmentation to identify occlusion. One approach is to naively choose the most front-facing viewpoint for each surface point, however this leads to visible seams when switching between views. Seams can be avoided by averaging over all views, but this then leads to texture blurring. An alternative is to solve the Poisson equation to combine patches from different views while enforcing the gradient between patches to be zero [Bradley et al., 2010], but this can lead to strong artifacts when neighboring pixels at the seam have high gradients - a situation that often occurs in our case due to the high contrast of a red blood vessel and white sclera. Our solution is to separate the high and low frequency content of the images. We then apply the Poisson patch combination approach only for the low frequency information, which is guaranteed to have low gradients. We use the naive best-view approach for the high frequencies, where seams are less noticeable because most seams come from shading differences and the shading on a smooth eye is low-frequency by nature. After texture mapping, the frequencies are recombined. Fig. 4.6b shows the computed texture map for the eye in Fig. 4.6a.

Our texturing approach will compute a color for each point that was seen by at least one camera, but the occluded points will remain colorless. Depending on the intended application of the eye reconstruction, it is possible that we may require texture at additional regions of the sclera, for example if an artist poses the eye into an extreme gaze direction that reveals part of the sclera that was never observed during capture. For this reason, we synthetically complete the sclera texture, using texture synthesis [Efros and Leung, 1999]. In our setting, we wish to ensure consistency of blood vessels, which should naturally continue from the iris towards the back of the eye. We accomplish this by performing synthesis in Polar coordinates, where most veins traverse consistently in a vertical direction, and we seed the synthesis

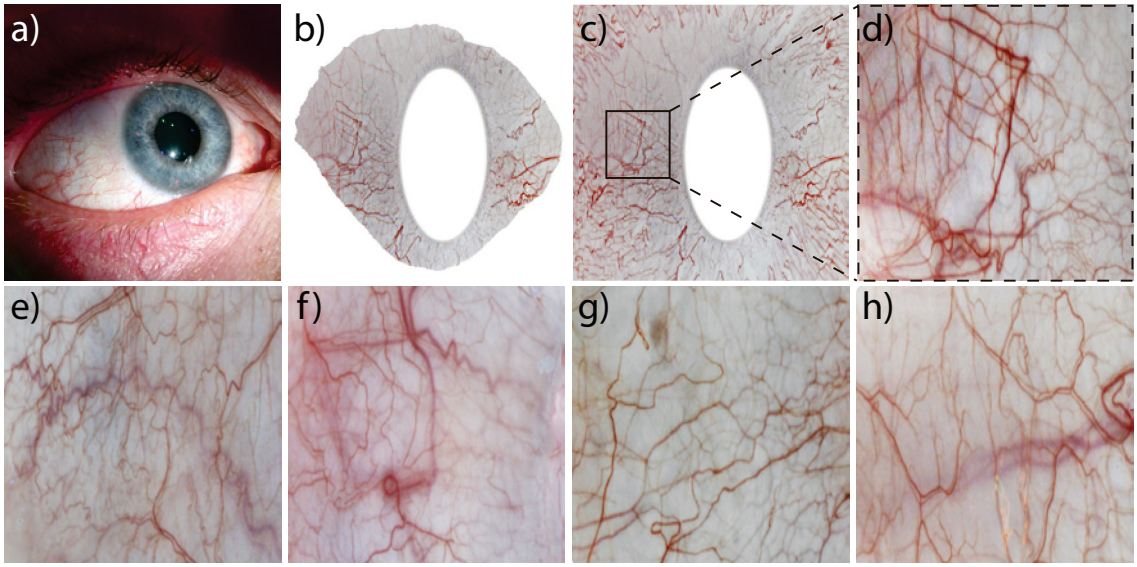

Figure 4.6: *The sclera is textured from multiple views of multiple different eye poses. The resulting texture map (b) for a given eye (a) contains all the visible parts of the sclera. We further complete the texture map through texture synthesis (c). Our textures can have very high resolution details (d-h).*

with a few vertical vein samples. Fig. 4.7 demonstrates the rotated synthesis, which we perform only on the high frequencies in order to avoid synthesized shading artifacts. Corresponding low-frequency content is created by smooth extrapolation of the computed low-frequency texture.

Finally, we can also synthesize missing surface details in the back of the eye. We use the same texture synthesis approach, but instead we operate on a displacement map, which is computed as the difference between the original and a smoothed version of the reconstructed eyeball. The final result is a complete eyeball with continuous texture and displacement at all points. We show a complete texture and zoom region in Fig. 4.6 (c-d), and highlight a few zoom regions of different eye textures in Fig. 4.6 (e-h).

4.3 Cornea

Given the reconstructed sclera, we now describe our technique to reconstruct the transparent cornea. Although the cornea consists of several thin layers with different optical properties, we found it sufficient to model the cornea as a single surface with a single medium respectively index of refraction inside the eye. We use a surface optimization method that aims to satisfy

constraints from features that are either reflected off or refracted through the cornea.

4.3.1 Theory

Reconstructing transparent surfaces requires different approaches than diffuse surface reconstruction since the surface is not directly visible. Transparent surfaces are generally not completely transmissive, but a fraction of light is reflected if the refractive indices of the media involved differ. Thus, a bright light placed in front of the cornea will cause a visible highlight that provides a cue about the surface. Unfortunately, the position of the highlight is view-dependent and cannot directly be used in a mulit-view setting.

On the other hand, for a single view there is an ambiguity between the depth along the viewing ray corresponding to a highlight and the normal of the surface. For every position along a viewing ray there exists a surface normal reflecting the ray to the origin of the light (Fig. 4.8a, green). This creates a surface normal field defined by all possible viewing ray direction and depth combinations. A similar surface normal field is produced from refractions (Fig. 4.8a, red).

The reflection and refraction surface normal fields of different views only coincide at the position of the actual surface as illustrated in Fig. 4.8b. We use this property to reconstruct the cornea.

Our system however produces only a sparse sampling of the normal fields as we employ only a few LEDs. We therefore need to add regularization to ensure a unique solution, which is provided through the chosen surface representation. We employ an open uniform B-spline surface with 100 control points. This surface has more representation power than the traditionally employed 4th order Zernike polynomials [Ares and Royo, 2006; Smolek and Klyce, 2003] yet can be controlled locally, which is beneficial for optimization. The control points are spaced regularly and initialized to the surface of the eyeball proxy introduced in Section 4.2.4. The position of the boundary control points are optimized such that the surface boundaries fit the proxy geometry. The boundary control points are kept fixed and are not part of the following surface optimization.

4.3.2 Constraint Initalization

The corneal surface is optimized using three different types of constraints: reflection, refraction and position constraints.

Reflection Constraints: The 9 calibrated LEDs placed in front of the cornea are imaged as highlights in the different views. From these highlights we extract reflection constraints, which prescribe the normal for any point along the viewing ray through the highlight. Since the cornea is convex, every LED-view pair contributes one constraint assuming the reflection of the LED is visible in the view. In addition, since we registered the poses in Section 4.2.3 we can combine constraints from all different poses. The highlights are detected and identified similarly as in the calibration phase (Section 4.1.2). While the highlights in the calibration images are acquired in complete darkness, now they appear superimposed on the iris in the input images, which can lead to false positive detections. Thus, we remove these unwanted detections by fitting a 2D Gaussian curve to the intensity profiles of all the highlight candidates to determine their width. Since all the LED highlights have a constant size we can remove false positives with a lower (3px) and upper (15px) threshold on the standard deviation of the Gaussian.

Refraction Constraints: Conceptually refraction constraints are very similar to reflection constraints. Instead of observing the reflected highlight of a known LED, we instead observe the refraction of a feature on the iris at unknown position. Furthermore, the angle of refraction depends on the refractive index. Both the position of the feature and the refractive index are included as unknowns into the optimization and solved for. A feature point on the iris contributes one refractive constraint per view. The corresponding image location in the different views is estimated via optical flow [Brox et al., 2004]. Features are filtered as described in Section 4.2.3 using NCC$>$0.6 and SD$<$0.02.

As for reflection constraints, we can combine refraction constraints from all poses. The distribution density of the features varies substantially, as we wont have any in the pupil for example. To account for this we weigh the constraints by the local density, approximated by the distance d to the 10th nearest constraint as $w^{refr} = NCC/d^2$ where NCC is the average normalized cross correlation score between corresponding image patches used as a measurement of the quality of the constraint.

Position Constraints: Position constraints are extracted from the merged sclera mesh (Section 4.2.4). Their purpose is to provide a continuous transition from the cornea to the sclera. We randomly sample position constraints on the sclera in the vicinity of the corneal boundary. To ensure a good distribution, we reject constraints that are closer than 1mm to each other.

4.3.3 Surface Reconstruction

With a given set of reflection, refraction and position constraints and an initial guess of the surface, the unknown parameters are optimized with a two stage approach. More specifically, we optimize the control points of the B-Spline, the refractive index and the unknown positions of the feature points on the iris which are used for the refraction constraints. This amounts to a non-linear optimization which we solve using the Levenberg-Marquardt algorithm by minimizing the error

$$E^{tot} = \lambda^{pos} E^{pos} + \lambda^{refl} E^{refl} + \lambda^{refr} E^{refr}, \tag{4.1}$$

where $\lambda^{pos} = 0.1$, $\lambda^{refl} = 1$, and $\lambda^{refr} = 1$ are user-defined parameters. The error for the position constraints $\mathcal{P}$ is given as

$$E^{pos} = \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \left\| \mathbf{p}_i - \mathbf{p}_i^{pos} \right\|^2, \tag{4.2}$$

where $\mathbf{p}^{pos}$ denotes the position of the constraint and $\mathbf{p}$ the nearest point on the corneal surface. The error for the reflection constraints $\mathcal{Q}$ is given as

$$E^{refl} = \frac{1}{|\mathcal{Q}|} \sum_{i \in \mathcal{Q}} \left\| \mathbf{n}_i - \mathbf{n}_i^{refl} \right\|^2, \tag{4.3}$$

where $\mathbf{n}$ is the current and $\mathbf{n}^{refl}$ the targeted surface normal. The error for the refraction constraints $\mathcal{R}$ is given as

$$E^{refr} = \frac{1}{|\mathcal{R}|} \sum_{i \in \mathcal{R}} w_i^{refr} \left\| \mathbf{p}_i^{iris} - \mathbf{p}_i^{refr} \right\|^2, \tag{4.4}$$

where $\mathbf{p}^{iris}$ is the point on the iris, $\mathbf{p}^{refr}$ the closest point on the refracted ray and w^{refr} its corresponding weight. Optimizing the distance to the closest point has proven to be more stable than optimizing the mismatch of the normals analogously to Equation 4.3.

In the first step we optimize for the control point positions of the B-spline surface. They are optimized only along the opical axis of the eye and the boundary control points are kept fixed at all times. After convergence the surface is kept fixed and we optimize for the refraction constraint points on the iris ($\mathbf{p}^{iris}$) and the refractive index. We iterate by alternating the two steps until the overall improvement drops below $10e^{-10}$. The initial and optimized corneal surface plus constraints are visualized in Fig. 4.9 for one dataset.

4.3.4 Cornea-Eyeball Merging

We update the eyeball mesh with the optimized cornea by smoothly blending the corneal surface into the eyeball mesh. First, corneal samples are computed for each eyeball vertex by intersecting the cornea in the direction of the eyeball normals. Second, the iris masks are dilated, blurred, projected onto the cornea, and averaged to compute blending weights. Finally, the eyeball vertices are combined with the corneal samples by weighting them with the weights.

4.4 Iris

We now move to the final component of the eye, the iris. In contrast to the sclera, we cannot perform traditional multi-view reconstruction to obtain the iris geometry because the refractive cornea distorts the views of the iris. Additionally, the cornea transitions smoothly in opacity from fully transparent to fully opaque at the sclera, and this smooth transition can confuse multi-view correspondence matching. For these reasons, we create a specific iris reconstruction algorithm that is designed to handle these constraints. Since the iris is coupled with the pupil, our method begins by localizing the pupil in 3D. The iris geometry is then reconstructed and filtered, using the pupil as initialization. Finally, we combine iris reconstructions from captures with different pupil dilations, allowing us to parameterize and animate the deformation of the iris during pupillary response.

4.4.1 Pupil Reconstruction

The pupil has a very prominent position at the center of the eye, which makes it visually important and artifacts on its boundary would be clearly visible. Therefore, we require a reconstruction method for the pupil boundary that is robust with respect to perturbations like, for example, those caused by the flash highlight. This robust boundary is used to constrain the iris and also to guide the initial meshing of the iris.

Initialization: The pupil is initialized with the pupil mask boundaries that we detect in image space. Each boundary is triangulated from multiple views, taking into account refraction at the cornea, and we fit a circle to the

triangulated points. The required image correspondences for the triangulation are obtained from the optical flow, which we already computed for the refraction constraints of the cornea optimization.

Refinement: As this initial estimate tends to be rather inaccurate due to inconsistencies between pupil masks, we refine the estimated 3D circle in an optimization that uses two data terms and two regularization terms. The data terms come from two additional cues about the pupil location, 1) an image term that incorporates the result of an image-based pupil detection algorithm, and 2) a mesh term that incorporates an approximate 3D surface reconstruction of the pupil region, triangulated from image correspondences found using optical flow. The two regularization terms control the overall shape and smoothness of the pupil. Based on these terms, we define an energy function for the pupil as

$$E = \lambda_I E_I + \lambda_M E_M + \lambda_C E_C + \lambda_S E_S, \tag{4.5}$$

which we minimize for a set of $n = 50$ pupil samples taken on the initial circle, with weights of $\lambda_I = 10$, $\lambda_M = 1000$, $\lambda_C = 10000$, and $\lambda_S = 1000$ for all data sets. In the following, we will describe each of the energy terms in more detail.

Image Term: We project the initial pupil circle into the cameras and blur the images radially along the produced ellipses. We then use a radial edge detector to locate the edge between the pupil and the iris, and we apply radial non-maximum suppression (NMS) to the response. We define the image data term as

$$E_I = \frac{1}{n} \sum_{i=1}^{n} \left\| P(\mathbf{p}_i) - \mathbf{p}_i^{edge} \right\|^2, \tag{4.6}$$

where $P(\mathbf{p})$ is the projection of sample point $\mathbf{p}$ into the image plane through the cornea, and $\mathbf{p}^{edge}$ is the position of the closest point on the detected edge.

Mesh Term: We create an approximate 3D surface mesh in the vicinity of the pupil by triangulating rays from multiple views refracted at the corneal interface, again with the help of optical flow to provide correspondences. The mesh term for the pupil location then consists of the distances between the pupil samples and the generated mesh, given by

$$E_M = \frac{1}{\sum_{i=1}^{n} c_i} \sum_{i=1}^{n} c_i \left\| \mathbf{p}_i - \mathbf{p}_i^{mesh} \right\|^2, \tag{4.7}$$

where the distances are weighted with the triangulation confidences c of the mesh. The triangulation confidence is defined as a linear function of the triangulation residuals, which maps a residual of 0mm to a confidence of 1 and a residual of 0.05mm to a confidence of 0 and clamps all the values outside this range.

Regularization Terms: We allow the samples to deviate orthogonally from the perfect circle, but we penalize these deviations with

$$E_C = \frac{1}{n} \sum_{i=1}^{n} \left\| \mathbf{p}_i - \mathbf{p}_i^{circle} \right\|^2, \tag{4.8}$$

where $\mathbf{p}^{circle}$ is the corresponding point of $\mathbf{p}$ on the circle. To obtain a smooth pupil we also penalize strong changes in the deviations from one sample to the next, using the following smoothness term, where r is the radial and o the orthogonal component of the offset with respect to the circle.

$$E_S = \frac{1}{n} \sum_{i=1}^{n} \left[(2r_i - r_{i+1} - r_{i-1})^2 + (2o_i - o_{i+1} - o_{i-1})^2 \right], \tag{4.9}$$

Finally, we minimize the sum of all these terms with the Levenberg-Marquardt algorithm to find the position, the radius, and the per-sample deviations from a circle of the pupil. During the optimization, we constrain the normal of the pupil circle to the normal of the plane fit to iris mesh samples taken 1 mm away from the initial pupil boundary estimate to be more robust. Fig. 4.10 illustrates the resulting sample positions both in 3D and projected onto an image (in green), given the initial estimate (in red).

4.4.2 Iris mesh generation

We use the reconstructed pupil boundary to initialize the iris mesh. Starting with a closed uniform B-Spline that we fit to the optimized pupil samples, we scale the spline radially in 0.025mm steps to create a sequence of larger and larger rings up to an iris radius of 7mm. These rings are sampled 600 times and a triangle mesh is created. This will serve as the topology for the iris.

In a second step, we reconstruct the correct position of each iris vertex. Each vertex is projected (through the cornea) into a reference camera, where flow-based correspondences to other views are computed. We triangulate the vertex position by minimizing the squared distances between the vertex and the refracted rays formed by the correspondences. This minimization is equivalent to minimizing the surface error defined in Section 4.3.3. In addition, the rays are weighted by the root mean square difference of the corresponding 7x7 pixel blocks in image space. In order to reduce high frequency noise, the entire mesh reconstruction process is repeated for a second reference camera to obtain a second mesh hypothesis which is combined with the first one through weighted averaging.

4.4.3 Mesh cleanup

The reconstructed iris mesh can be noisy and distorted at the boundaries due to the translucent sclera affecting the optical flow. We perform four operations to filter the iris mesh.

Spike Filtering: Spikes are detected by computing a 3-ring neighborhood around each vertex. If the distance between the vertex and the mean of the neighboring vertices exceeds a threshold (set to 0.05mm), then the vertices inside the ring are smoothed by solving a Laplacian system, keeping the rest of the vertices fixed.

Boundary Deformation: Two criteria are used to label distorted boundary vertices: a threshold on the triangulation residuals (set to 0.05mm) and an angle threshold between the smoothed vertex normal and the normal of the pupil set to 30 degrees. We dilate the labeled region and smooth those vertices in the normal direction.

Mesh Relaxation: The mesh is relaxed locally to improve the triangulation by removing skinny or overlapping triangles.

Pupil Constraint: The vertices at the pupil boundary are constrained to the detected pupil shape. The constraint is enforced with a local Laplacian system, where the pupil vertices as well as all mesh vertices farther than 1mm from the pupil are constrained. The vertices in-between are deformed but the local shape is preserved.

Finally, the two independently triangulated and cleaned mesh hypotheses are averaged to create the iris mesh.

4.4.4 Mesh Propagation

We now combine iris reconstructions from captures with different pupil dilations. Each mesh is reconstructed independently, with different topology and vertex counts. We wish to compute a new set of iris meshes that are in vertex-correspondence, allowing us to compute a per vertex deformation model.

We begin by computing per camera optical flow [Brox et al., 2004] between neighboring poses. Since the vertices are propagated from one pose to the next, drift might accumulate. To minimize the total amount of drift we select a reference pose in the middle of the dilation sequence and compute the optical flow in both dilation directions from there. To find the vertex correspondences we project each vertex from the source mesh into all the target pose cameras taking into account the refraction at the cornea. With the resulting image positions and the optical flows we compute a set of rays that we refract at the cornea and intersect with the iris of the target pose. The target pose vertex is computed as the median of all the intersections. To ensure a clean pupil we enforce the pupil constraint and relax the mesh in the same way as described in Section 4.4.3.

4.4.5 Temporal Smoothing and Interpolation

In order to animate the pupil dilation, we will use the captured pupil poses as keyframes and interpolate linearly in-between. In practice we found that the dilation of the pupil cannot be accurately controlled, and so the pupil diameter tends to decrease in irregular steps. This can lead to multiple poses with very similar diameters and geometry, but with different high frequency reconstruction noise, which leads to artifacts when interpolating. In order to smoothly integrate meshes from similar pupil radii, we compute two linear regression models for all poses within a distance of 1mm pupil radius. The first regression model expresses the vertex position and the second model the Laplacian vector as a function of the pupil radius. We solve for the smoothed mesh by evaluating both models and solving the resulting Laplacian system with equal weights given to the Laplacians and the positions.

4.4.6 Iris Texturing

Iris textures can be computed from a single view, but these textures will contain undesired artifacts like highlights, washed out regions close to the boundary, dust on the cornea, etc. These artifacts can be attenuated by combining the textures from multiple views of the same iris dilation. We compute a contribution map for each view which is set to 1 if the pixel is the most saturated from all the candidates and to 0 otherwise. These maps are then blurred with a small Gaussian kernel of 3 pixels. Based on these contribution maps, the textures from the different views are blended into a single texture. Picking the most saturated pixels will reduce artefacts caused by illumination pollution from the flash light and by superposition of the white sclera at the semi-transparent sclera-cornea transition alike. Then, we combine the textures from several iris dilations using the median to attenuate shading changes caused by the deforming iris.

4.5 Results

In this section we highlight the results of our eye capture technique by illustrating the reconstructions of a variety of human eyes, each with its own intricacies and details.

We begin by analyzing the common assumption that eyes can be modelled as two spheres, a large one for the eyeball and a smaller one for the cornea. In our work we show that this assumption is inaccurate, which we can illustrate by overlaying a cross-section of a captured eye on top of the simple model (Fig. 4.11, left). Furthermore, it is often assumed that an eye is symmetric about the view vector and that the left and right eye can be modelled similarly. By capturing both the left and right eye of an actor, we demonstrate that each eye is in fact unique and shows strong asymmetry individually, but when combined the expected left/right symmetry is clearly visible. We believe these results have the potential to change how eyes are traditionally modelled in computer graphics.

Our eye capture method is robust, which we highlight by reconstructing 9 different eyes from 6 different actors. The full set of reconstructions, shown in Fig. 4.12, contains a variety of different iris colors, surface details, textures, and overall eye shapes. Each eye has unique details, but we observed that the differences between people are more significant than the differences between the two eyes of the same person, an expected phenomenon that helps to validate our reconstruction results. For example, the two brown eyes in

the center (5th and 6th from left) are larger than the rest. These represent the eyes of an actor with severe *myopia* (or short-sightedness), which is often correlated with larger-than-normal eyes [Atchison et al., 2004].

Every human eye is unique and contains minor intricacies that add to the identity of the person. Our capture approach aims to reconstruct all the visible intricacies. In particular, our sclera reconstruction is able to acquire high-resolution surface variation including small details and Pingueculas, as shown in Fig. 4.13.

Even more unique is the iris. Fig. 4.14 illustrates one pose of the reconstructed irises for our 9 actors, visualized on their own with blue shading for comparing the geometry. The individuality of iris shape from eye to eye is clearly visible, again highlighting the importance of capturing real eyes using the proposed technique. Fig. 4.15 shows a close-up view of a captured iris with both surface details and texture, rendered with refraction through the cornea.

One of the most interesting features of human eyes is the time-varying deformation of the iris during pupillary response. Our method is able to recover this deformation, which we illustrate for one actor in Fig. 4.16. As the pupil changes size, our reconstruction shows that the iris dilator muscle creates significant out-of-plane deformation, which largely contributes to the realistic appearance of the eye. To further illustrate how unique this effect is for each iris, we provide side-view renders for two additional irises and three pupil radii in Fig. 4.17.

The ability to reconstruct a per-vertex deformation model for the iris during pupil dilation allows us to animate the captured eyes. We show two different applications for iris animation in Fig. 4.18. The first is a motion capture scenario. Analogous to the way facial animation rigs are often built from high-quality scan data and then later animated from low-resolution mo-cap markers, our captured irises can be animated from a single low-quality video stream. As a demonstration, we detect the pupil size of an actor in each frame of such a video and compute the corresponding iris shape for a captured actor (Fig. 4.18, top). A second application is to automatically make a digital character respond to lighting changes in the 3D environment. Using predicted pupillary response curves introduced in the seminal work of Pamplona et al. [2009], we can animate the captured iris geometry to show a character dynamically responding to a light source turning on and off (Fig. 4.18, bottom). As these applications target iris animation, the results are best viewed in the accompanying supplemental video.

We compare our results qualitatively with the seminal work of François et

al. [François et al., 2009] in Fig. 4.19. While the strength of their approach is its simplicity, our method arguably excels in quality. Since we aim to accurately reconstruct all the intricacies of the eye, we more faithfully capture the uniqueness and realism of eyes. In particular, our reconstructions show the asymmetric shape of the sclera and fine scale surface variation. Our iris geometry is reconstructed rather than heuristically synthesized, and we even recover small defects like the aforementioned pingueculas and the non-circular transition between sclera and iris in Fig. 4.19.

In order to provide context for visualizing the captured eyes we combine them with the partially reconstructed face scans of the actors. We use a simple combination process that automatically fits the face geometry around the back of the eyeball using a Laplacian deformation scheme. While the approach is rudimentary, the result is sufficient to simulate an eye socket for holding the reconstructed eye. Several results for different actors are shown in Fig. 4.20, rendered from multiple viewpoints. We note that more sophisticated methods for capturing the face region around the eyeball would be ideal topics for future research.

Finally, we wish to highlight the potential impact that capturing real eyes can have in creating artistic digital doubles - a task that is often performed for visual effects in films. To this end, we combine both of the captured eyes of an actor together with a face scan to create a compelling rendition of an artistically designed digital human character, as shown in Fig. 4.21. Such a result would traditionally take significant artistic skill and man-hours to generate, in particular if the digital character should closely resemble a real actor. Our result was created with very little effort, thanks to our new method for capturing real human eyes.

All our textured results are rendered in a single pass using Autodesk Maya with Octane Render. We use built-in diffuse materials with subsurface scattering for the sclera and the iris, and reflective/refractive materials for the cornea plus a water layer created by extruding the sclera by 0.1 mm. The total processing time to reconstruct a complete eye on a standard Windows PC with a 3.2 Ghz 6-core CPU is approximately 4 hours (2 hours for initial reconstruction, 20 minutes for the sclera, 5-10 minutes for the cornea, 1 hour for the iris, and 40 minutes for unoptimized texture synthesis). The main bottleneck is the computation of optical flow.

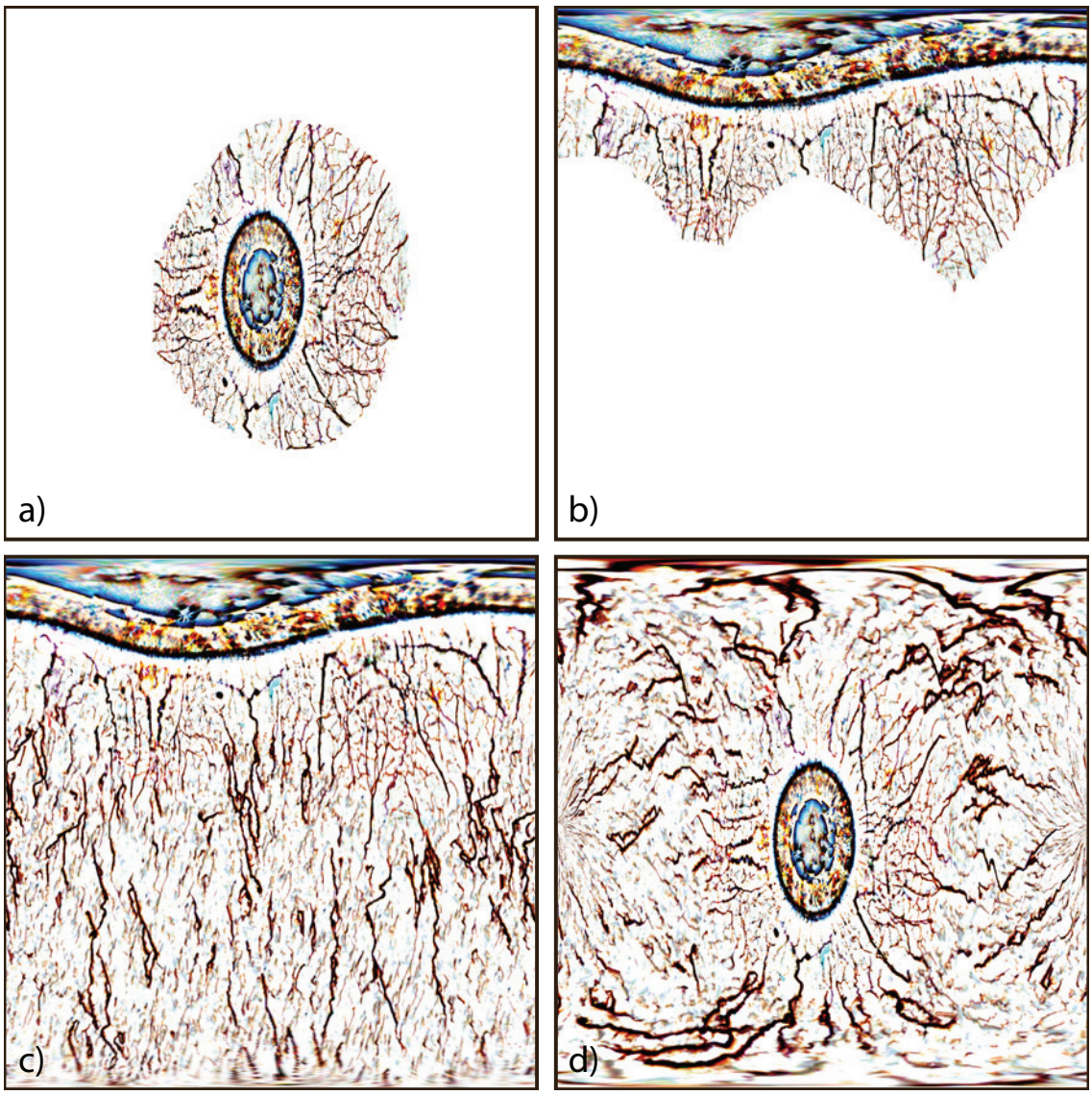

Figure 4.7: *Texture synthesis is performed on the high frequency information in order to complete the texture. A captured texture (a) is rotated to polar coordinates (b) and synthesis is performed in a way that preserves vein orientation (c). The final texture is rotated back to Cartesian coordinates (d).*

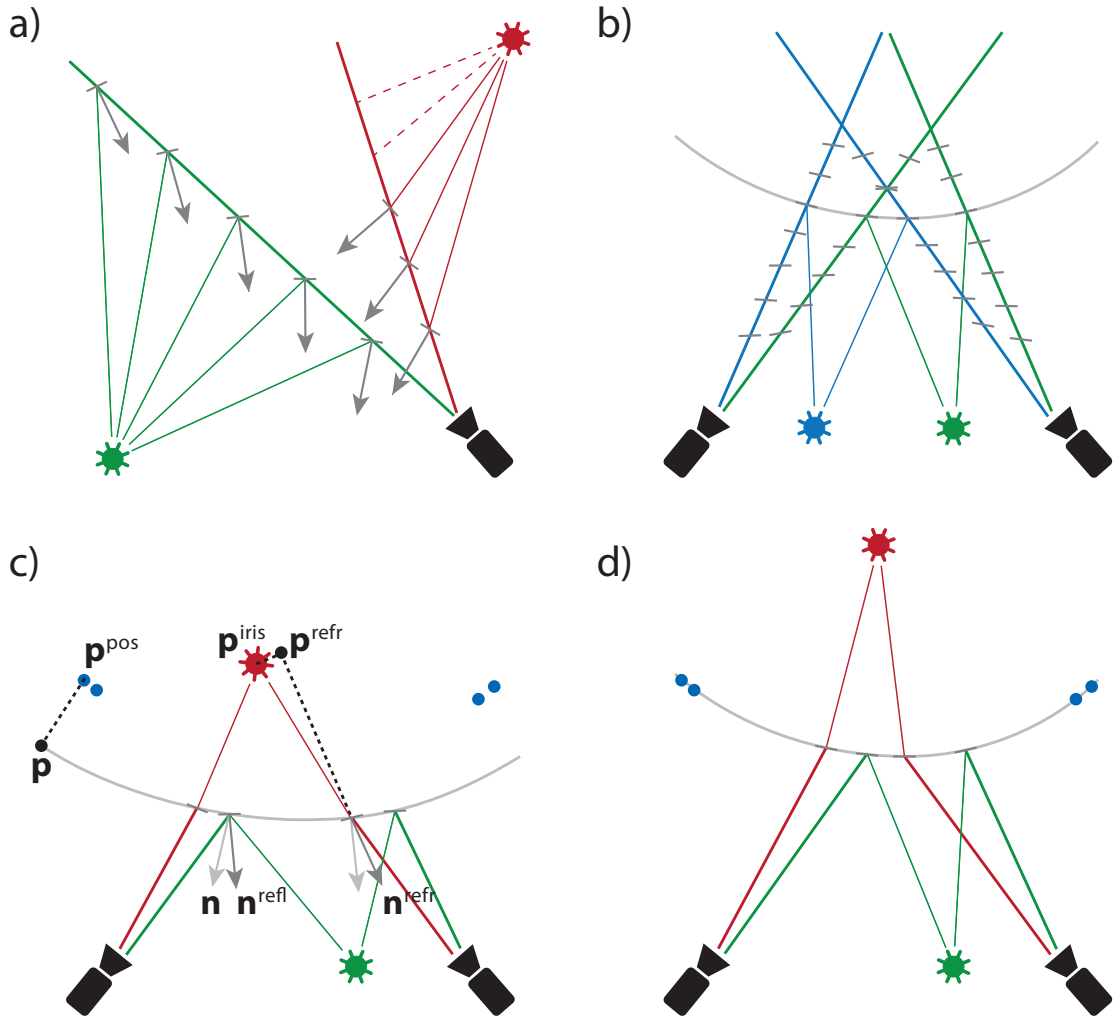

Figure 4.8: *The depth/normal ambiguity of a highlight (a) and the sparse normal field in a multi-view setting (b). Corneal constraints before (c) and after optimization (d).*

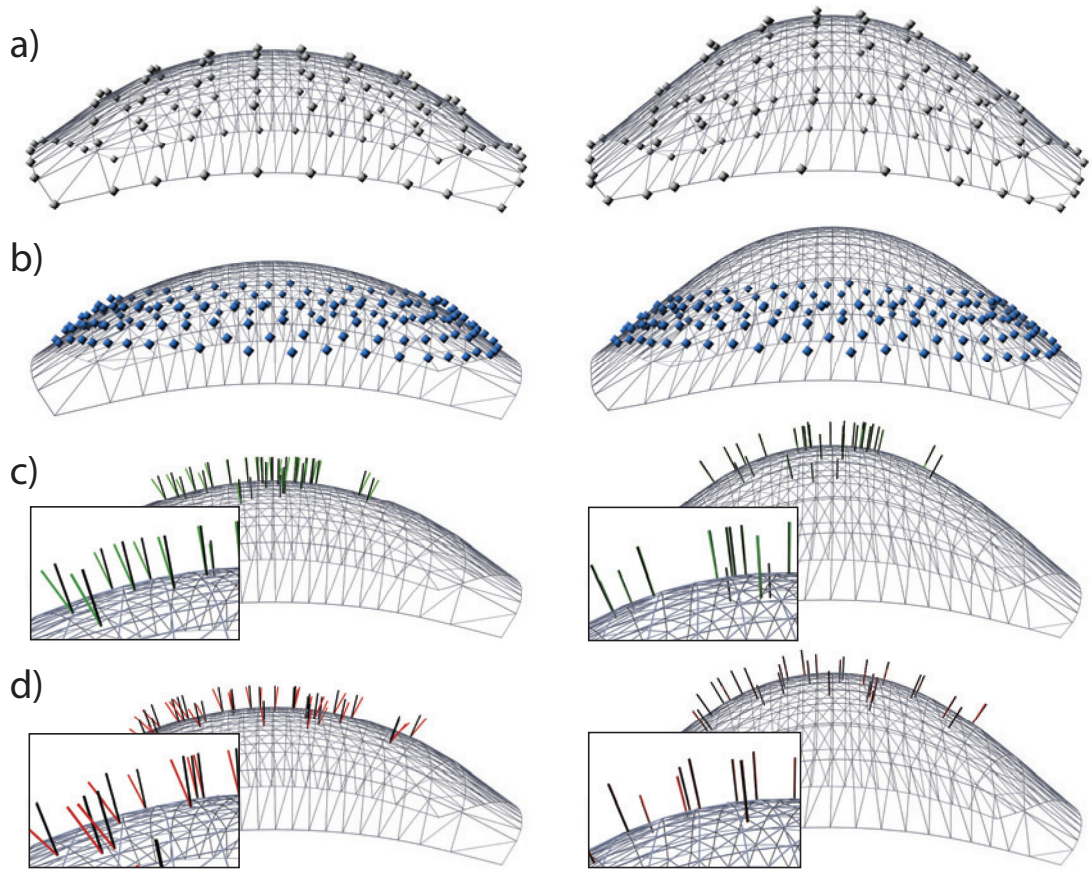

Figure 4.9: *Visualization of the B-spline control points (a), the position constraints (b), and a subset of the reflection (c) and refraction (d) constraints on the initial (left) and optimized (right) surfaces.*

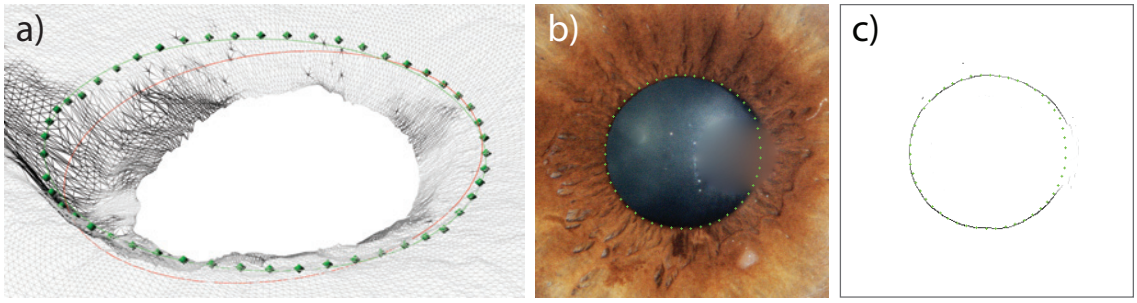

Figure 4.10: *Pupil reconstruction. Given an initial pupil boundary estimate (red) from the triangulated image-based pupil masks, we solve for the optimal pupil boundary (green). The resulting pupil samples are shown in 3D (a), projected onto one image (b), and overlaid onto the response of the pupil edge detector (c).*

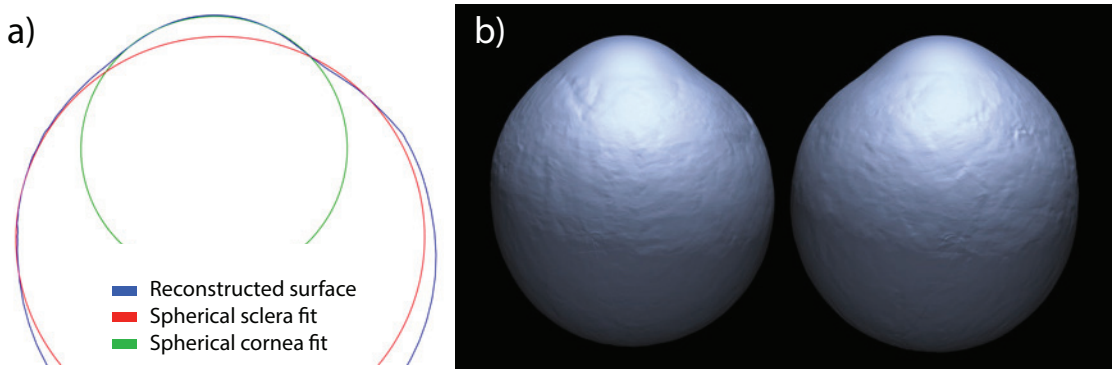

Figure 4.11: *The traditional assumption that an eye can be modelled as two spheres (red and green) is inaccurate, as indicated by a top-view cross-section of our reconstruction in blue (left). Eyes also exhibit strong asymmetry, which we show by reconstructing both the left and right eyes of the same actor (right).*

Figure 4.12: *We highlight the robustness of our technique by capturing a wide variety of eyes. This dataset consists of different iris colors, individual sclera textures, and unique geometry for each eye. In particular, we can see that the two brown eyes in the center are larger than the others - further highlighting the importance of capture over generic modelling. The measured index of refraction is listed under each eye.*

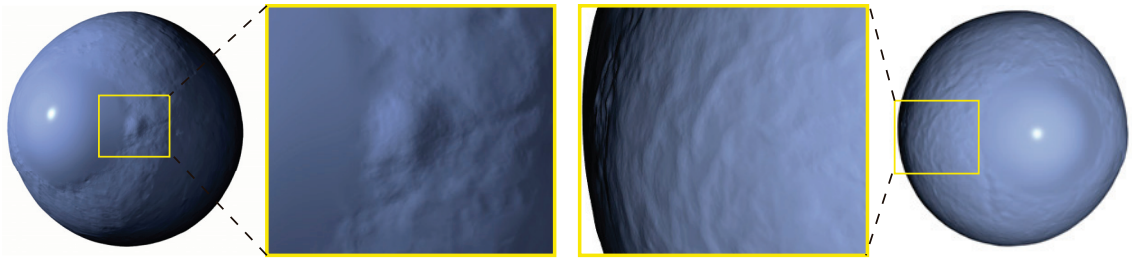

Figure 4.13: *Our sclera reconstruction technique is able to acquire fine scale details including Pingueculas (left) and surface variation (right) that is unique to each eye.*

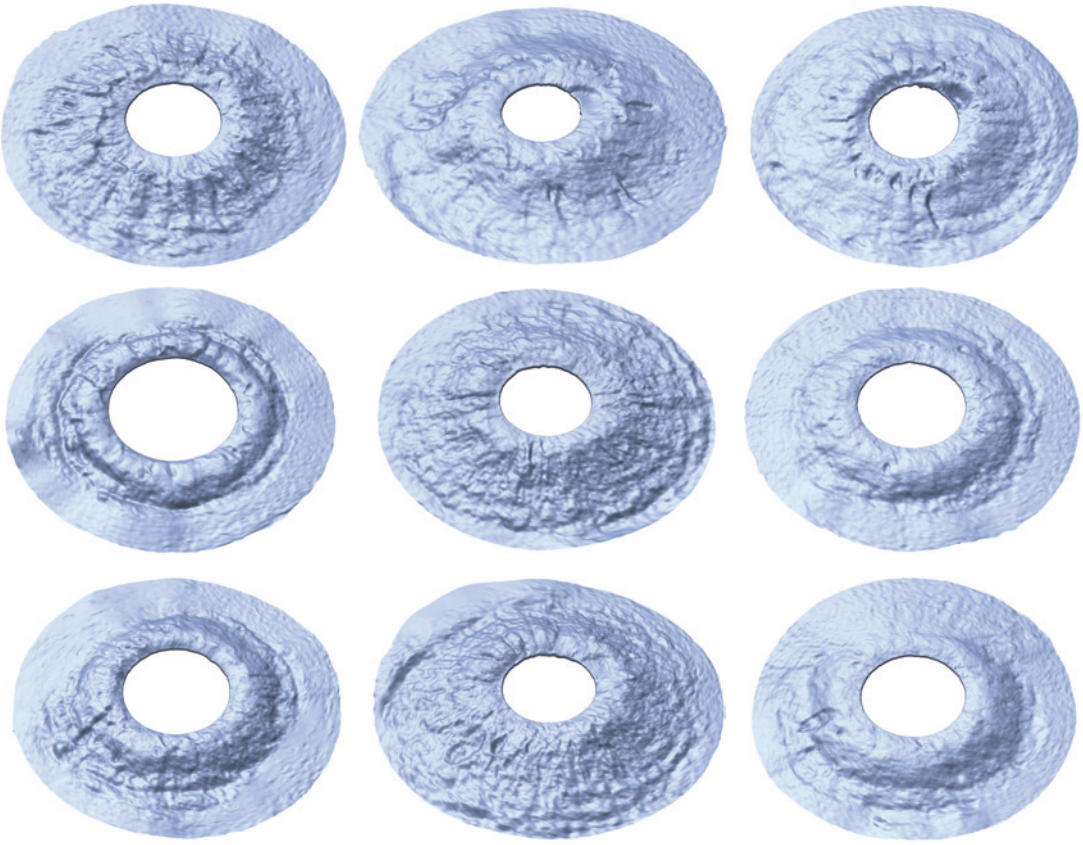

Figure 4.14: *We highlight the uniqueness of each individual iris by visualizing the 9 captured irises with blue shading.*

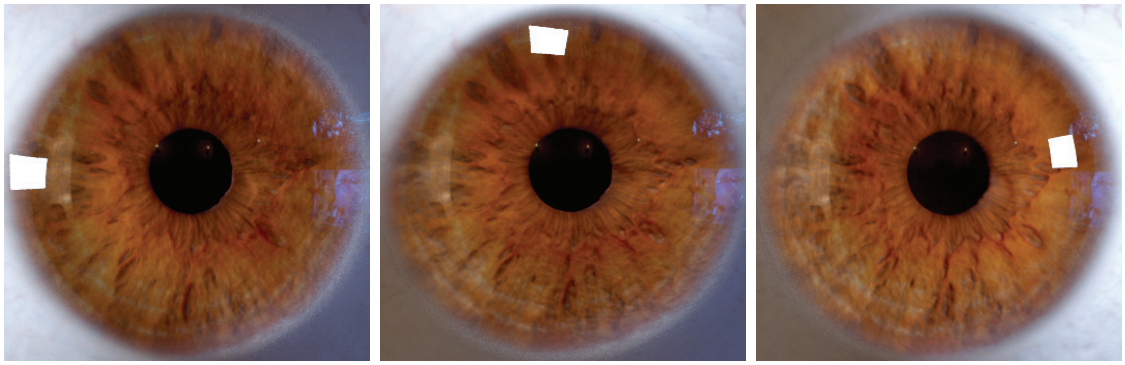

Figure 4.15: *A close-up of detailed iris geometry and texture captured with our method, rendered in high-quality with refraction through the cornea and three different light positions.*

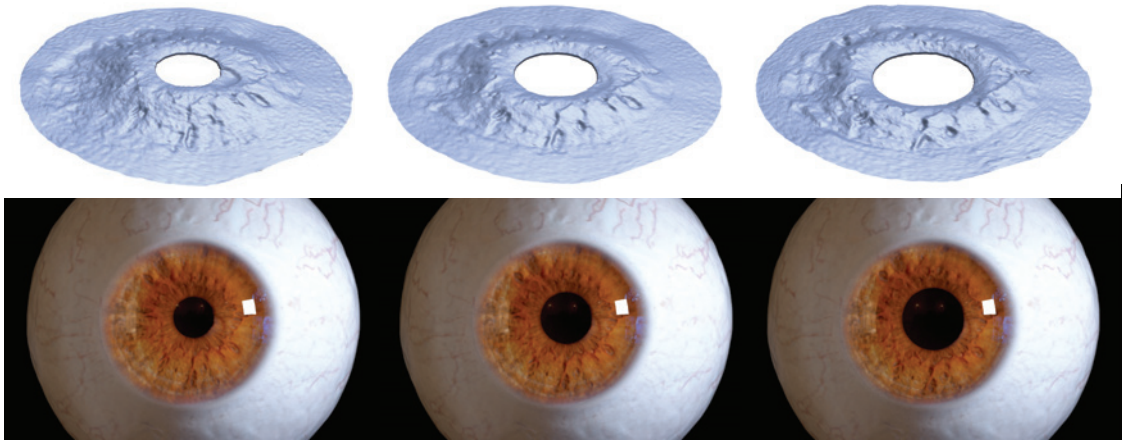

Figure 4.16: *We measure the iris under various amounts of pupil dilation. As can be seen, the iris dilator muscle creates significant out-of-plane deformation as the pupil becomes larger (left to right).*

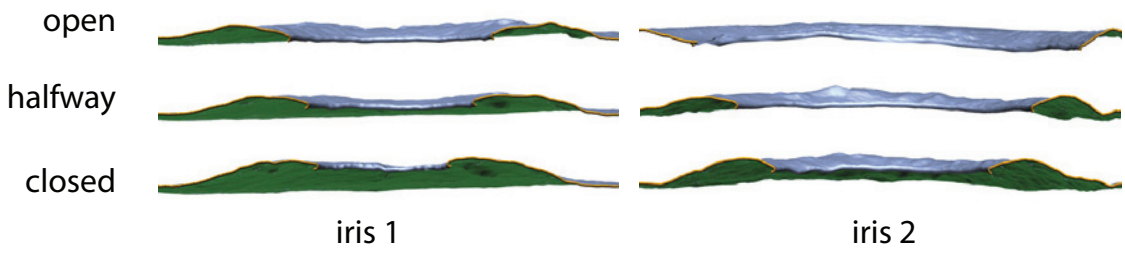

Figure 4.17: *We highlight the uniqueness of each eye's iris deformation during pupil dilation by showing the deformations from a side view for two different eyes and three different pupil sizes.*

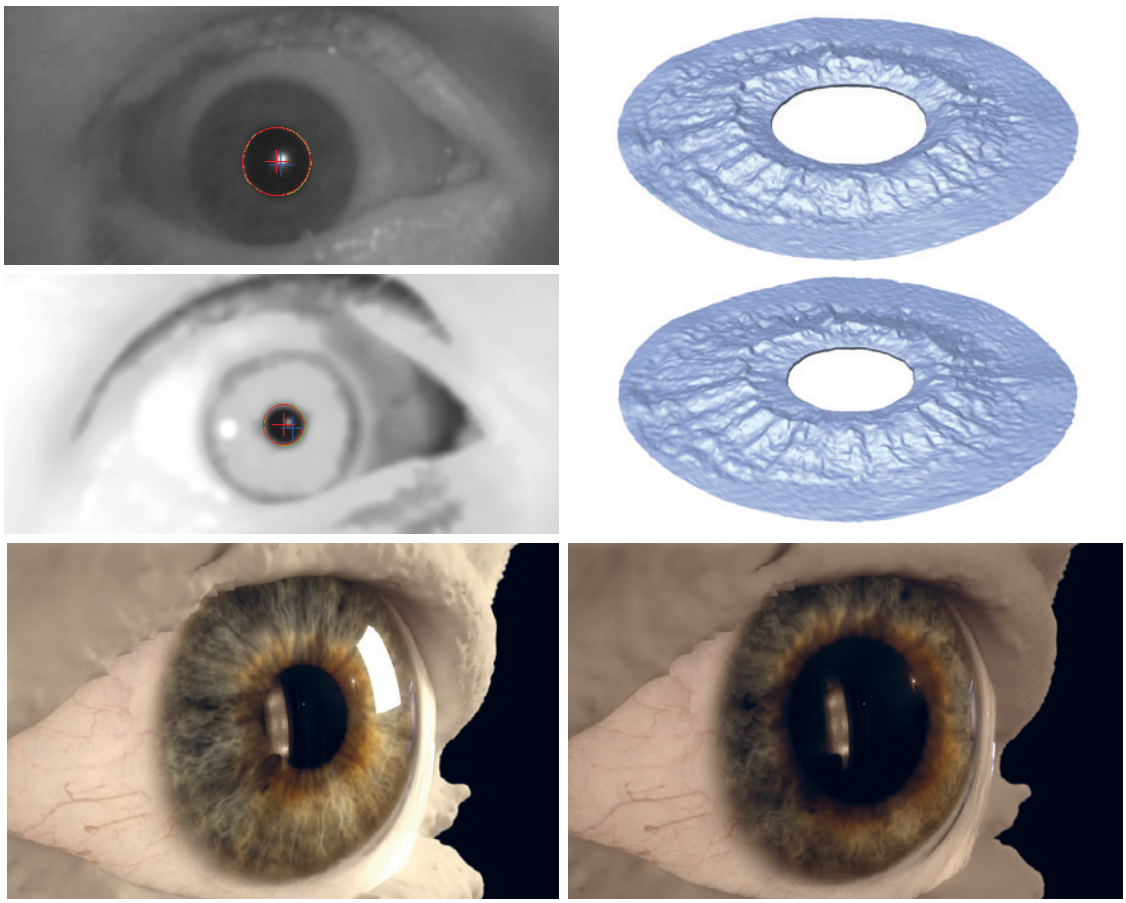

Figure 4.18: *We can apply the measured iris deformation in a pupil dilation animation. Here we show two applications: one where an actor's pupil is tracked in a single low-quality infra-red video and the corresponding radius is applied to our model (top). A second application is to automatically make a digital double respond to lighting changes in the 3D environment (bottom).*

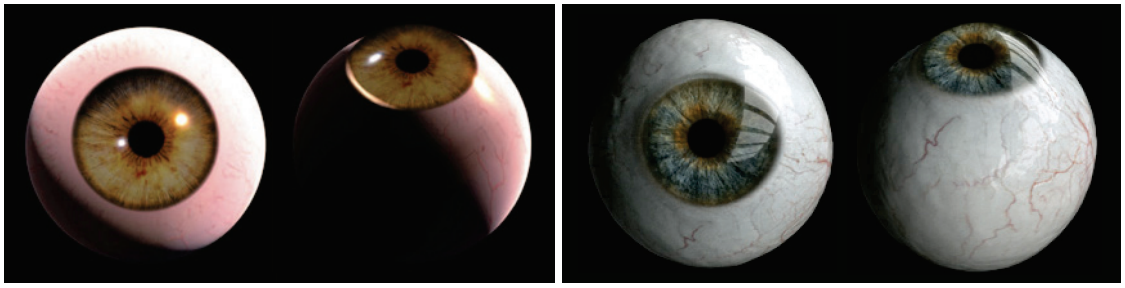

Figure 4.19: *We show a comparison with François et al. [2009] on the left. They employ a generic eyeball model combined with a heuristic to synthesize the iris morphology. Note how our results shown on the right faithfully capture the intricacies of this particular eye, such as its asymmetric shape, the small surface variation, and the non-circular iris-sclera transition.*

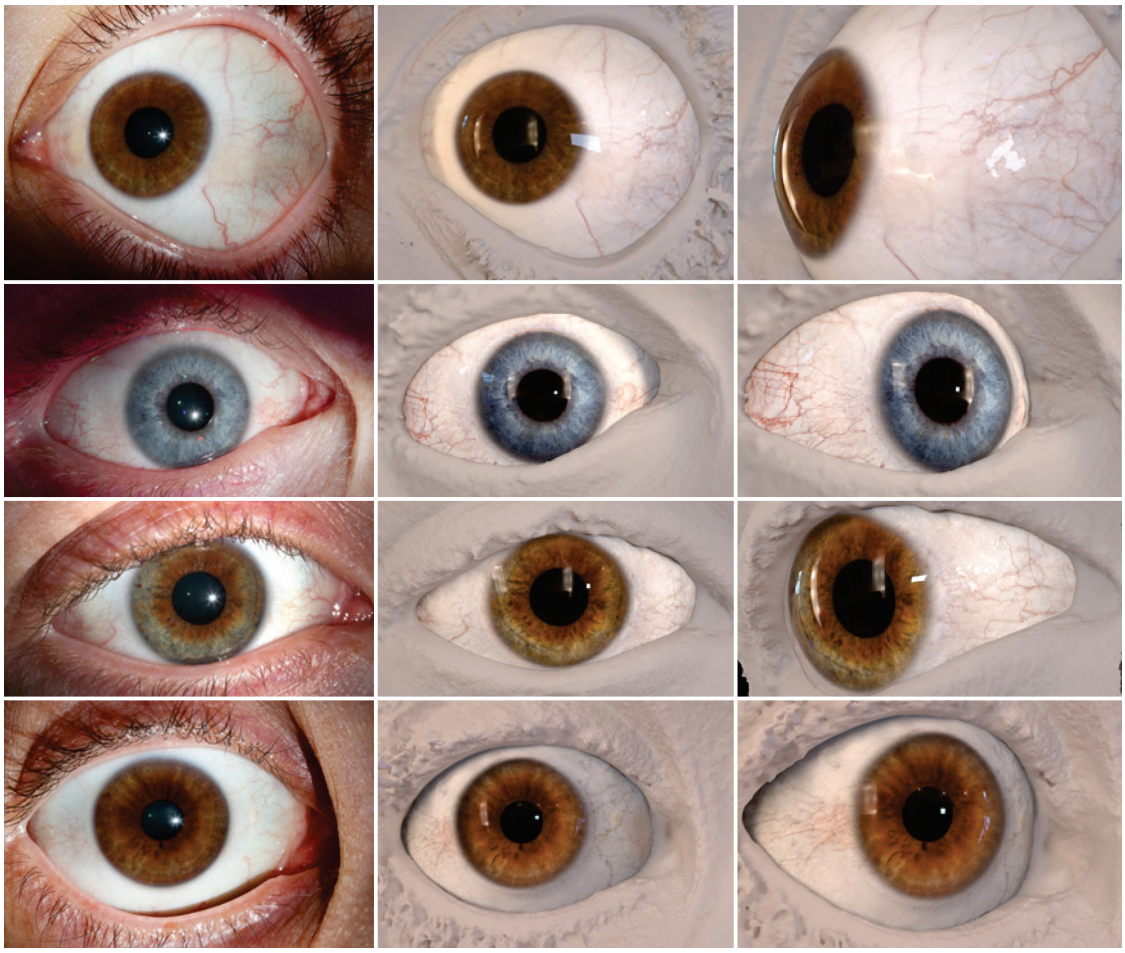

Figure 4.20: *We further demonstrate our results by combining the captured eyes with partial face scans, and rendering from various viewpoints with different environment lighting. This figure shows how the reconstruction results could be used in the visual effects industry for creating digital doubles.*

Figure 4.21: *We combine both captured eyes of an actor together with a face scan to further demonstrate how our results can be used to create artistic digital doubles.*

CHAPTER 5

Parametric Eye Model

Figure 5.1: *We present a new parametric eye model and image-based fitting method that allows for lightweight eye capture at very high quality. Our eye capture method can be integrated with traditional multi-view face scanners (as seen here), or can operate even on a single image.*

In Chapter 4 we present a system to capture the shape and appearance of the human eye at a very high level of detail. This system, however, is tedious to use for the operator and the subject and requires a setup with several cameras and lights. In this chapter we introduce a method to simplify the acquisition process, while generating eyes with a similar level of detail.

Our main goal is to generate a parametric eye model that can be fit to sparse image data, leveraging a database of high-resolution eye reconstructions. Since eyes are composed of several different components and contain interesting variations at multiple scales, a single all-encompassing parametric model is not practical. For this reason we compose a model built from three separate components, namely an *eyeball model* (Section 5.2) that represents the low-frequency variability of the entire eyeball shape, an *iris model* (Section 5.3) that represents the high-resolution shape, color and pupillary

deformation of the iris, and a *sclera vein model* (Section 5.4) that represents the detailed vein structure in the sclera, including the vein network and the width and depth of individual veins, as well as fine-scale geometric surface details. In Section 5.5 we show how the model parameters can be estimated by fitting the model to 3D face scans, single images, or even artistic portraits, drastically simplifying the process of creating 3D high-quality eyes.

5.1 Input Data

With our eye reconstruction method we captured an eye database containing 30 high-quality eyes. This database provides high-resolution meshes and textures for the white sclera and the colored iris (please refer to the schematic in Fig. 1.3). The iris geometry is provided as a deformation model making it possible to create meshes for an entire range of pupil dilations. The database contains eyes of different iris colors ranging from brown to green-brown to blue, and the high resolution geometry captures intricate eye-specific surface details. A subset of the database eyes are shown in Fig. 5.2. We assume that right and left eyes are anti-symmetric and we thus mirror the left eyes when building the model for the right eye. Similarly, a mirrored version of the model can be used to represent the left eye. The data provided contains also a limbus opacity mask defining the transparency transition from sclera to cornea, from which the position of the limbus can be extracted by mapping the 50 percent opacity level to the mesh.

5.2 Eyeball Model

The eyeball is represented by a morphable model [Blanz and Vetter, 1999], which has been demonstrated to be a good representation to capture low-frequency variation. A morphable model is a linear combination of a set of samples. To avoid overfitting to the samples, the dimensionality is oftentimes reduced using methods such as principal component analysis (PCA). PCA computes the mean shape plus a set of mutually orthogonal basis vectors from the samples, ordered according to their variance. Truncating the dimensions with lower variance leads to a subspace that captures the major variation in the samples and is resilient to noise. In addition to the shape variation, our model also includes a rigid transformation for the eyeball as well as a uniform scale factor.

A morphable model requires all samples to be in perfect correspondence, which is unfortunately not the case for our database of eyes. In our case,

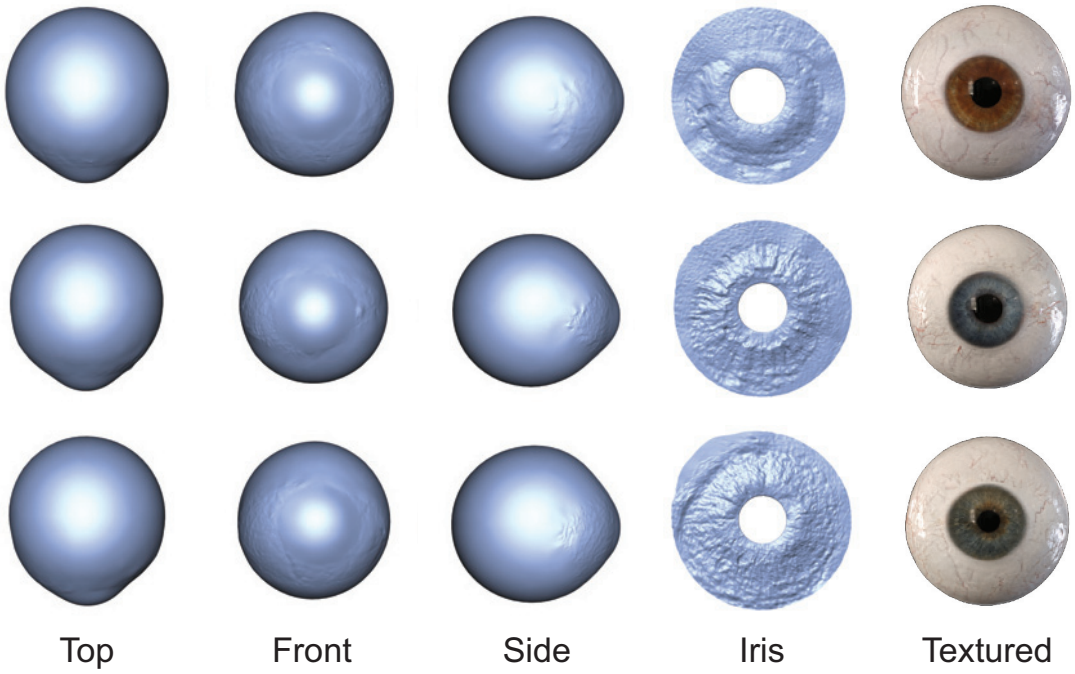

Figure 5.2: *We create a database of eyes, which contains high-resolution meshes and textures for eyeball and iris. Notice how the geometric structure of the iris (4th column) is linked to its color (5th column), in that browner irises are smoother while bluer ones are more fibrous.*

eyeballs exhibit only few semantic features that can be used to establish correspondence. The most important one is the limbus, the boundary between the white sclera and the transparent cornea (Fig. 1.3). Other features are less salient, such as the overall asymmetry of the eye, but have to be encoded as well. These features, however, are not well defined and thus the traditional two step approach to build a morphable model by first establishing correspondences between all samples and then computing the model does not lead to satisfactory results.

Instead, we perform an iterative scheme that alternates between establishing correspondences and computing the model. The algorithm iteratively refines the model in three steps, first by fitting the previous guess of the model to the sample shapes, second by deforming this fit outside of the model in order to more closely fit the samples, and third by recomputing the model from these fits. Next we will discuss these three steps in more detail.

Step 1: Within-Model Fit. The eyeball model $\mathcal{M}$ is fit to a sample shape $\mathcal{S}$ by finding the model parameters $\mathbf{p}$ that minimize the energy

$$E_{model} = \lambda_{shape}E_{shape} + \lambda_{limbus}E_{limbus} + \lambda_{coeff}E_{coeff} \tag{5.1}$$

where the shape term

$$E_{shape} = \frac{1}{|\mathcal{M}|} \sum_{\mathbf{x}_i \in \mathcal{M}|_{\mathbf{p}}} \|\mathbf{x}_i - \chi(\mathbf{x}_i, \mathcal{S})\|^2 \tag{5.2}$$

penalizes the distance between points $\mathbf{x}_i$ on the model $\mathcal{M}$ evaluated at $\mathbf{p}$ and their closest points $\chi(\mathbf{x}_i, \mathcal{S})$ on the sample shape $\mathcal{S}$, and the limbus term

$$E_{limbus} = \frac{1}{|\mathcal{L}^{\mathcal{M}}|} \sum_{\mathbf{y}_i \in \mathcal{L}^{\mathcal{M}}|_{\mathbf{p}}} \left\| \mathbf{y}_i - \phi(\mathbf{y}_i, \mathcal{L}^{\mathcal{S}}) \right\|^2 \tag{5.3}$$

penalizes the distance between points $\mathbf{y}_i$ on the model limbus $\mathcal{L}^{\mathcal{M}}$ evaluated at $\mathbf{p}$ and their closest points $\phi(\mathbf{y}_i, \mathcal{L}^{\mathcal{S}})$ on the limbus of the sample shape $\mathcal{L}^{\mathcal{S}}$. The shape coefficients term

$$E_{coeff} = \frac{1}{k} \sum_{i=1}^{k} \left(\frac{c_i - \mu_i}{\sigma_i} \right)^2 \tag{5.4}$$

penalizes shape coefficients c_i far away from the mean coefficients of the current model $\mathcal{M}|_{\mathbf{p}}$, where μ_i and σ_i are the mean and the standard deviation of the i-th shape coefficient. The number of shape coefficients is k. We set the constants to $\lambda_{shape} = 1$, $\lambda_{limbus} = 1$, and $\lambda_{coeff} = 0.1$.

The parameter vector $\mathbf{p}$ consists of a rigid transformation, uniform scale, as well as an increasing number of shape coefficients as discussed in step 3.

Step 2: Out-Of-Model Fit. The morphable model $\mathcal{M}|_{\mathbf{p}}$ fit in the previous step will not match the sample $\mathcal{S}$ perfectly since it is constrained to lie within the model space, which has only limited degrees of freedom. In order to establish better correspondences for the next step, we therefore need to further deform the mesh non-rigidly to bring it out-of-model. We use a variant of the non-rigid deformation method proposed by Li et al. [2008], which was designed for aligning a mesh to depth scans using a deformation graph and a continuous approximation of the target shape. In our context, we wish to align the fitted model mesh to the database samples. We modify the method of Li et al. to operate in the spherical domain rather than the 2D depth map

domain, and we add additional constraints to match both the limbus boundary and the mesh normals during deformation. We use two spherical coordinate parameterizations which are wrapped like the patches of a tennis ball so that the distortion in the domains is minimal. Closely following Li et al., the energy that is minimized by the algorithm can be expressed as the sum of the following terms:

$$E_{nonrigid} = \lambda_r E_r + \lambda_s E_s + \lambda_f E_f + \lambda_n E_n + \lambda_l E_l, \tag{5.5}$$

where E_r and E_s correspond to the *rigid* and *smooth* energies as defined in the original paper of Li et al. [2008]. We set the constants to $\lambda_{r,s} = 0.01$ and $\lambda_{f,n,l} = 1$. The shape and limbus energies E_s and E_l correspond to the ones used in the previous step as defined in Equation 5.2 and Equation 5.3, respectively. The normal energy E_n is defined analogously to the shape energy as the Euclidean difference between the normals of the model and the normals of the respective closest points on the sample shapes. The non-rigid deformation produces meshes $\{\tilde{\mathcal{M}}\}$ which closely resemble the database samples $\{\mathcal{S}\}$ but have the same topology as the eyeball model.

Step 3: Update Eyeball Model. From the non-rigidly aligned shapes $\{\tilde{\mathcal{M}}\}$ an updated version of the model is computed using PCA and keeping only the mean shape plus the k most significant dimensions. In order to be robust towards initial misalignment, the algorithm starts with a very constrained model that consists of the mean shape only ($k = 0$).

The proposed algorithm iterates these three steps and increases the dimensionality k of the model every 10 iterations by including the next most significant PCA vector. Increasing the dimensionality allows the model to better explain the data and by doing so gradually provides robustness. We use a fixed amount of iterations because the error is not comparable from one iteration to the other since the model has been updated at the end of each iteration. After expanding the model three times ($k = 3$), we found that the first mode of the deformable model accounts for 92 percent of the variance, the first two for 96 percent, and the first three for 98 percent of the variation, which covers the low-frequency variation we are targeting with this model. The final eyeball model thus contains 10 dimensions, six of which account for rigid transformation, one for uniform scale, and three for shape variation. Fig. 5.3 shows the deformation modes of our eyeball model.

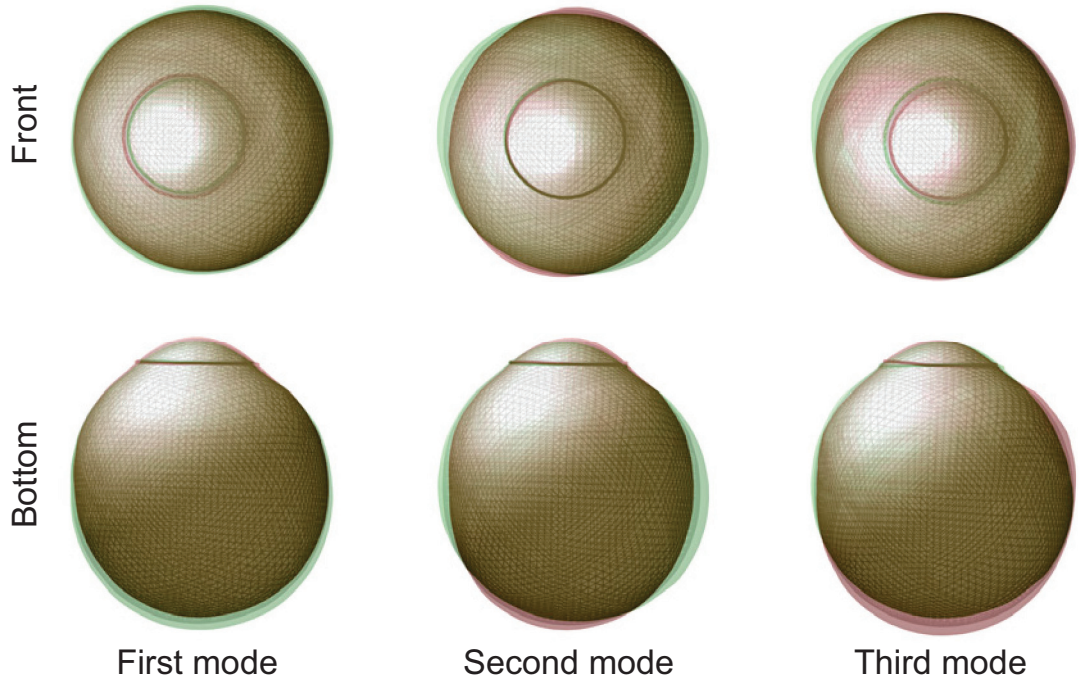

Figure 5.3: *This figure visualizes the three modes of our eyeball prior. For visualization purposes we show the normalized dimensions, scaled by a factor of 50. As the eyeball does not contain much variation, we found three modes to be sufficient as shape prior.*

5.3 Iris Model

We now turn our attention to the iris and describe our model for parameterizing the texture and geometry of an iris given the database of captured eyes. The iris is arguably the most salient component of the eye, and much of the individuality of an eye can be found in the iris. A large variety of irises exist in the human population, and the dominant hues are brown, green and blue. In addition to the hue, irises also vary greatly in the number and distribution of smaller features like spots, craters, banding, and other fibrous structures. Interestingly, iris color and geometry are related, as the iris color is a direct function of the amount of melanin present in the iris. Irises with little melanin have a blueish hue, where irises that contain more melanin become more brown. This accumulation of melanin changes the geometric structure of the iris, with blueish irises being more fibrous and brown irises being smoother overall as shown in Fig. 5.2. We exploit the relationship between color and structure in our iris model and propose a single parameterization that will account for both.

Since irises have such a wide range of variation, it would be impractical to parameterize them using a Guassian-distributed PCA space as we do for the eyeball. Instead, we account for the variability by parameterizing the iris using a low-resolution control map, which represents the spatially varying hue and the approximate distribution of finer-scale features (see Fig. 5.4.b for an example control map). The control map will guide the creation of a detailed high-resolution iris through constrained texture synthesis, using the irises in the database as exemplars. The use of a control map is a very intuitive and convenient way to describe an iris, as it can be extracted from a photograph when reconstructing the eye of a specific person, or it can be sketched by an artist when creating fictional eyes. Based on the low-resolution control map, we propose a constrained synthesis algorithm to generate a detailed color texture in high resolution (Section 5.3.1), and then extend the synthesis to additionally create the geometric iris structure (Section 5.3.2).

5.3.1 Iris Texture Synthesis

Guided by the low-resolution control map our goal is to synthesize a high-resolution texture for the iris based on our eye database, similar to example-based super resolution [Tai et al., 2010] and constrained texture synthesis [Ramanarayanan and Bala, 2007]. We achieve this by composing the high-resolution texture from exemplar patches from the database, following the well-studied image quilting approach introduced by Efros and Freeman [2001]. In our application the process is guided by the control map and selects suitable patches from the database ensuring they conform both with the control map and the already synthesized parts of the texture. Once the patches have been selected, they are stitched together using graph cuts and combined to a seamless texture using Poisson blending. Finally, this texture is merged with the intial control map in order to augment the low-res control map with high-res detail. Fig. 5.4 shows the individual steps, which we will describe in more detail in the following.

Patch Layout. The structure of an iris is arranged radially around the pupil. Operating in polar coordinates (angle/radius) unwraps the radial structure (Fig. 5.4) and presents itself well for synthesis with rectangular patches. We synthesize textures of resolution 1024x256 pixels with patch sizes of 64x64 pixels that overlap each other by 31 pixels in both dimensions. While iris features are distributed without angular dependency, they do exhibit a radial dependency since the structure close to the pupil (pupillary zone) differs substantially from the one closer to the limbus (ciliary zone).

To synthesize a specific patch in the output, the algorithm can thus consider sample patches at any angle with similar radii ($\pm 10\%$). The only drawback of the polar coordinate representation is that the synthesized texture must wrap smoothly in the angular dimension (i.e. across the left and right image boundaries), which is handled by temporarily extending the texture by one block in the angular direction. To guarantee a consistent wrapping the first and last block are updated as pairs.

Output Patch Selection. The iris is synthesized by iteratively placing patches from the database iris textures. In each iteration, we first need to decide where to synthesize the next patch in the output texture. Many synthesis algorithms employ a sequential order, typically left to right and top to bottom. We found that this leads to unsatisfactory results since important features, such as spots or freckles, can easily be missed because neighboring patches in the output may provide a stronger constraint than the control map. Instead we select the next patch based on control map saliency, which synthesizes patches in visually important areas first, thus allowing them to be more faithful to the control map and spreading the control map residual error into less salient areas. Saliency is computed via steerable filters as proposed by [Jacob and Unser, 2004].

Exemplar Selection. Once the location for the next patch to be synthesized has been determined, a suitable patch exemplar has to be retrieved from the database. This exemplar should be faithful to both the control map and any neighboring patches that have already been chosen. Similarity to the control map, denoted e_c, is computed as the mean squared error between a downscaled version of the exemplar and the patch of the control map. To gain invariance over differences in exposure and because the most important quantity at this stage is faithful color reproduction, the error is computed over the RGB channels, but the mean intensity of the exemplar is scaled globally to match the mean intensity of the control patch. Similarity to the already synthesized texture, denoted e_n, is computed as mean squared error over the overlapping pixels. The two similarity measures are linearly combined into a single quantity

$$e = \alpha e_n + (1 - \alpha) e_c, \tag{5.6}$$

where we use $\alpha = 0.25$ for all examples in this chapter. The exemplar patch with the smallest error is chosen.

Patch Merging. The end result of the above steps is a set of overlapping patches that cover the entire texture. Even though the patches have been carefully selected they will still exhibit seams. To alleviate this we follow Kwatra et al [2003] and employ a graph cut to find seams between patches that better respect the underlying image structure, i.e. finding a seam that minimizes the color difference across the cut. For each patch, pixels at the boundary of the patch that overlap neighboring patches are labeled as sinks and the pixel at the center of the patch as a source. A graph for the current block is constructed with horizontal and vertical edges. The capacity of each edge is set to be the difference of the two connected pixels. We use the max-flow/min-cut algorithm of Boykov et al. [2004] to solve for the cut.

Patch Blending. Once merged, the texture has a unique color per pixel with minimal, yet still visible seams between patches. To completely remove the seams we employ Poisson blending [Pérez et al., 2003], setting the desired color gradients across patch seams to be zero while preserving color detail within patches.

Texture Blending. By definition, the synthesized texture T should well-match the control map C and contain more high frequency detail. However, the control map itself already contains a lot of structural information that we wish to preserve. Therefore we propose to blend the synthesized texture with the control map, however we need to take care not to superimpose the same frequencies. Assuming the captured image is in focus, the frequency content of the control map is determined by the resolution at which it was acquired. If we base our synthesis on a low resolution image that captures the complete face, then we want to add a larger range of spatial frequencies than if the original input image was focused onto the eye and hence of high resolution. To avoid superimposing the same frequency bands we thus need to bandpass filter the synthesized texture before blending with the control map. We model the bandpass filter as a Gaussian $\mathcal{G}$ with the standard deviation computed from the ratio in width of the synthesized texture T_{width} and the control map C_{width} as

$$\sigma = \frac{T_{width}}{C_{width}}\sigma', \tag{5.7}$$

where σ' is the standard deviation of the Gaussian at the resolution of the control map, typically set to 1px. In some cases it makes sense to pick a

larger σ' to account for noise or defocus of the control map. The high-pass filtered texture and low-pass filtered control map are then combined as

$$T \leftarrow (T - \mathcal{G} * T) + \mathcal{G} * C, \tag{5.8}$$

and then re-converted from polar coordinates to create the final texture of the iris.

5.3.2 Iris Geometry Synthesis

As mentioned above, there is an inherent coupling between iris texture and geometric structure. The idea is thus to exploit this coupling and synthesize geometric details alongside the iris texture. The database is created with our eye reconstruction method and contains both high-res iris textures and iris deformation models, which encode iris geometry as a function of the pupil dilation. Since the iris structure changes substantially under deformation, we aim to synthesize the geometry at the observed pupil dilation. In addition, the algorithm will also provide extrapolations to other pupil dilations, allowing us to control the iris virtually after reconstructing the eye.

Geometry Representation. The iris geometries in the database are encoded in cylindrical coordinates (angle/radius/height), which renders them compatible to the domain used for texture synthesis. Spatially, the iris deformation model is discretized such that it has one vertex per pixel of the corresponding texture, with full connectivity to its 8 neighbors. Temporally, the deformation model is discretized at four different pupil dilations, spaced equally to span the maximum dilation range common to all exemplars. One of the pupil dilations is picked to match the dilation of the input image.

Synthesizing geometry cannot be performed using absolute spatial coordinates since patches are physically copied from one spatial location in the exemplar to another in the output texture. For this reason, we find it convenient to encode the geometry using differential coordinates that encode the difference in angle, radius and height between neighboring vertices, and then the synthesized geometry can be reconstructed. Specifically, for every vertex, the iris geometry is encoded by the eight differential vectors to its spatial neighbors (in practice, to avoid redundant storage we only need the four vectors corresponding to top, top-right, right, and bottom-right), plus three differential vectors forward in time, which we refer to as trajectories in the following. See Fig. 5.5 for a schematic. These seven differential vectors

result in 21 additional dimensions that are added to the three color channels for synthesis.

Trajectory Scaling. The synthesis algorithm can place patches at different radii than they were taken from in the exemplar. Even though this radius difference is limited to $\pm 10\%$, we still need to adjust for the fact that deformation trajectories closer to the pupil are longer than trajectories closer to the limbus (see Fig. 5.5 for a schematic explanation). Therefore, we scale the difference vectors of each trajectory by

$$\rho = \frac{r_l - r_{to}}{r_l - r_{from}}, \tag{5.9}$$

where r_{from} is the radius at which the patch is extracted and r_{to} the radius where it is placed. r_l is the limbus radius at which we assume no deformation.

Reconstruction. The synthesized differential vectors in the final iris texture are assembled to a linear Laplacian system for generating the final iris geometry similarly to Sorkine et. al [2004]. Since all vectors are relative, the system is under-constrained and we need to provide some additional constraints. The most natural choice is to constrain the positions of the pupil, which ensures a faithful fit to the observed pupil. Since the geometry is encoded in cylindrical coordinates, we need to scale the angular dimension (radians) to render it compatible with the radial (mm) and height (mm) dimensions. Thus, the angular dimension is multiplied by 5 mm, which corresponds to the average iris radius present in the database.

The result of this section is an iris model parameterized by a low-resolution control map, which allows high-resolution geometric and texture reconstructions using constrained synthesis given the database of eyes as exemplars.

5.4 Sclera Vein Model

Finally, we complete the eye by presenting a model for synthesizing the sclera. By far the most dominant features of the sclera are the veins, which contribute substantially to the visual appearance of the eye. Depending on the physical and emotional state, the appearance of these veins changes. For example they swell when the eye is irritated or when we are tired, causing

the infamous "red eye" effect. Veins also travel under the surface at varying depths, and deeper veins appear thicker and softer, while veins at the surface appear more pronounced.

We propose to model the different states of veins with a parametric vein model. Such a model allows us to continuously change parameters and blend between different states. Also, besides modeling changes we can create additional detail not visible in the input data. Our model grows veins from seed points following a parameter configuration. In the following, we first describe our vein model and how the vein network is synthesized, and then describe how the synthesized veins are rendered to create the sclera texture, including a synthesized normal map to provide fine-scale surface details.

5.4.1 Vein Model

When scrutinizing the veins of a real sclera, one can see that they exhibit an enormous visual richness and complexity (see Fig. 5.6.d), caused by the superposition of a large number of veins with varying properties such as color, thickness, scale, shape, and sharpness. To resemble this complex structure, we model the sclera vein network as a forest, where an individual tree corresponds the the vein network generated from a single large vein. The large veins are the most salient structures when looking from afar, and will be referred to as *primary level* veins. These veins branches off into smaller second, third and lower level veins. Similar to L-Systems [Rozenberg and Salomaa, 1976] and tree growing methods [Palubicki et al., 2009; Sagar et al., 1994] we create the vein network based on a set of rules, which control the vein properties described next.

Vein Properties. A single vein is represented by a series of control points, which are interpolated with a spline to provide a smooth and continuous curve in the texture domain. These positional control points govern the shape of the vein. Similarly, other spatially varying properties can also be discretized along this spline and interpolated when required. The properties we synthesize include position offsets along the curve normals, the vein thickness, the vein depth, which relates to its visibility, and vein branching points. Note that the discretization is independent per property, as some properties vary more quickly when traversing a vein network. To account for the irregularity present in nature, we define the properties with a certain amount of randomness. We employ two types of random functions, where one follows a Gaussian distribution $\mathcal{N}$, parameterized by the mean

and standard deviation. The second one is a colored noise function $\mathcal{C}$, which is parameterized by the amplitude controlling the amount of perturbation and the spectral power density, which is controlled by the exponent in $1/f^x$ and specifies the noise color.

The position offsets are defined by the colored noise function $\mathcal{C}_{offset}$ in the range of pink ($x = 1$) and red ($x = 2$) noise. Thickness is specified at the starting point $\rho_{thickSeed}$, along with a decay factor $\rho_{thickDecay}$, again perturbed with a colored noise function ($\mathcal{C}_{thick}$). Depth is computed as an offset to a given average depth ρ_{depth}, created by adding colored noise ($\mathcal{C}_{depth}$). Finally, the locations of the branching points and the corresponding branching angles are determined by two Gaussian distributions, $\mathcal{N}_{branchPos}$ and $\mathcal{N}_{branchAngle}$, respectively.

Vein Recipes. Our model allows us to generate veins given a set of parameters (a *vein recipe*) describing the properties of the vein (width, depth, crippling, noise, etc.). Multiple veins can be synthesized with the same recipe. However, a single recipe does not account for the variation observed in nature. Therefore, we empirically create multiple vein recipes that describe veins belonging to different branching levels (primary, secondary, tertiary). . We found that a set of 24 recipes (10 primary, 6 secondary, and 12 tertiary) can produce vein networks of adequate visual complexity. In addition to the parameters described above, the recipes will also prescribe the parameters used for vein growing described below.

Vein Growing. Vein synthesis takes place in an unwrapped texture domain, with the limbus at the top and the back of the eyeball at the bottom. Veins on the sclera grow from the back of the eyeball to the front, and hence we grow them from bottom to top.

Growth is governed by a step size ρ_{step} and a direction $\mathbf{d}$ at every point. The step size is attenuated during growth by a decay factor $\rho_{stepDecay}$. The growing direction is influenced by three factors, 1 - a Gaussian distribution $\mathcal{N}_\beta$ that provides a general growth bias towards the top of the domain, 2 - a second Gaussian distribution $\mathcal{N}_\gamma$ that controls how much the vein meanders, and 3 - a repulsion term that discourages veins from growing over each other. The repulsion term stems from a repulsion map $\mathcal{R}$ that is computed while growing the veins, by rendering the veins into an image, indicating that a particular area has become occupied. The best growing angle can be computed with the distributions defined above as

$$\alpha = \max_{\alpha}(\mathcal{N}_\beta(\alpha) + \varepsilon)(\mathcal{N}_\gamma(\alpha) + \varepsilon)(1 - \mathcal{R}(\mathbf{x} + \mathbf{d}) + \varepsilon). \tag{5.10}$$

The direction $\mathbf{d}$ is computed from the angle α and current step size, and $\mathbf{x}$ denotes the current position. Also, $\mathcal{N}_\gamma$ is evaluated relative to the last growing angle. Since the terms could fully deactivate each other in pathological cases, we add a ε to the terms ($\varepsilon = 0.001$).

Vein Seeds. Veins start growing at seed points at the bottom of the texture for primary veins, or at branching points for higher levels, and growing is terminated if they reach a predescribed length or grow past the limbus. The primary vein seeds are generated at random positions at the bottom of the texture. We use 10 seeds in all our results. The seeds are generated sequentially. To prevent that two seeds are too close to each other we reject the seeds that are closer than 300 pixels. The final texture is 4096 by 2048 pixels.

5.4.2 Vein Rendering

Given a synthesized network of veins with spatially varying properties, we now render the veins into the texture using an appearance model learned from the database of eyes.

The appearance of a vein is influenced by many different factors, such as its diameter, how shallow it grows, its oxygenation, and others. The most important factor is the depth, which influences the color since the sclera has a higher absorption coefficient in the red wavelengths and as a consequence deeper veins appear blueish. The depth also influences the sharpness of the vein, since more subsurface scattering blurs out the cross-profile. Next to depth, thickness plays a central role since thin and thick veins are visually quite different. Note that the two parameters are not independent, since thin veins for example can only appear close to the surface as they would be washed out further in, and consequently have to be of reddish color. We use a data-driven approach to map depth and thickness to appearance, determined from exemplary textures in the eye database, as described in the following.

Cross-Section Model. We manually label 60 short vein segments in exemplary textures, which span the vein appearance. From these segments we sample cross-section profiles of the RGB space by fitting an exponential along the profile:

$$\mathbf{c}(r) = \mathbf{c}_{bkgnd} - \delta \exp\left(\frac{-\|r\|_1}{2\psi}\right), \tag{5.11}$$

where r is the distance from the labeled vein along the profile, in pixels. The fitting estimates thickness ψ, depth δ and background color $\mathbf{c}_{bkgnd}$ of these cross-sections. Subtracting the background from the cross-section will allow us to add the model to any background.

Given the synthesized thickness ψ and depth δ, we retrieve all samples with similar depth and thicknesses from the labeled veins, where similarity is computed as Euclidean *distances* on normalized tickness and depth values. A similarity threshold *th* is set to 1.1 times the distance to the third nearest neighbour. The retrieved cross-profiles are scaled to match to the query parameters, and the final cross-profile used for rendering is computed as their weighted average, where the weights are set to $1 - \frac{distance}{th}$.

This model allows us to compute a cross-section for any pair of thickness and depth parameters. Finally, the cross-section model is evaluated for each pixel in the neighborhood of a vein with the local width and depth, and added to the backplate.

Backplate. Our vein model describes the vein network but not the background into which the veins are to be rendered. This background contains two components: the low frequency variation and the high-frequency structure of the sclera texture. The mid-frequency features are provided by the vein model.

The high-frequency component accounts for visual noise and imperfections. This high-frequency texture is created manually by copying sclera patches that contain no veins from the database textures. Since it does not contain any recognizeable structures we can employ the same high-frequency components for every eye.

The low-frequency component is extracted from the smoothed input images with the intent to match the perceived overall color variation. Since only parts of the sclera texture can be computed from the images, we extrapolate the low-frequency component of the sclera to the entire eyeball by fitting a smooth spline surface to the visible parts of the texture. The spline surface is cyclic in the horizontal direction so that the left and right border match seamlessly. We also constrain the bottom of the texture to a reddish hue since there is no data present at the back of the eyeball and visually eyes appear more red near the back.

The high- and low-frequency components are combined into a single backplate image, into which the veins are rendered. An example of a final vein texture is shown in Fig. 5.6 (a), which additionally shows the impact of the depth (b,c) and thickness (e,f) parameters.

Geometric Surface Details. The geometric surface details of the sclera are important for the visual appearance of the eye since these little bumps affect the shape of specular highlights. The bumps consist of a mix of random bumps and displacements that correlate with the positions of big veins. Thus, we create a normal map based on a combination of procedural noise and displacements that follow the thick veins to render all our results.

This concludes the parametric model, which is able to synthesize all visible parts of the eye, including the eyeball, the iris, and the veins.

5.5 Model Fitting

The model described in the previous sections allows us to create a wide range of realistic eyes based on a few parameters and an iris control map. In this section we describe how these parameters can be estimated automatically and how the required iris control map is extracted from various sources. We focus on two different use-case scenarios. In a first use-case, we demonstrate how the proposed method may be used to complement existing photogrammetric face scanners to augment the facial geometry that is inaccurate for the eye itself with high-quality eye reconstructions, and in a second use-case we show how our method can be used to compute eye geometry and textures from single, uncalibrated input images.

5.5.1 Multi-View Fitting

In the multi-view scenario we fit our eye model to a 3D face scan provided by a multi-view stereo (MVS) reconstruction algorithm. In this work we leverage the system of Beeler et al. [2010], but any other system that provides calibrated cameras and 3D geometry would also work. The MVS algorithm reconstructs the white sclera reasonably well since its surface is mostly diffuse, albeit at lower quality than skin due to strong specular reflections which result in a noisier surface. Here, our model will serve as a regularizer to get rid of the noise. Most other parts of the eye, such as the cornea or the iris, pose greater challenge to the system, as they are either invisible or heavily distorted. Here, our model will fully replace any existing 3D data

and rely solely on the imagery to reconstruct geometry and texture. In the following we will describe fitting of the model to a single or even multiple face scans with different eye gazes simultaneously.

Eyeball Fitting. The input images are annotated by labelling the limbus (red), the pupil (black), the sclera (white), and the iris (green) as shown in Fig. 1.3. Manually labelling these features is quick and could potentially be automated with existing eye detection techniques. Based on the input mesh and the labels we estimate the parameters for each eye. Specifically, we estimate the rigid transformation, the scale, the coefficients of the deformable model, as well as the radius and position of the pupil, yielding a total of 14 unknowns for a single eye. The orientation of the pupil is constrained by our model to the average pupil orientation of the database. Fitting is based on four weighted energy terms, which form the total energy E_{total} to be minimized:

$$E_{total} = \lambda_s E_{\text{sclera}} + \lambda_l E_{limbus} + \lambda_p E_{pupil} + \lambda_c E_{coeff}. \tag{5.12}$$

The sclera energy term (E_{sclera}) penalizes the distance between the model mesh $\mathcal{M}$ and the sclera mesh $\mathcal{Z}$ from the face scan, and is defined as

$$E_{sclera} = \frac{1}{|\mathcal{Z}|} \sum_{\mathbf{x}_i, \mathbf{n}_i \in \mathcal{Z}} \left\| \langle (\mathbf{x}_i - \chi(\mathbf{x}_i, \mathcal{M})), \mathbf{n}_i \rangle \right\|^2, \tag{5.13}$$

where $\mathbf{x}_i$ are the sclera mesh points and their closest points on the model are $\chi(\mathbf{x}_i, \mathcal{M})$. Distance is only constrained along the normal $\mathbf{n}_i$, which allows tangential motion. The sclera mesh is segmented from the full face mesh using the sclera and limbus annotations.

The limbus energy term (E_{limbus}) penalizes the distance between the projection of the model limbus into the viewpoint and the limbus:

$$E_{limbus} = \frac{1}{|\mathcal{L}^{\mathcal{S}}|} \sum_{\mathbf{y}_i \in \mathcal{L}^{\mathcal{S}}} \left\| \mathbf{y}_i - \phi(\mathbf{y}_i, \mathcal{L}^{\mathcal{M}}) \right\|^2, \tag{5.14}$$

where $\mathbf{y}_i$ are the limbus annotations and their closest points to the projected model limbus are $\phi(\mathbf{y}_i, \mathcal{L}^{\mathcal{M}})$.

Similarly, the pupil energy term (E_{pupil}) penalizes deviation of the projected model pupil from the pupil annotations. Unlike the limbus energy, this energy has to take into account the refraction taking place at the cornea inter-

face when projecting the pupil into the camera. For the refraction computation we use a continuous spline approximation of the cornea surface.

The last term corresponds to the coefficient term defined in Equation 5.4. All terms are weighted equally, i.e. all lambdas are set to 1.

Since this is a highly non-linear energy, we optimize it iteratively following an Expectation-Maximization (EM) schema. In the E-step we recompute all the correspondences based on the current estimate of the model, and in the M-step we fix the correspondences and optimize for the parameters using the Levenberg-Marquart algorithm. Typically, the optimization converges in about 5 iterations.

Iris Control Map. The optimization above yields the eyeball geometry and a disk centered at the fitted pupil, which will serve as proxy to compute the iris control map. As this disk only approximately corresponds to the real iris geometry, each view will produce a slightly different iris texture. Since the cameras of the MVS system frame the full head and lack resolution in the eye area, we employ two zoomed in cameras to compute the iris texture. From the two, we manually select the one producing the sharpest texture as our primary view. The other view is used to inpaint the highlights only. The algorithm computes a highlight probability using the method of Shen et al. [2009] for each view and combines the iris texture maps according to

$$C = \frac{C_p w_p + C_s(1 - w_p)w_s}{w_p + (1 - w_p)w_s}, \tag{5.15}$$

where C_p and C_s are the colors of the primary and secondary textures, and w_p and w_s are the highlight confidence maps. As discussed in Section 5.3, the resolution of the control map depends on the resolution of the input images. In our particular setup, the resolution of the control map in polar coordinates is 256x64 pixels.

Eye Pair Fitting. The properties of a pair of eyes are typically highly correlated, as was also shown in our eye reconstruction work. This correlation can be leveraged to reduce the dimensionality of the fitting task from naïvely 28 dimensions to 21. Since it is reasonable to assume that the eyes have a similar (but antisymmetric) shape we can use the same shape coefficients and scale for the second eye. Furthermore, the rigid transformation of the second eye is linked to the first and can be reduced from 6 to 3 degrees of freedom, one for the vergence angle and two for the inter-ocular vector. The remaining

parameters are then pupil radius and position, which may differ between the two eyes.

Multi-Pose Fitting. If we assume that the shape of the eyeball is rigid, we can leverage multiple eye poses to better constrain the optimization. The shape coefficients and global scale, as well as the inter-ocular vector are then shared amongst all poses, as are the pupil positions. Fig. 5.7 shows an example of multi-pose fitting, where we jointly optimize the parameters based on three poses.

5.5.2 Single Image Fitting

Fitting our eye model to a single image is much less constrained than the multi-view scan fitting since less data is available. Neither can we rely on depth information nor do we have multiple views to constrain the optimization. Still, by making some assumptions we are able to extract plausible model parameters for a given image.

The optimization for single image fitting is based on the same energy formulation as the multi-view case, described in Equation 5.12, but since we do not have 3D information, the sclera term is removed. Thus the proposed method requires just limbus and pupil annotations, and relies stronger on the model prior. For example, we fix the scale of the eye to 1 due to the inherent depth/scale ambiguity in the monocular case. Furthermore, we rely on the position of the model pupil and optimize for pupil radius only. To project the limbus and pupil into the image, the method requires a rough guess of the camera parameters (focal length, and sensor size), which can be provided manually or extracted from meta-data (EXIF).

5.6 Results

In this section we will demonstrate the performance of the proposed method on a variety of input modalities, ranging from constrained multi-view scenarios to lightweight reconstruction from single images. Before showing fitting results, we will demonstrate the benefits of the parametric eye model for manipulation.

The appearance of the vein network in the sclera varies as a function of the physiological state of the person, leading to effects such as red eyes caused

by fatigue. The proposed parametric vein model can account for such effects (and others) as shown in Fig. 5.8, where we globally change the depth at which veins grow from shallow (a) to deep (b), which influences their visibility, as well as the overall vein thickness from thick (c) to thin (d).

Since we do reconstruct the complete dilation stack of an iris, the pupil size can be manipulated to account for virtual illumination conditions or to simulate some physiological effects, such as hippus, which is an oscillation of the pupil diameter. Fig. 5.9 shows three different irises at four different dilation stages. Our method nicely models the geometric detail that varies as the the pupil dilates (left to right). The three irises differ in color, ranging from brown (top) to blue (middle) to dichromatic (bottom). One can clearly see the different surface structure, which is inherently linked to the color, with brown irises being smoother and blueish more fibrous. Since our method generates the structure as a function of the iris color, one can indirectly control the structure by changing the color of the iris. In the special case of the dichromatic iris (bottom), the method produces structural details that vary spatially and match the color. The dichromatic iris corresponds to the right iris in Fig. 5.10.

Fig. 5.11 shows reconstruction results on a variety of different eyes, all captured in the multi-view setup and reconstructed using multi-view fitting. The eyes exhibit varying iris color and structure, eyeball shape and sclera vein networks. Since we operate on the same input data as multi-view face reconstruction algorithms, namely a set of calibrated images, our method seamlessly integrates with existing facial capture pipelines and augments the face by adding eyes, one of the most critical components, as can be seen in Fig. 5.1.

Fig. 5.12 demonstrates the robustness of our method. It shows the result of a single image fit and the effect of reducing the resolution of the input image by a factor of 35. Credible high-frequency detail missing in the low-resolution image is synthesized by our method to produce similar quality outputs.

Fig. 5.13 shows a comparison of our eye reconstruction method with our lightweight fitting approach for an eye not contained in the eye database. The results are generated from the same multi-view data from which also the image for the comparison in Fig. 5.12 stems. Despite fitting the model to just a single pose our approach produces results which are very close to the more laborious eye reconstruction method. The mismatch of the back parts

of the eyeballs is of little significance since neither of the methods produces an accurate reconstruction of these hidden parts.

The computational effort required to reconstruct an eye is about 20 minutes. The most time intense parts are the iris synthesis and the reconstruction of the Laplacian system formed by the iris stack. Labelling a single image takes about 3 minutes, which is the only user input required.

Being able to reconstruct eyes from single images as shown in Fig. 5.14 provides a truly lightweight method to create high-quality CG eyes, not only from photographs but also from artistic renditions, such as sketches or paintings and even extending beyond human eyes, as shown in Fig. 5.15.

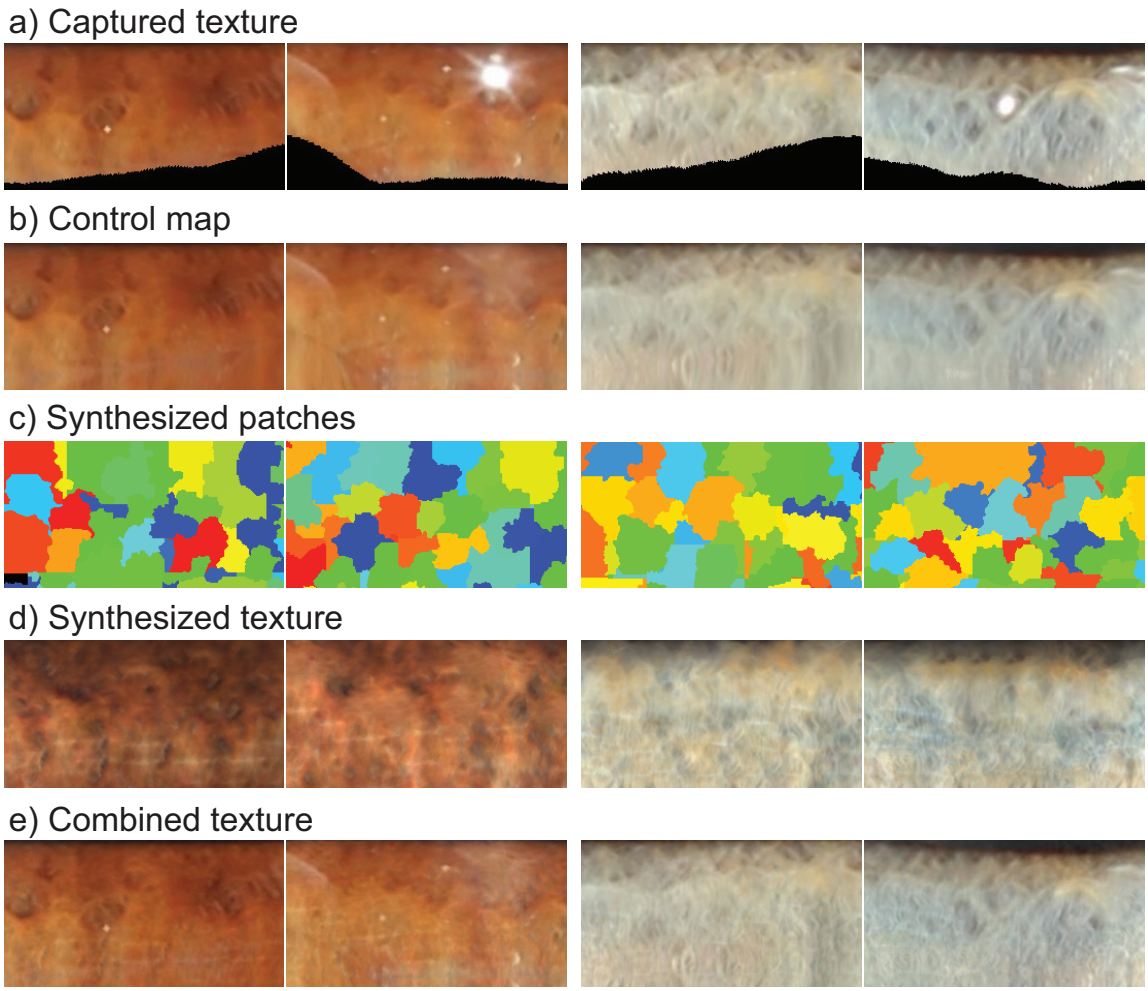

Figure 5.4: *Synthesizing an iris consists of capturing initial textures (a), from which control maps are generated by removing specular highlights (b). This control map is input to a constrained texture synthesis that combines irregular patches (c) from the database to a single texture (d), which is then filtered and recombined with the control map to augment the low-res control map with high-frequency detail (e). The figure shows close-ups from a brown iris on the left and from a blueish iris on the right.*

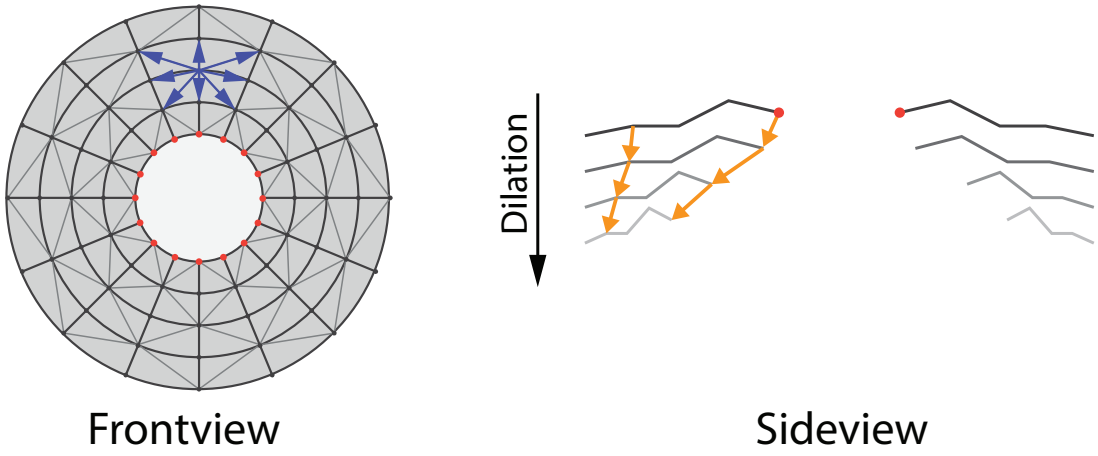

Figure 5.5: *The iris geometry is tesselated uniformly in the polar domain, yielding eight neighbors per vertex spatially (blue, left) as well as the forward neighbors in time (orange, right), which describe the trajectory a vertex follows during dilation. The trajectory is longest at the pupil and has to be properly scaled during synthesis.*

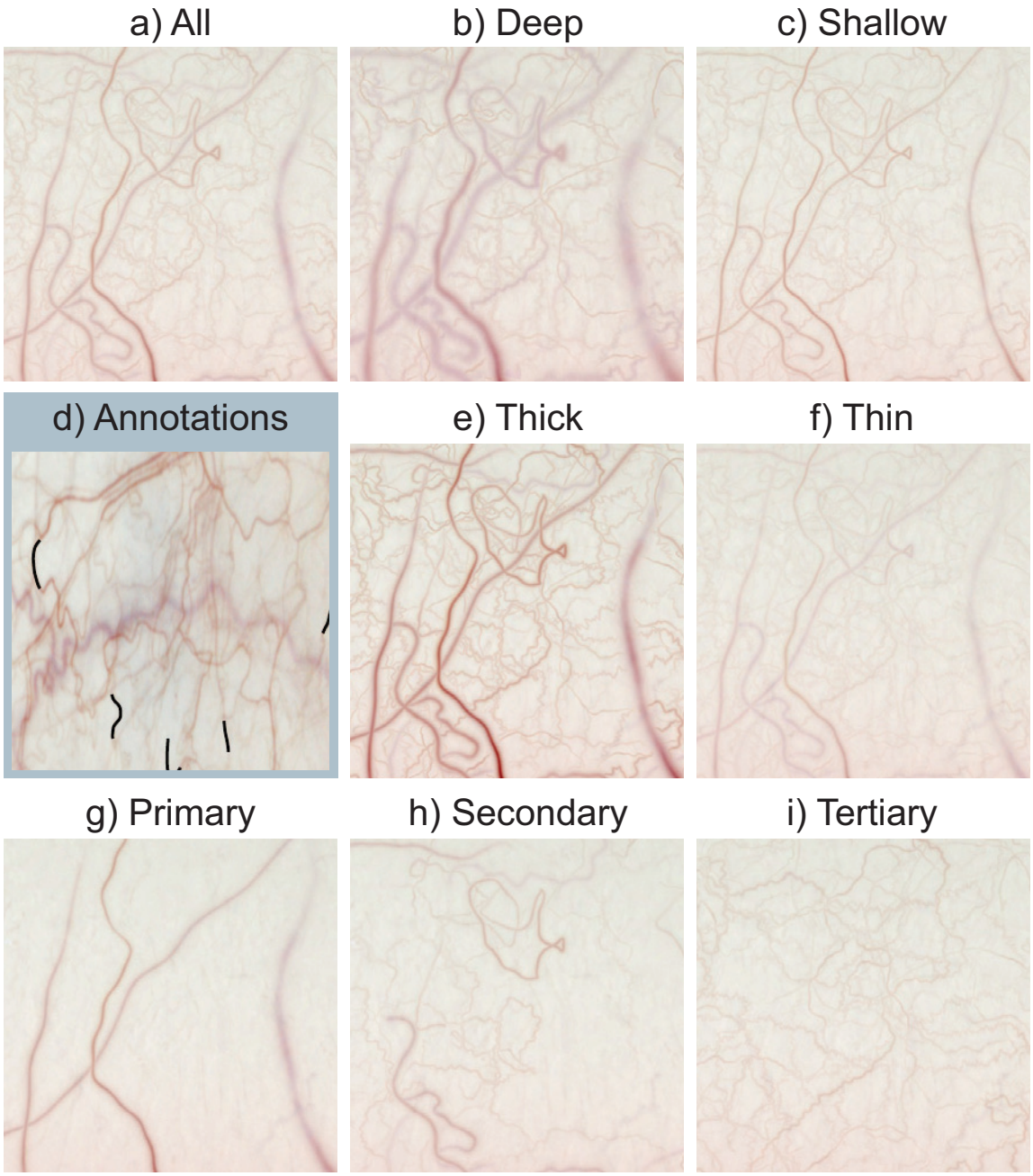

Figure 5.6: *Our parametric vein model allows the manipulation of the appearance of the veins (a) using a parameter for thickness and one for depth. The vein appearance is computed from an annotated exemplar texture (black segments in d), and our parametric vein model allows to independently manipulate depth (b,c) and thickness (e,f) to control the appearance. Veins are defined by different vein recipes for the three different level (g,h,i).*

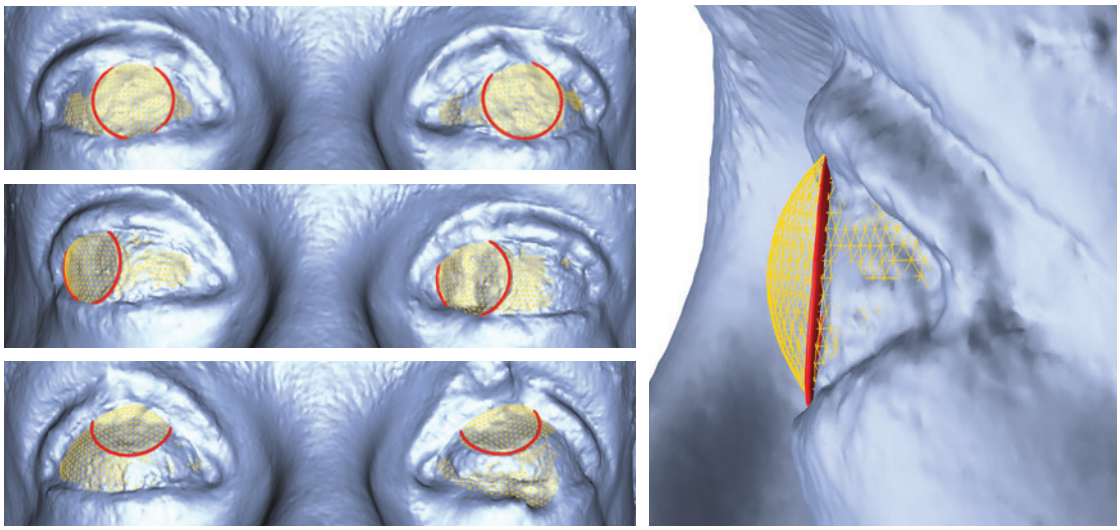

Figure 5.7: *We can leverage multiple eye poses to better constrain the fitting optimization. Here we fit simultaneously to three poses.*

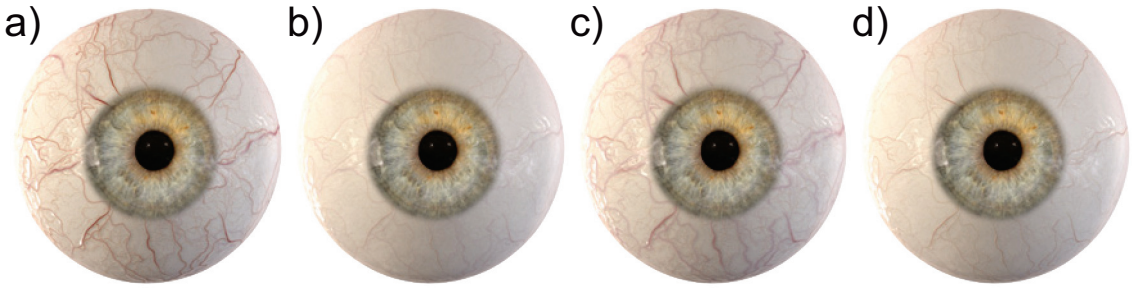

Figure 5.8: *Since we can parametrically control the sclera vein network and appearance, we can simulate physiological effects such as red eyes due to fatigue. Here we globally change the depth at which the veins grow from shallow (a) to deep (b), as well as their thickness from thick (c) to thin (d).*

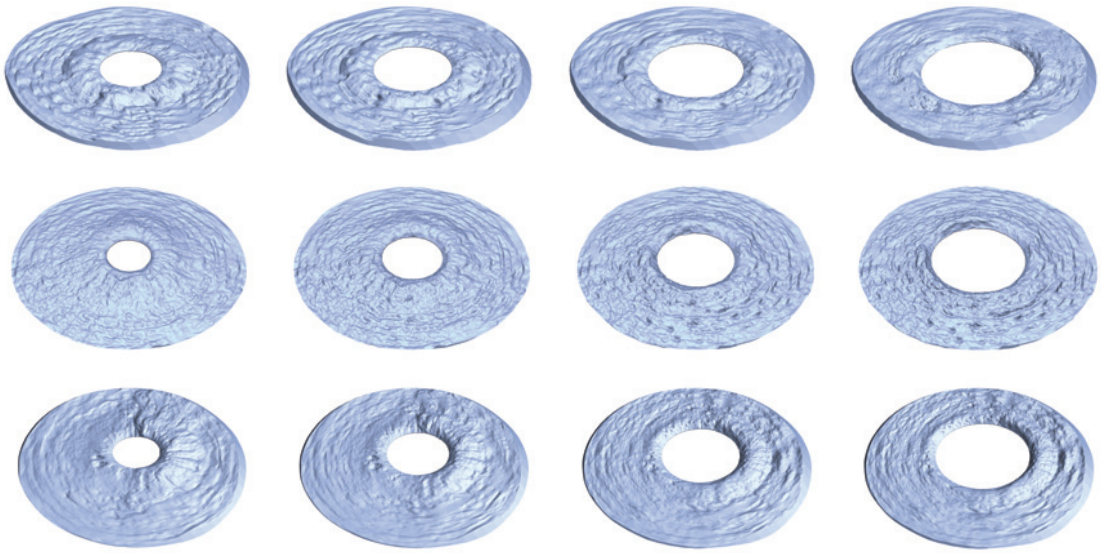

Figure 5.9: *Since geometric detail is inherently linked to the color of an iris, we can synthesize realistic microstructure, ranging from smooth for brown (top) to fibrous for blueish irisis (center). The bottom row shows a dichromatic iris that mixes gray-green and red-brown colors, which is clearly visible in the synthesized structure.*

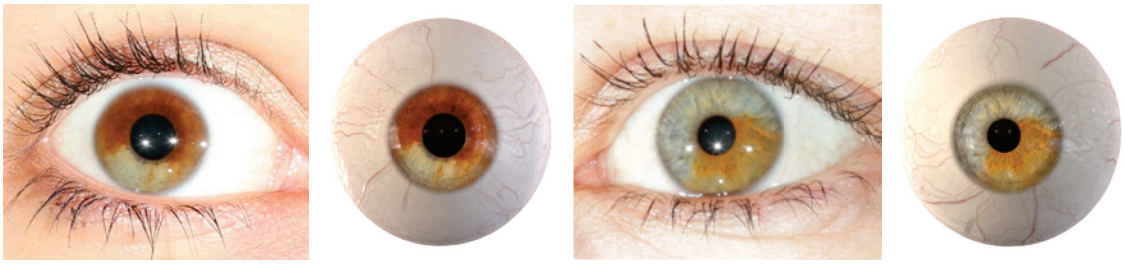

Figure 5.10: *Our method is able to reconstruct complex dichromatic irises by combining different color exemplars from the database.*

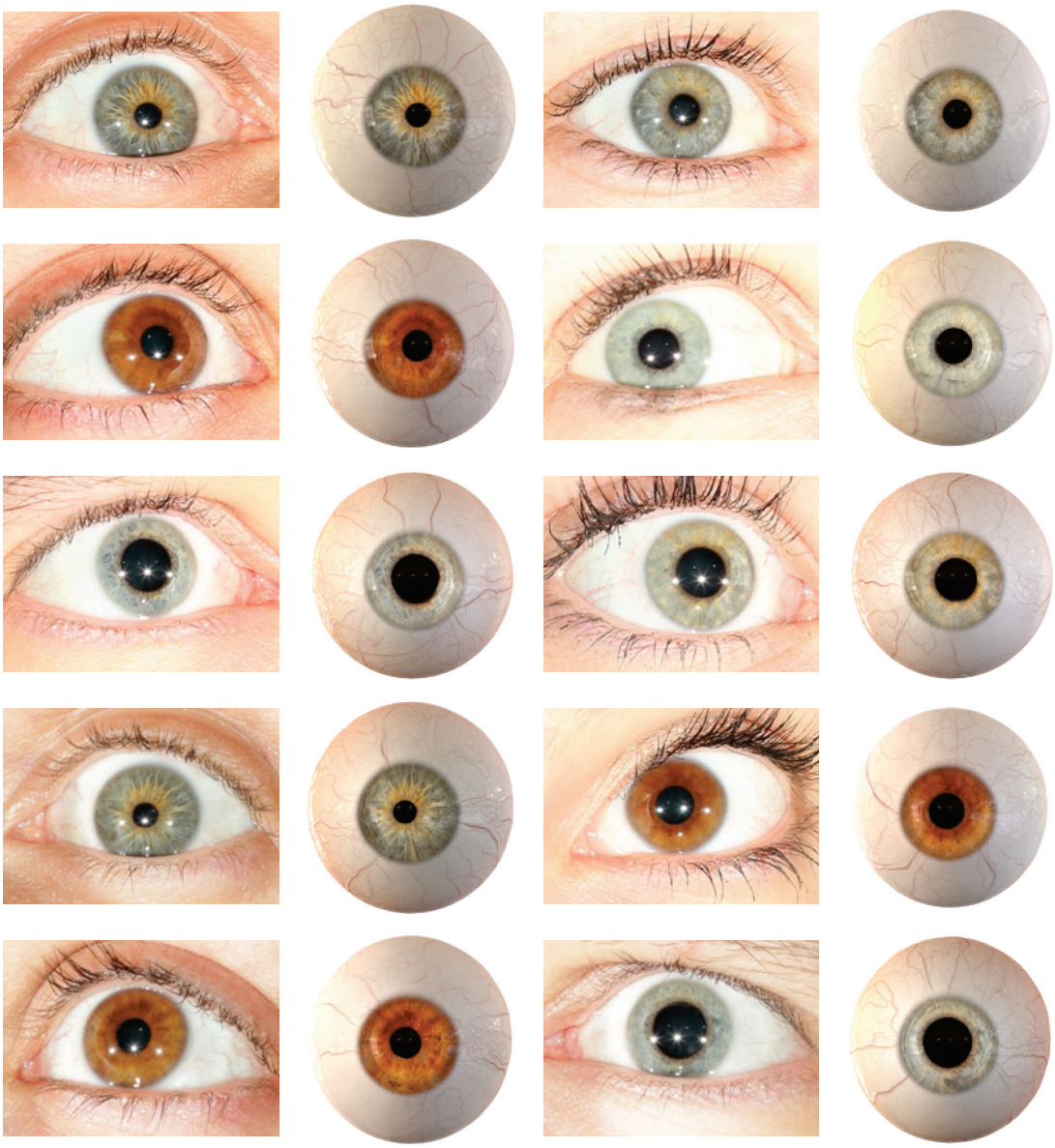

Figure 5.11: *Our method can reconstruct a variety of different eyes with varying eyeball shape, iris structure and color, and synthesize realistic scleras with vein textures and surface details.*

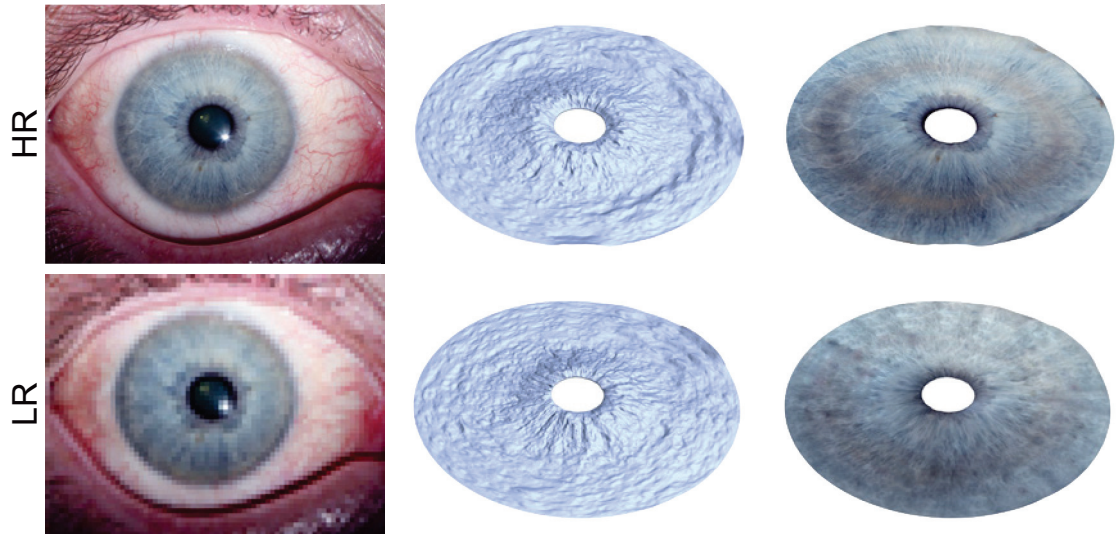

Figure 5.12: *We demonstrate the robustness of our method by fitting the eye model to a single high-resolution (HR) image and a low-resolution image (LR) obtained by down-sampling the first by a factor of 35. The figure shows the reference images (left), the reconstructed iris geometries (center), and the textured iris geometries (right).*

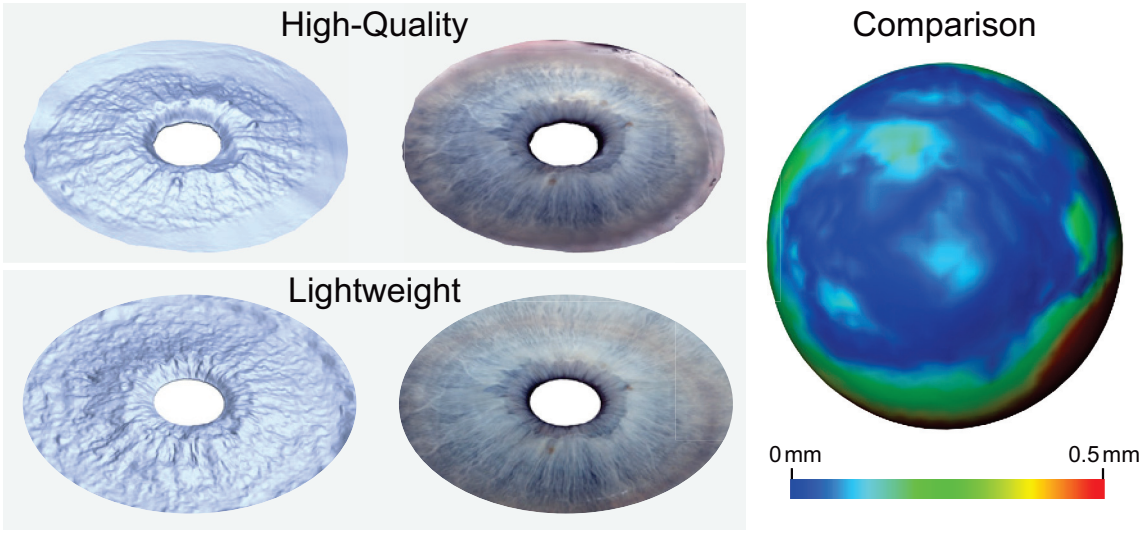

Figure 5.13: *Reconstruction comparison between our high-quality and lightweight methods. The figure shows the iris geometries (left) and textures (center) generated from the same multi-view data from which also the image for the comparison in Fig. 5.12 stems. The right side shows a comparison of the reconstructed eyeball meshes. The color map visualizes the error between the eyeball meshes of two methods.*

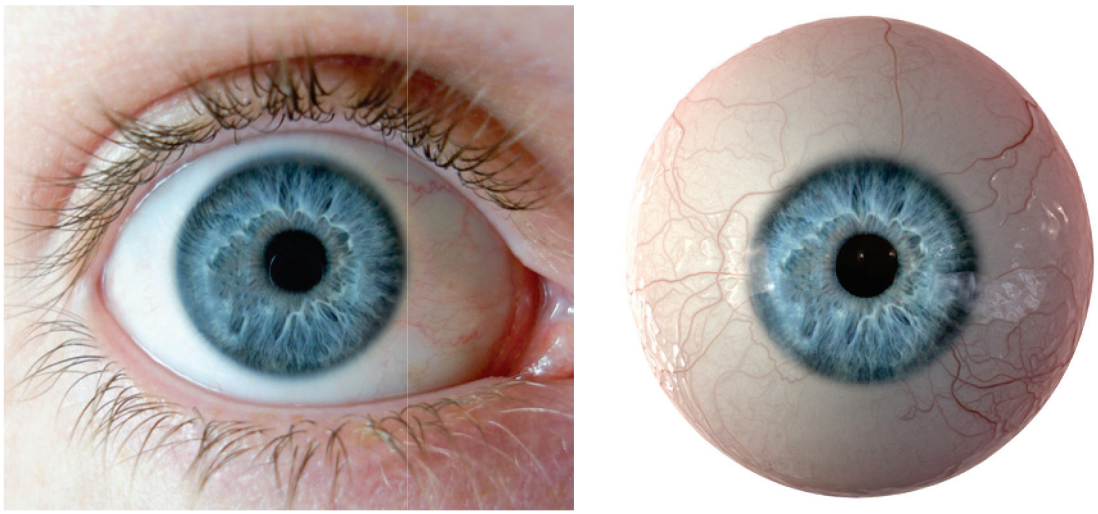

Figure 5.14: *The proposed method can fit eyes even to single images such as this one, opening up applications for eye reconstruction from internet photos. Source: [Wikimedia Commons, 2006].*

Figure 5.15: *We show the robustness of our method by fitting eyes even to artistic paintings and single images of animals. Sources: [Wikimedia Commons, 1887; Wikimedia Commons, 1485; Flickr, 2006].*

CHAPTER 6

Eye Rigging

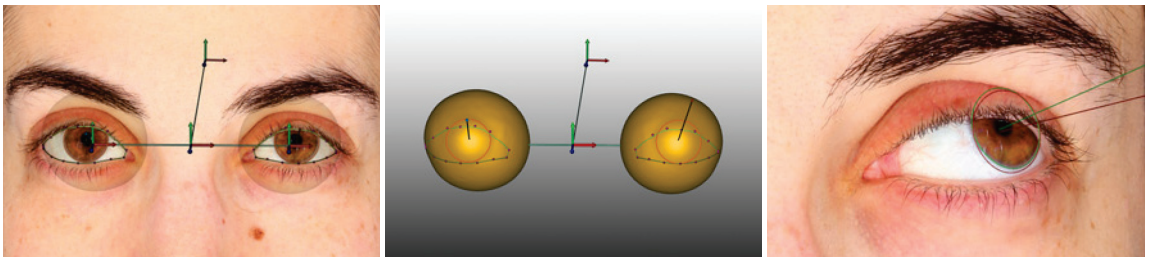

Figure 6.1: *We present a new physiologically accurate eye rig based on accurate measurements from a multi-view imaging system and show where assumptions often made in computer graphics start to break down. This figure shows an input image with the eye rig overlaid, the eye rig, and a comparison between our (green) and a simple eye rig (red) traditionally used in computer graphics (left to right).*

In Chapter 5 we present a novel parametric eye model. This eye model allows for the quick and robust generation of realistic eyes from as little as a single image. This model, however, is limited to the eyeball and does not model the movement nor the positioning of the eyes within the head.

In this chapter we present a parametric eye rig and a method to estimate its person-specific parameters from images as an extension to the parametric eye model introduced in Chapter 5. We define this novel eye rig in Section 6.1. In Section 6.2 we describe the image capture setup that we need for estimating the rig parameters. This estimation has two phases. First, we fit the eyeball shape and per frame pose (Section 6.3) and second we fit the actual rig (Section 6.4). We validate the effectiveness of the presented rig and

investigate the importance of accurate eye motion modeling in the context of computer animation (Section 6.5).

6.1 Eye Rig

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

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.

6.1.1 Eye Shape

Fig. 3.1 (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 intruduced in chapter Chapter 5. 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), and so in order to allow for simple and 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 $\dot{\bar{\mathcal{P}}}$, any point $\mathbf{x}_{uv} \in \mathbb{R}^2$ in texture space can be transformed to a point $\mathbf{x}_{world} \in \mathbb{R}^3$ in world space via

$$\mathbf{x}_{world} = Eyeball(\mathbf{x}_{uv}, \dot{\bar{\mathcal{P}}}), \tag{6.1}$$

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

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 $\mathbf{x}_{uv} \in \mathbb{R}^2$ in texture space as

$$\mathbf{x}_{uv} = Limbus(x_{ctr}). \tag{6.2}$$

Pupil Our parametric eye 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 static 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 vary per pose and are thus dynamic.

Visual axis The gaze direction of an eye does not coincide with the optical axis, but with the visual axis of the eye, which 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 static parameters, a symmetric polar angle and antisymmetric azimuth that provide the main directions, coupled with an antisymmetric polar angle 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 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. 6.2. The control points are constrained to lie on equidistant lines perpendicular to the horizontal line connecting 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 $\dot{\tilde{\mathcal{P}}}$, we define the mapping of points $x_{ctr} \in \mathbb{R}^1$ on the spline to points $\mathbf{x}_{uv} \in \mathbb{R}^2$ in texture space as

$$\mathbf{x}_{uv} = Eyelid(x_{ctr}, \dot{\tilde{\mathcal{P}}}). \tag{6.3}$$

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.

6.1.2 Eye Motion

As depicted in Fig. 3.1 (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 thesis and we refer the interested reader to medical textbooks [Carpenter, 1988]. 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

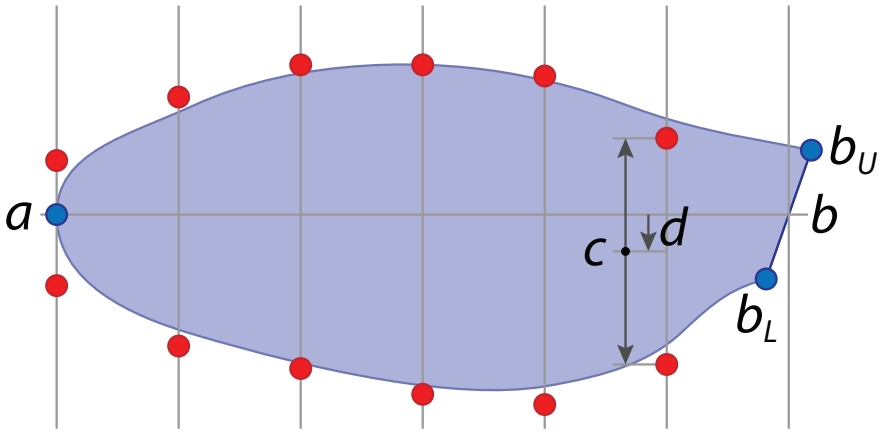

Figure 6.2: *The eyelid interface consists of two B-spline curves (from a to b_U and a to b_L) defined by their control points (red and blue). The blue control points can move freely. The middle control points (red) are equally distributed on the middle line connecting the eye corner (a) and the tear 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).*

that the eye rotates only horizontally and vertically around a static pivot is incorrect. In reality the eye not only exhibits rotation around all axes, but also translates within its socket during rotation [Fry and Hill, 1962].

Rotation We model the eye rotation $\dot{\boldsymbol{\Theta}}$ based on a Helmholz gimbal with three degrees of freedom (up/down=$\dot{\Theta}_x$, right/left=$\dot{\Theta}_y$, torsion=$\dot{\Theta}_z$). According to Donders' law, for a given gaze direction $(\dot{\Theta}_x, \dot{\Theta}_y)$ the torsion angle $\dot{\Theta}_z$ 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. [1993]. 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 ($\bar{\Theta}_x$,$\bar{\Theta}_y$)

$$\dot{\Theta}_z = \mathcal{L}\left(\dot{\Theta}_x - \bar{\Theta}_x, \dot{\Theta}_y - \bar{\Theta}_y\right). \tag{6.4}$$

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 [1962; 1963] 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, and as shown in Fig. 6.11 our measurements match theirs very well. Unfortunately, to the best of our knowledge, no model that predicts eye translation over the entire range has been developed to date. Based on theirs and our measurements we hence suggest the following model to account for translation.

To account for the translational motion of the eye, we introduce the function $\boldsymbol{\Psi}(\cdot)$ that adds a pose dependent offset to the person-specific rotational pivot $\bar{\mathbf{p}}$ for a given eye gaze:

$$\dot{\mathbf{p}} = \boldsymbol{\Psi}\left(\dot{\Theta}_x, \dot{\Theta}_y\right) + \bar{\mathbf{p}}. \tag{6.5}$$

Based on our measurements (Fig. 6.10) we model $\boldsymbol{\Psi}(\cdot)$ as a bivariate quadratic function

$$\begin{aligned}\boldsymbol{\Psi}\left(\dot{\Theta}_x, \dot{\Theta}_y\right) = &(\bar{\boldsymbol{\alpha}}_0 \pm \bar{\boldsymbol{\delta}}_0) + (\bar{\boldsymbol{\alpha}}_1 \pm \bar{\boldsymbol{\delta}}_1) \cdot \dot{\Theta}_x + (\bar{\boldsymbol{\alpha}}_2 \pm \bar{\boldsymbol{\delta}}_2) \cdot \dot{\Theta}_y \\ &+ (\bar{\boldsymbol{\alpha}}_3 \pm \bar{\boldsymbol{\delta}}_3) \cdot \dot{\Theta}_x^2 + (\bar{\boldsymbol{\alpha}}_4 \pm \bar{\boldsymbol{\delta}}_4) \cdot \dot{\Theta}_y^2 \\ &+ (\bar{\boldsymbol{\alpha}}_5 \pm \bar{\boldsymbol{\delta}}_5) \cdot \dot{\Theta}_x \cdot \dot{\Theta}_y,\end{aligned} \tag{6.6}$$

where the $\bar{\boldsymbol{\alpha}}_{0-5}$ parameters are symmetric between the left and right eye, and the $\bar{\boldsymbol{\delta}}_{0-5}$ are antisymmetric.

6.1.3 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. 6.3.

World $\rightarrow$ Skull A first step is to model the head motion. This will require one rigid transformation per frame $\dot{M}_{world \rightarrow skull}$, which can be given by animation curves or estimated from captured data (e.g. [Beeler and Bradley, 2014]).

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 $\bar{\mathbf{p}}$, and the x-axis rotation is kept identical with the x-axis rotation of the skull. The pair coordinate frame is person-specific but static 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 static 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.

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.

6.1.4 Eye Control

Once fit to a person (Section 6.4), 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 lookat point). Furthermore, the opening of the pupil can be controlled by a single user parameter during animation.

6.2 Data Acquisition

In order to develop our eye rig we depend on high-quality data of real eye motion. We employ a multiview 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. [2010]. 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,

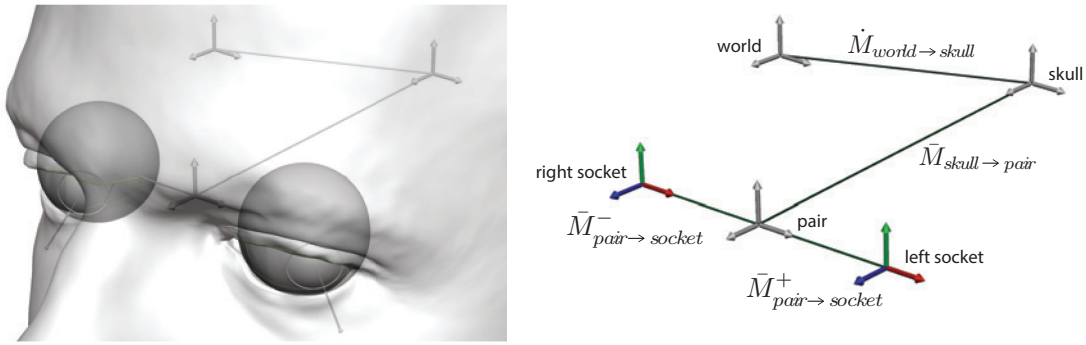

Figure 6.3: *The proposed rig rotates and offsets the eye relative to its socket. The left and right sockets are defined via antisymmetric transformations relative to the joint pair coordinate frame, which in turn is relative to the coordinate frame of the skull. While all of these transformations are static, the skull moves relative to the world coordinate frame over time.*

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 [Beeler et al., 2011] and compute the underlying skull pose using a rigid stabilization technique [Beeler and Bradley, 2014]. Our setup is shown in Fig. 6.4.

We further record the 3D look-at point for each pose using an HTC Vive tracking system[1]. We modified one of the Vive controllers by adding a small light bulb, which the subject is instructed to fixate on during acquisition. To register the camera coordinate frame with the coordinate frame of the Vive, we need to compute the rigid transformation that aligns them. This is a trivial task if we have a set of point correspondences in each coordinate frame. To this end, we record a series of points with the Vive, and for each point we reconstruct the 3D position in camera space by detecting the light bulb in the cameras and triangulating. Since the light bulb position does not coincide exactly with the tracking point of the vive controller, we also capture the controller at various orientations, allowing us to solve for the light bulb offset as part of the coordinate frame transformation.

To add robustness outside the working volume of the cameras, we also record the position of the cameras with the tracked controller. Since we cannot move the controller to the center of the camera we record a point on the camera lens axis, close to the back of the camera, and account for this one-

[1] www.vive.com

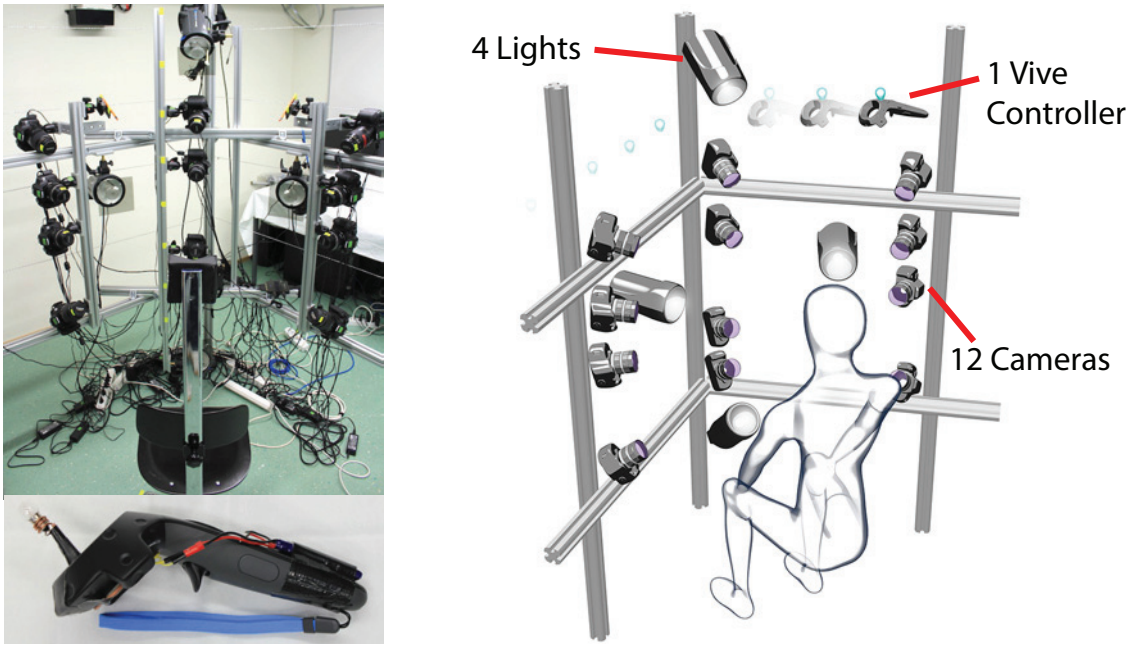

Figure 6.4: *Our capture setup consists of 12 DSLR cameras and 4 industrial light flashes, providing synchronized multi-view imagery of static eye poses. We modified an HTC Vive Controller by adding a small light bulb, which the subject fixates on during acquisition, giving ground truth 3D look-at points.*

dimensional offset when solving for the transformation between the camera and Vive coordinate frames.

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 three different subjects.

6.3 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 6.2. 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 6.4.

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 introduced in Chapter 5, which was itself generated from measured data acquired with the system presented in Chapter 4. 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 $\dot{M} \in \mathbb{R}^6$, for each pose of each eye. One viable option would be to estimate $\dot{M}$ independently per frame per eye without any a-priori knowledge. On the other hand, we do know certain things about eye movements from the medical literature, such as Listing's model, and it will be helpful to be able to rely on such information where possible. Therefore, we propose to actually use a subset of our rig introduced in Section 6.1 to reconstruct the individual eye poses of the dataset.

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 $\dot{\Theta}_z^\epsilon$ to Equation 6.4, yielding

$$\dot{\Theta}_z = \mathcal{L}\left(\dot{\Theta}_x - \bar{\Theta}_x, \dot{\Theta}_y - \bar{\Theta}_y\right) + \dot{\Theta}_z^\epsilon. \tag{6.7}$$

Secondly, we add a per-pose residual $\dot{\mathbf{p}}^\epsilon$ for the rotational pivot point in Equation 6.5, yielding

$$\dot{\mathbf{p}} = \mathbf{\Psi}\left(\dot{\Theta}_x, \dot{\Theta}_y\right) + \bar{\mathbf{p}} + \dot{\mathbf{p}}^\epsilon. \tag{6.8}$$

As we did not know the general shape of $\mathbf{\Psi}(\cdot)$ initially, we simply set it to $\mathbf{0}$. However, when reconstructing the poses for future subjects, one can use Equation 6.6 instead as a-priori information. These slack variables are weakly regularized to be small, incentivising the other variables to capture as much of the signal as possible and only represent the residual. Together with the gaze direction $(\dot{\Theta}_x, \dot{\Theta}_y)$, this amounts to six dynamic degrees of freedom per eye and allows to accurately reconstruct eye poses while still leveraging prior knowledge.

We obtain the eye configurations by fitting to manual annotations (Section 6.3.1), which makes fitting very robust. Automatic labelling is challenging due to the complexity of the eye region in terms of geometry and appearance. Manual annotations, however, are not pixel-perfect and therefore the fits contain errors. Thus, we refine the positions with photometric constraints (Section 6.3.2). The final eye configurations will be passed on to

Figure 6.5: *Example image annotations: limbus (yellow/turquoise), lower eyelid interface (red/blue), tear duct (orange/cyan), and pupil (brown/gray).*

our full rig fitting algorithm described in Section 6.4. We employ the Ceres solver [2018] to solve for optimal parameters $\mathcal{P}$.

6.3.1 Annotation Fitting

The eyeball positions are first fitted to image annotations. As shown in Fig. 6.5, 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 for which the feature is best visible. For each of these annotations we formulate a constraint, which together form the following optimization problem

$$E_{annotation} = E_{limbus} + E_{eyelid} + E_{shape} + E_{corners} + E_{pupil}. \tag{6.9}$$

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 $\mathbf{a}_i^{lim}$, 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 Equation 6.2 and Equation 6.1 the curve parameter c_i^{lim} is mapped to world space and then projected via $Camera(\cdot)$ into the image plane

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

Table 6.1: *Weights used to balance the individual energy terms.*

$$\mathbf{x}_i^{lim} = Camera(\ Eyeball(\ Limbus(c_i^{lim}),\ \dot{\mathcal{P}})\ )$$
$$E_{limbus} = w_{limbus} \cdot \frac{1}{n^{lim}} \cdot \sum_{i=1}^{n^{lim}} \left\| \mathbf{x}_i^{lim} - \mathbf{a}_i^{lim} \right\|^2, \tag{6.10}$$

where we compute the weighted L_2 norm. The weights for this and the other energy terms are tabulated in Table 6.1.

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 $\mathbf{a}_i^{lid}$ and their projected correspondences $\mathbf{x}_i^{lid}$:

$$\mathbf{x}_i^{lid} = Camera(\ Eyeball(\ Eyelid(c_i^{lid},\ \dot{\mathcal{P}}),\ \dot{\mathcal{P}})\ )$$
$$E_{eyelid} = w_{eyelid} \cdot \frac{1}{n^{lid}} \cdot \sum_{i=1}^{n^{lid}} \left\| \mathbf{x}_i^{lid} - \mathbf{a}_i^{lid} \right\|^2. \tag{6.11}$$

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 re-compute 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 $\mathbf{c}_{i-1}$, $\mathbf{c}_i$, and $\mathbf{c}_{i+1}$ 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

$$E_{shape} = w_{shape} \cdot \frac{1}{n^{shp}} \cdot \sum_{i=1}^{n^{shp}} \left\| d_i^{shp} \right\|^2 \tag{6.12}$$

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

$$\alpha_i = angle(\mathbf{c}_{i-1}, \mathbf{c}_i, \mathbf{c}_{i+1}). \tag{6.13}$$

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 $\mathbf{x}_i^{cor}$ of the eyelid interface end points $c_i^{cor} \in 0, 1$ and their corresponding corner annotations $\mathbf{a}_i^{cor}$

$$\mathbf{x}_i^{cor} = Camera(\, Eyeball(\, Eyelid(c_i^{cor}, \dot{\bar{\mathcal{P}}}), \dot{\bar{\mathcal{P}}})\,)$$
$$E_{corners} = w_{corners} \cdot \frac{1}{n^{cor}} \sum_{i=1}^{n^{cor}} \| \mathbf{x}_i^{cor} - \mathbf{a}_i^{cor} \|^2. \tag{6.14}$$

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 $\mathbf{a}_i^{pup}$ with the cornea, providing the intersection point $\mathbf{y}_i^{pup}$ in texture space. We then refract the ray at this point and compute the distance between the refracted ray $\mathbf{r}_i^{pup}$ and the model pupil circle $Pupil(\dot{\bar{\mathcal{P}}})$

$$\mathbf{r}_i^{pup} = Refract(\, Camera^{-1}(\mathbf{a}_i^{pup}),\, Eyeball(\mathbf{y}_i^{pup},\, \dot{\tilde{\mathcal{P}}})\,)$$
$$E_{pupil}^{a} = w_{pupil}^{a} \cdot \frac{1}{n^{pup}} \cdot \sum_{i=1}^{n^{pup}} \left\| \mathbf{r}_i^{pup},\, Pupil(\dot{\tilde{\mathcal{P}}}) \right\|_{ray-circle}^{2}. \tag{6.15}$$

However, since the shape of the cornea changes during the optimization, we cannot keep $\mathbf{y}_i^{pup}$ 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 $\mathbf{a}_i^{pup}$

$$\mathbf{x}_i^{pup} = Camera(\, Eyeball(\mathbf{y}_i^{pup},\, \dot{\tilde{\mathcal{P}}})\,)$$
$$E_{pupil}^{b} = w_{pupil}^{b} \cdot \frac{1}{n^{pup}} \cdot \sum_{i=1}^{n^{pup}} \left\| \mathbf{x}_i^{pup} - \mathbf{a}_i^{pup} \right\|^{2}. \tag{6.16}$$

The final pupil energy is given by the sum of Equation 6.15 and Equation 6.16.

6.3.2 Photometric Refinement

The annotation-based fitting presented in section 6.3.1 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

$$\begin{aligned} E_{refinement} = E_{annotation} \\ &+ w_{inter-camera} \cdot E_{inter-camera} \\ &+ w_{inter-frame} \cdot E_{inter-frame} \\ &+ w_{reference-frame} \cdot E_{reference-frame}. \end{aligned} \tag{6.17}$$

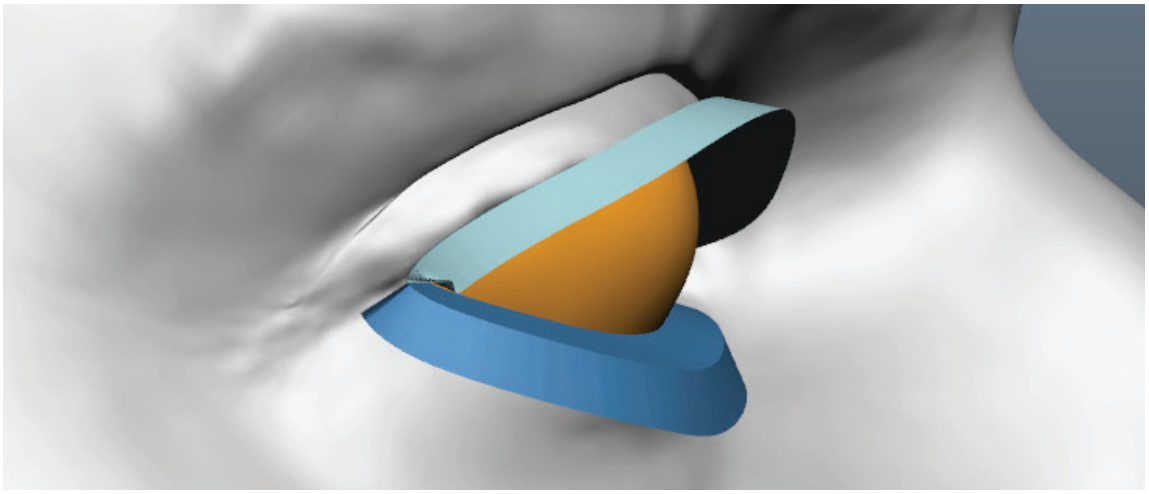

Figure 6.6: *The proxy eyelid geometry used to compute the sclera masks. The figure shows the eyeball (orange) and skin (gray) geometries. The lower and upper eyelash proxies (blue) consist of an eyelid margin (perpendicular to the eyeball surface) and an eyelash part.*

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. 6.6. 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.

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. [2010] 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 $\mathbf{x}_i^{opt}$ and the corresponding closest point on the eyeball surface. The closest point is defined by a texture coordinate $\mathbf{y}_i^{inter-camera}$ and is part of the optimization parameters.

$$\begin{aligned} \mathbf{x}_i^{inter-camera} &= Eyeball(\mathbf{y}_i^{inter-camera}, \dot{\mathcal{P}}) \\ E_{inter-camera} &= \left\| \mathbf{x}_i^{inter-camera} - \mathbf{x}_i^{opt} \right\|^2 . \end{aligned} \tag{6.18}$$

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 [Fischler and Bolles, 1981] 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 $\mathbf{f}_i^j$ and $\mathbf{f}_k^j$ into the camera yielding $\mathbf{a}_i^j$ and $\mathbf{a}_k^j$ for frames i and k, respectively:

$$\mathbf{a}_i^j = Camera(\ Eyeball(\mathbf{f}_i^j, \dot{\bar{\mathcal{P}}}_i)\ ). \tag{6.19}$$

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

$$\begin{aligned} \mathbf{x}_i^j &= Camera(\ Eyeball(\mathbf{f}^j, \dot{\bar{\mathcal{P}}}_i)\ ) \\ E_{inter-frame} &= \sum_{i,j} \left\| \mathbf{x}_i^j - \mathbf{a}_i^j \right\|^2. \end{aligned} \tag{6.20}$$

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. 6.7) 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 $\dot{\Theta}_z$ to align the two poses.

With this final refinement step we can accurately compute the poses of the eyes in all frames individually. In the next section we describe how eye rigs may be fitted to this data and in Section 6.5 we discuss the captured data in detail and elaborate how it has informed the design of the proposed eye rig.

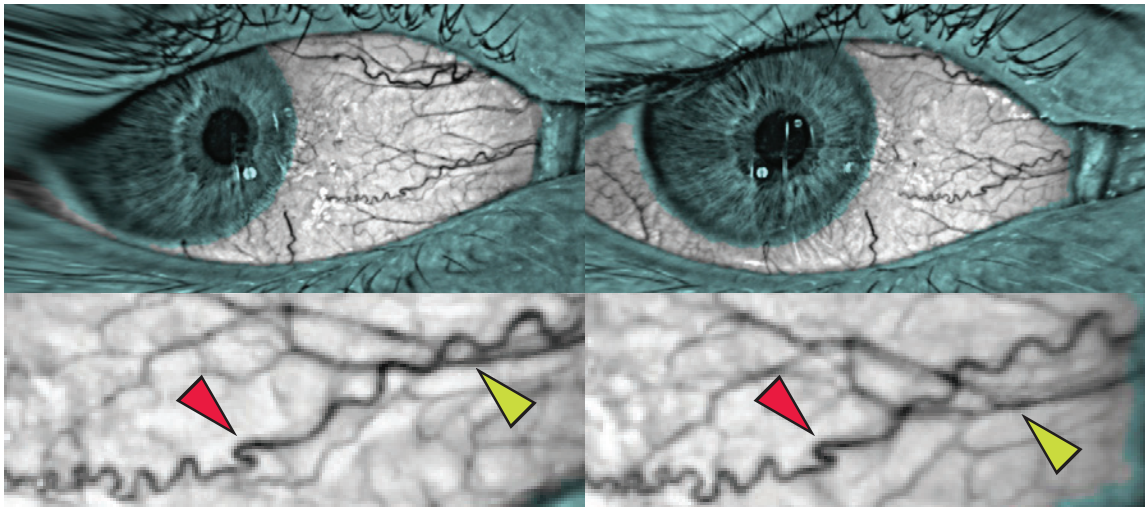

Figure 6.7: *The eyeball (shown here in texture space, with the sclera masked) is coated by a protective, mostly transparent tissue layer called the conjunctiva, which is not firmly attached to the eyeball but slides over it during rotation. As a consequence, the veins in the conjunctiva (green arrow) deform relative to the sclera (red arrows). This complicates alignment of eye poses considerably.*

6.4 Eye Rig Fitting

In the previous section we introduced residual variables that add additional degrees of freedom to the rig and allows 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.

6.4.1 Listing's Model

Listing's model predicts the per frame torsion $\dot{\Theta}_z$ based on the eye gaze $(\dot{\Theta}_x, \dot{\Theta}_y)$. 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. 6.8 the model predicts the torsion well in the central field of view but degrades with more extreme gazes.

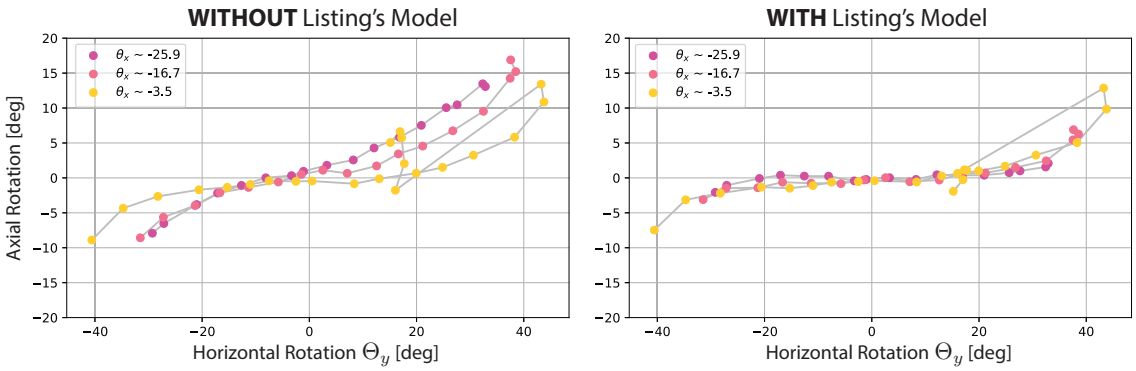

Figure 6.8: *Not predicting rotation around the optical axis amounts in large residuals across the entire range of motion. Listing's model predicts the torsion reliably for the largest part but fails to explain the extremes where it appears to deviate from the true physiology of the eye. Note that the model correctly predicts the dependency on elevation of the eye (Θ_x).*

6.4.2 Translation Model

The translational component of the eye center is estimated using the proposed mapping function $\mathbf{\Psi}(\cdot)$ that predicts each translational component from the eye gaze ($\dot{\Theta}_x$,$\dot{\Theta}_y$). The parameters are estimated from the measured per frame translation offsets using bivariate quadratic regression. As shown in Fig. 6.9 the model predicts the translational behaviour of the eye well.

6.4.3 Eye Rig

Once the static parameters of the predictive models have been computed, we optimize the other parameters of the rig based on the per frame eye configuration. We uniformly sample the front of 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.

Simple Rig Disabling the predictive models reduces the proposed eye-rig to the typical eye-rigs used in computer animation, where the eye motion is modelled by two rotations ($\dot{\Theta}_x$,$\dot{\Theta}_y$) around a fixed center of rotation $\bar{\mathbf{p}}$.

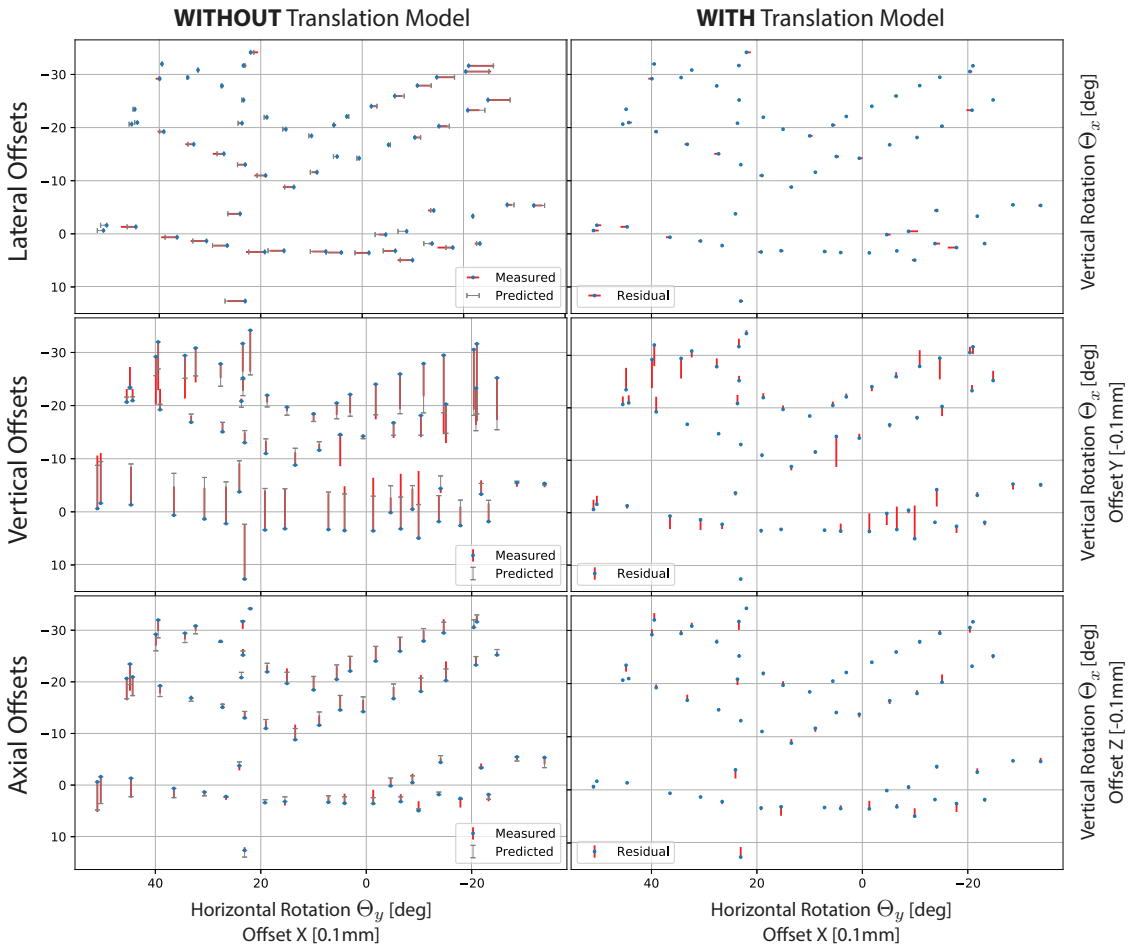

Figure 6.9: *The left column shows the remaining translation residuals (red bars) without and the right one with the proposed translation model for offsets along x- (top), y- (middle) and z-axis (bottom). The proposed model reduces the residual substantially. Note, for example, how the lateral offset show that the eye translates to the side of the gaze direction as the muscles pull it, while the vertical residuals without model indicate that the eye moves upwards when looking down and down when looking up. This is in line with findings from ophthalmology [Fry and Hill, 1962; Fry and Hill, 1963] and the proposed regression succeeds at modeling the effect.*

6.4.4 Visual Axis

In addition to the per frame eye configuration reconstructed in the previous section we also record the look-at point for every frame (Section 6.2). This allows to compute the visual axis per eye, given by the offset to its optical axis in spherical coordinates.

6.5 Results

We start by validating the accuracy of the proposed system as well as investigating the relevancy of our more accurate eye rig for computer vision and graphics applications.

The algorithm presented in Section 6.3 allows to reconstruct eye poses from multi-view imagery at submillimeter precision. Our results are in line with the findings presented in ophthalmology research papers measured using specialized hardware [Fry and Hill, 1962; Fry and Hill, 1963; Carpenter, 1988]. Fig. 6.10 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. 6.4) 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, which is due to the discrepancy between the optical and visual axis (Fig. 3.1 (b)).

The left column in Fig. 6.12 shows the measured slack variables, more specifically the translation offset of the pivot as well as the rotation of the eye around its optical axis, called torsion. One can identify clear patterns that show that for example the eye moves to the left when rotating left, since the muscles pull it in that direction. Interestingly, when rotating up the eye actually translates down and sideways towards the nose.

These measurements are in line with findings reported by ophthalmology researchers as shown in Fig. 6.11, but we can capture them with a general-purpose multi-view camera system that allows to go beyond the more constrained ophthalmological acquisition, which typically limits motion to a single direction.

Obviously the simple eye rigs typically employed in computer vision and graphics cannot explain these measurements. Hence we propose a novel eye

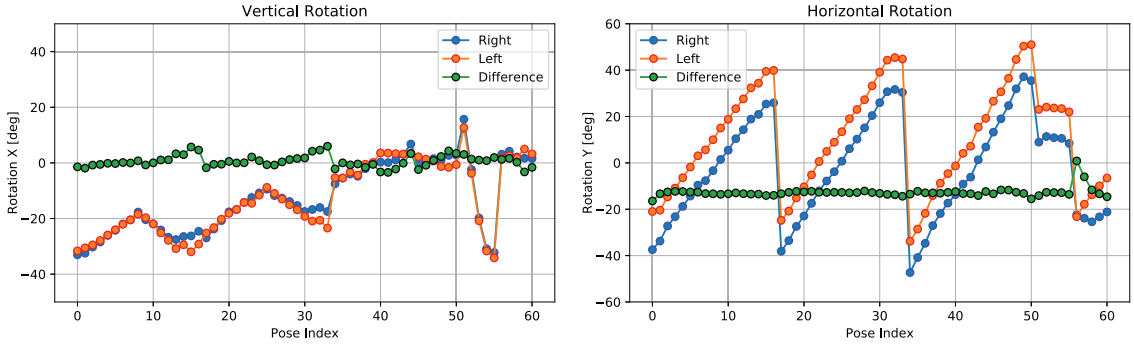

Figure 6.10: *This figure shows the raw measurements of gaze angles for one subject, for both the left (orange) and (blue) eyes. The subject did three sweeps left to right at different eye elevations (frames 0-15, 16-33, and 34-50) and finally a vertical sweep from neutral upwards (51-55). Since the look-at points were distributed on the capture gantry running over a corner, the vertical eye motion is higher on the sides than at the front which is clearly visible in the left plot.*

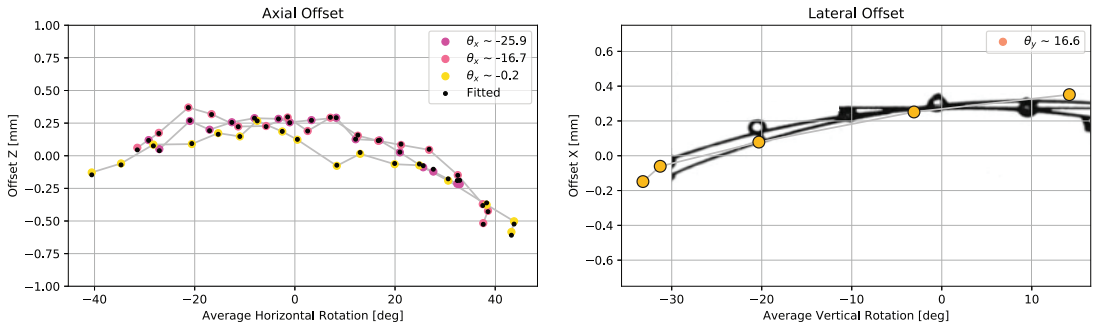

Figure 6.11: *Our measurements match the findings presented by Fry and Hill. The left plot shows the three horizontal sweeps from Fig. 6.10, plotted in relation to Θ_y. The colors indicate the three different elevations and the black dots shows the projection onto the centrode (the orbit of the eye pivot) as suggested by [Fry and Hill, 1962]. The estimated radius of the sphere is 4mm, which is within the range they reported. The right plot shows lateral motion for the vertical sweep plotted over the figure from [Fry and Hill, 1963] demonstrating that our measurements match theirs very well.*

rig that incorporates knowledge from ophthalmology, which fits the measurements much better as shown in Fig. 6.12. The translation model introduced in Section 6.1 succeed at removing most of the signal, leaving just fitting noise behind, which lies in the order of a tenth of a millimeter. The Listings model also predicts the torsion well, except for the extreme gaze poses, where the physiology of the eye motion appears to disagree the theoretical model.

The fact that these measurements have been computed from ordinary cameras is a strong indicator that phenomenons such as torsion and eye translation can be important for computer vision applications, such as accurate eye gaze estimation (Fig. 6.14). Eye gaze is also central for computer animation, where a common mistake is to presume the visual axis to align with the optical axis of the eye, which will lead to cross-eyed characters (Fig. 6.13). As the visual axis is rotated nasally by about 8 degrees on average relative to our eye model coordinate frame, the optical axis is actually pointing outwards when a person is looking at infinity.

To understand the relevancy of our findings for computer graphics applications we synthesize several eye poses with both the presented and a traditional eye rig (Fig. 6.14). The traditional eye rig neglects torsion and translation of the rotational pivot, which leads to subtle yet very noticeable effects especially for extreme eye poses resulting in a difference in perceived gaze for the two rigs.

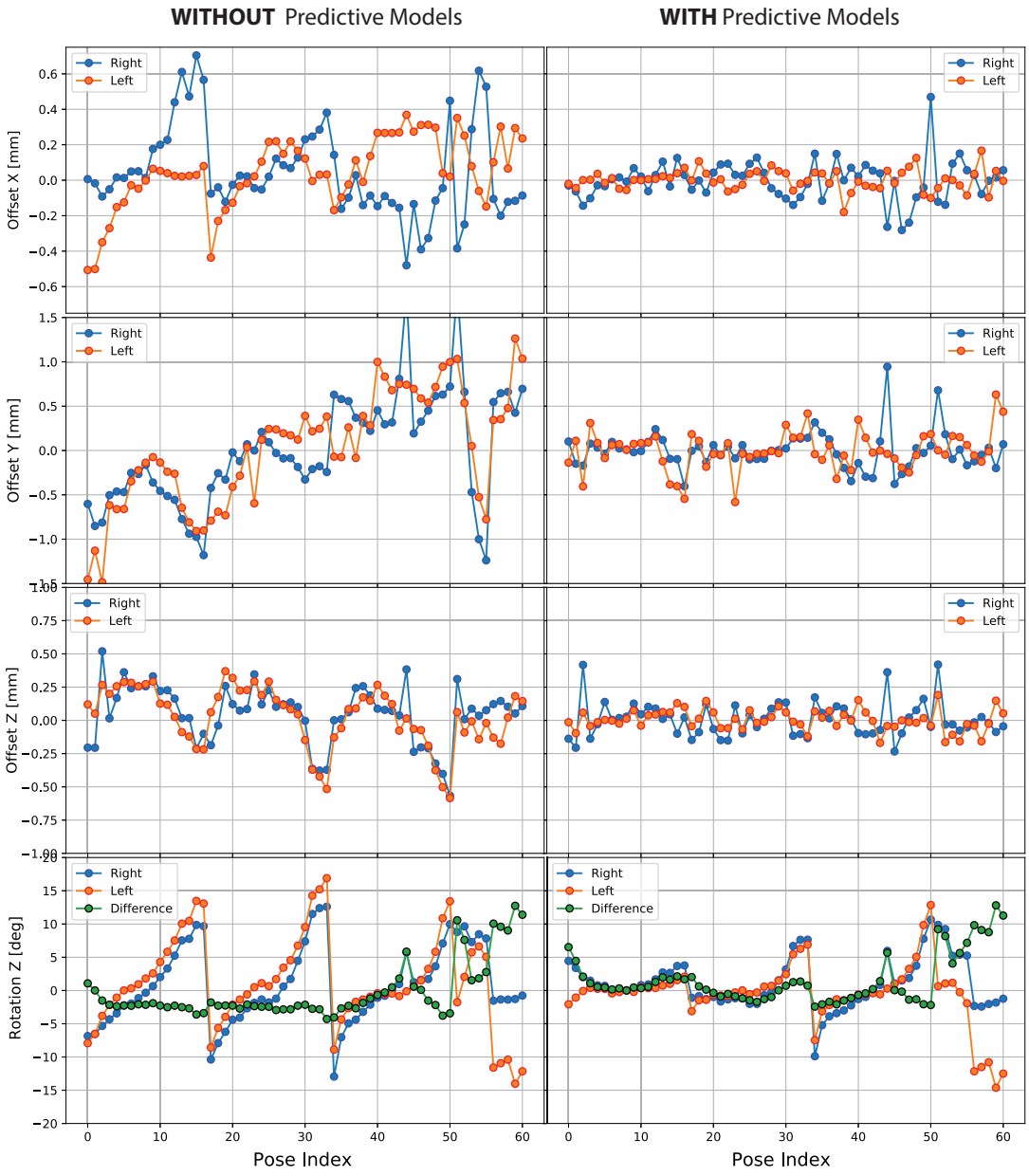

Figure 6.12: *With the proposed method we can measure both translation of the pivot (top three rows) and rotation of the eye around its axis (bottom row). The measurements reveal patterns which are strongly correlated with the eye movement (Fig. 6.12). For example it is apparent that the eye moves backwards when rotating to either side (bottom-right), sideways in the direction of gaze (bottom-left), and downwards when looking up (bottom-center). These results are in line with findings from the ophthalmology community (Fig. 6.11). We introduce predictive models in Section 6.1 that succeed at explaining most of the patterns, hence allowing the presented rig to better resemble the physiologically correct eye motion.*

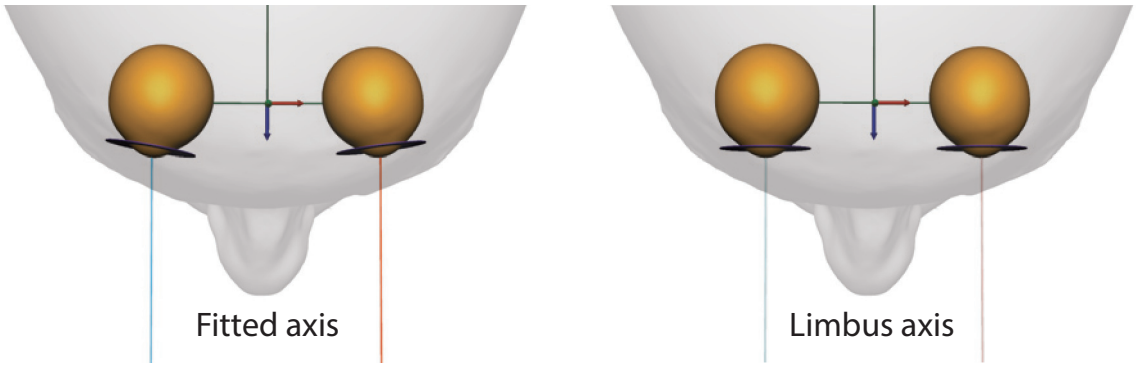

Figure 6.13: *The visual axis is tilted towards the nose and is not perpendicular to the limbus. This results in the limbus planes (violet) being oriented away from the nose if the subject's gaze is at infinity.*

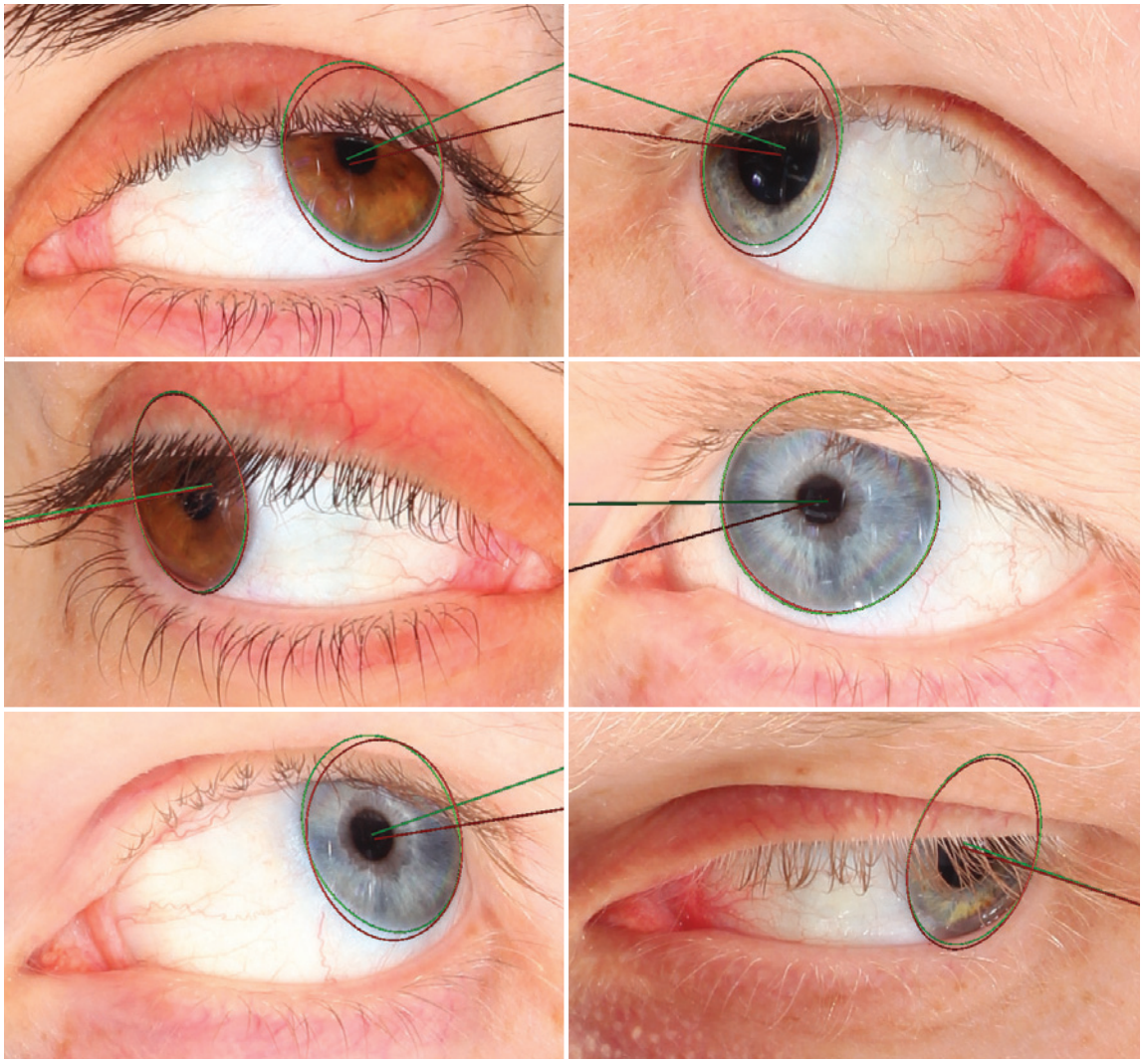

Figure 6.14: *Neglecting torsion and translation of the rotational pivot leads to subtle yet very noticeable effects as can be seen in this figure. The green limbus contour plus orientation of the limbus plane represent the results of the proposed method, where the red ones stem from a simple rig without these components, but with optimized center of rotation. As you can see, the simple model degrades in particular towards the extreme eye poses, resulting in a difference in perceived gaze.*

Figure 6.15: *A lot of the mismatches in Fig. 6.14 stem from the fact that traditional eye models neglect the rotation around the z-axis called torsion. This results in a mismatch of up to 15 degrees, which is clearly visible on the left side where the red and blue channels do not align. Using the torsion predicted by the Listing's model alleviates this as can be seen on the right.*

CHAPTER 7

Conclusion

In this thesis we present methods for the creation of eyes for digital humans. This includes algorithms for the reconstruction, the modeling, and the rigging of eyes for animation and tracking applications.

We capture the shape and texture of the most prominent components of the eye at an unprecedented level of detail, including the *sclera*, the *cornea*, and the *iris*. We demonstrate that the generic eye models typically used in our community are not sufficient to represent all the intricacies of an eye, which are very person-specific, and we believe that the findings of this thesis have the potential to alter our community's current assumptions regarding human eyes. In addition, we present the first method for reconstructing detailed iris deformation during pupil dilation, and demonstrate two applications of how data-driven iris animations can be combined with our high-quality eye reconstructions. A data set has been published on our website[1].

These eye reconstructions enable the creation of a new parametric model of 3D eyes built from a database of high-resolution scans with both geometry and texture. Our model contains a shape subspace for the eyeball, a coupled shape and color synthesis method for the iris parameterized by a low-resolution control map, and a sclera vein synthesis approach also with tunable parameters to generate a variety of realistic vein networks. We also present an image-based fitting algorithm that allows our parametric model to be fit to lightweight inputs, such as common facial scanners, or even single images that can be found on the internet. Our parametric model and fitting approach allow for simple and efficient eye reconstructions, making eye cap-

[1] https://www.disneyresearch.com/publication/high-quality-capture-of-eyes/

ture a more viable approach for industry and home use. Furthermore, the model allows to manipulate the captured data as it is fully parametric, such as changing the amount and appearance of sclera veins to simulate physiological effects or controlling the pupil size to have the eye react to synthetic illumination.

Based on the parametric eye model we present a novel eye rig informed by ophthalmology findings and based on accurate measurements from a multi-view imaging system that can reconstruct eye poses at submillimeter accuracy. Our goal is to raise the awareness in the computer graphics and vision communities that the eye movement is more complex than oftentimes assumed. More specifically, we show that the eye is not a purely rotational device but actually translates during rotation and that it also rotates around its optical axis. These are important facts, for example, for foveal rendering and head-mounted display devices, which are gaining a lot of popularity due to the emerging augmented and virtual reality entertainment. Another important aspect that animators have to consider is the fact that the visual axis of the eye, which defines its gaze, is not identical with the optical axis. Neglecting this leads to cross-eyed gazes which quickly lead into the uncanny valley. We investigate the effect of ignoring or modeling these phenomena in the context of computer graphics and computer vision to provide the reader an intuition of their potential importance for their application.

We believe that these tools and methods are a valuable contribution to the current eye creation pipeline. These methods allow for the more accurate and realistic creation of eye shape, appearance, and motion on one hand, but they also allow for an easier, faster, and more robust creation on the other hand. We believe that these tools have the potential to change how eyes are being modeled in the visual effects industry and we hope that they will infuse a soul into a new generation of digital doubles in films.

7.1 Limitations

The methods presented in this thesis are great for various visual effects applications. But unfortunately, they have their limitations.

Our eye reconstruction introduced in Chapter 4 system approximates the iris as a surface. Since the iris is a volumetric object with partially translucent tissue it is difficult to reconstruct accurately. We believe, however, that the optical flow correspondences used to reconstruct that surface are sufficiently accurate to represent the iris with adequate details suitable for rendering,

already a step forward from traditional practices that approximate the iris as a plane or cone.

Our parametric eye model (Chapter 5) allows for single-image reconstructions, but unfortunately not without limitations. As can be seen in some of the single-image reconstructions, reflections off the cornea are difficult to identify and ignore and thus can become baked-in to the iris texture (see the bird example in Fig. 5.15). Additionally, our sclera vein synthesis does not guarantee to produce vein networks that match any partial veins that might be visible in the input images. Also, our model is naturally limited by the size and variation of the input database, and since only a limited number of scanned high-quality real eyes are currently available, our results may not optimally match the inputs, but this will be alleviated as more database eyes become available.

At this stage our fitted rig is person-specific and we have not investigated how it can generalize to others. This would, of course, be highly desirable since building the rig requires a dedicated capture session and a lot of manual annotations, which is realistically only feasible for hero assets in large productions. It will, however, enable to acquire a large corpus of eye motion and to create a generalizable model based on that data, which we consider interesting future work.

Nevertheless, even with these limitations our methods provide a great starting point for computer graphics artists to create realistic eyes from images.

7.2 Outlook

Besides addressing the limitations of our system, there are still many opportunities to extend and improve the presented techniques including reconstruction of the eye region, modeling of the eye appearance, and modeling of the eye motion.

Eye region Reconstructing, modeling, and rigging the geometry of the eye region is still a remaining challenge. The complex geometries, the various materials, the interfaces between the different parts of the eye, and the occlusions make this a very challenging undertaking. Furthermore, the eyes and the surrounding skin substantially influence each other. The eyelid is deformed as the eye moves underneath it and we expect that opening the eyes wide or firmly closing them does actually influence the position of the eye in the socket, hence affecting its motion. Future work should thus look

at ways to couple these two models and provide a rig for the entire eye region. Currently, a digital artist creates an eye region rig by hand based on his experience. The model-guided eyelid reconstruction of Bermano et al. [2015] is a first step, but it is limited to the reconstruction of eyelids. The eyelids cannot be animated or modified since there is no underlying rig that allows for this interpolation. Wood et al. [2016a] present a model based on the principal component analysis. We believe that better models can be found, that are better suited to represent the nonlinear deformations of the eye region.

Eye appearance Capturing and modeling the appearance of the eye as well as eye rendering are topics that we do not investigate in this thesis. We do not capture reflectance properties of the eyes, such as BRDF or BSSRDF parameters. Also, the eye consists of various materials, some varying in space, such as the sclera and the cornea. Anatomically they are the same part, but the appearance transitions from white to transparent due to a structural change. Also the tear layer, covering the eye, leads to glints, many of them close to the eyelids. The amount of liquid of this layer changes depending on the physical and emotional state of the person. New appearance capture methods need to be developed in the future which model the eye with an even greater attention to detail to produce more realistic digital humans.

Eye dynamics In this thesis, we model the motion of the eye in a static sense. We model the different poses and states the eyeball and the iris can reach over time, but we do not analyze the temporal characteristics and dynamic behavior such as saccades, hippus, or tremor. These motions have been investigated extensively, but the analysis is often limited to a 2D gaze analysis. The 3D techniques, presented in this thesis, might reveal previously unknown effects. This might lead to potential applications outside of our community, for example in ophthalmology, where accurate image-based eye acquisition could help in discovering, monitoring and treating eye diseases.

APPENDIX A

Appendix

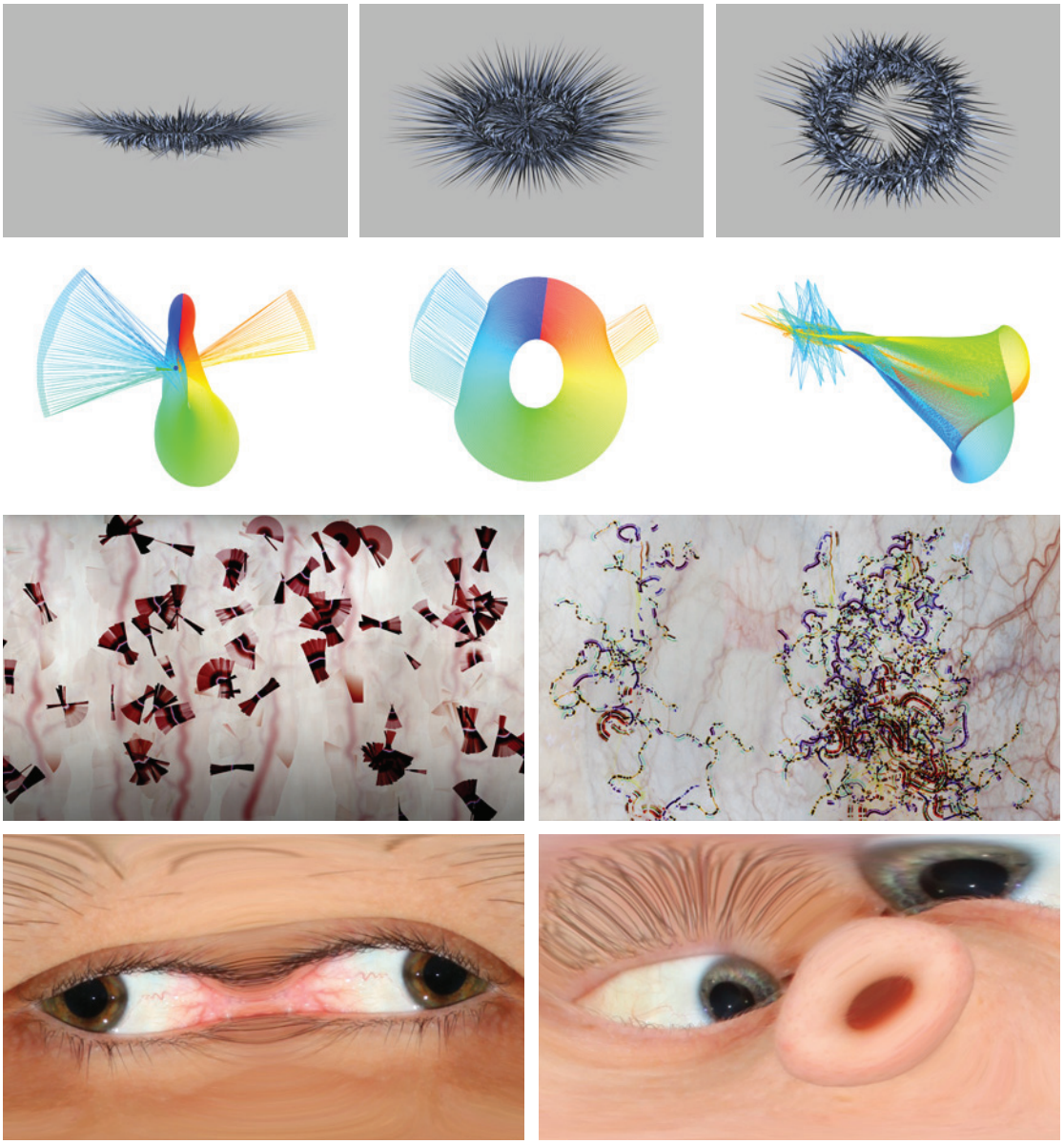

Figure A.1: *This figure shows what can go wrong. The two top rows show artifacts produced by wrong indexing and constraint definition in the Laplacian system solve of the iris. In the third row we show the result of a range overflow in vein rendering, and in the bottom row we show the resulting eyeball textures based on badly fitted eyes.*

References

[Agarwal et al., 2018] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`, 2018.

[Allen et al., 2003] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 587–594. ACM, 2003.

[Amberg et al., 2007] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[Anguelov et al., 2005] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 408–416. ACM, 2005.

[Ares and Royo, 2006] M Ares and S Royo. Comparison of cubic b-spline and zernike-fitting techniques in complex wavefront reconstruction. *Applied optics*, 45(27):6954–6964, 2006.

[Atcheson et al., 2010] Bradley Atcheson, Felix Heide, and Wolfgang Heidrich. CALTag: High precision fiducial markers for camera calibration. In *International Workshop on Vision, Modeling and Visualization*, 2010.

[Atchison et al., 2004] David A Atchison, Catherine E Jones, Katrina L Schmid, Nicola Pritchard, James M Pope, Wendy E Strugnell, and Robyn A Riley. Eye shape in emmetropia and myopia. *Investigative Ophthalmology & Visual Science*, 45(10):3380–3386, 2004.

[Beeler and Bradley, 2014] Thabo Beeler and Derek Bradley. Rigid stabilization of facial expressions. *ACM Transactions on Graphics (TOG)*, 33(4):44, 2014.

[Beeler et al., 2010] T. Beeler, B. Bickel, R. Sumner, P. Beardsley, and M. Gross. High-quality single-shot capture of facial geometry. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 29(4):40, 2010.

[Beeler et al., 2011] Thabo Beeler, Fabian Hahn, Derek Bradley, Bernd Bickel, Paul Beardsley, Craig Gotsman, Robert W Sumner, and Markus Gross. High-quality passive facial performance capture using anchor frames. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 30(4):75, 2011.

[Benel et al., 1991] D. C. R. Benel, D. Ottens, and R. Horst. Use of an eye tracking system in the usability laboratory. In *Proc. of the Human Factors Society 35th Annual Meeting*, pages 461–465, 1991.

[Bermano et al., 2015] Amit Bermano, Thabo Beeler, Yeara Kozlov, Derek Bradley, Bernd Bickel, and Markus Gross. Detailed spatio-temporal reconstruction of eyelids. *ACM Transactions on Graphics (TOG)*, 34(4):44, 2015.

[Besl and McKay, 1992] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Trans. on PAMI*, 14(2):239–256, 1992.

[Blanz and Vetter, 1999] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proc. of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.

[Boykov and Kolmogorov, 2004] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004.

[Bradley et al., 2010] Derek Bradley, Wolfgang Heidrich, Tiberiu Popa, and Alla Sheffer. High resolution passive facial performance capture. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 29(4):41, 2010.

[Brown and Rusinkiewicz, 2004] Benedict J Brown and Szymon Rusinkiewicz. Non-rigid range-scan alignment using thin-plate splines. In *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, pages 759–765. IEEE, 2004.

[Brox et al., 2004] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, pages 25–36. Springer, 2004.

[Cao et al., 2014] Chen Cao, Qiming Hou, and Kun Zhou. Displaced dynamic

expression regression for real-time facial tracking and animation. *ACM Transactions on Graphics (TOG)*, 33(4):43, 2014.

[Carpenter, 1988] Roger HS Carpenter. *Movements of the Eyes, 2nd Rev*. Pion Limited, 1988.

[Chen et al., 2013] Kan Chen, Henry Johan, and Wolfgang Mueller-Wittig. Simple and efficient example-based texture synthesis using tiling and deformation. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 145–152. ACM, 2013.

[Chopra-Khullar and Badler, 2001] Sonu Chopra-Khullar and Norman I. Badler. Where to look? automating attending behaviors of virtual human characters. *Autonomous Agents and Multi-Agent Systems*, 4(1):9–23, 2001.

[Collewijn, 1999] H. Collewijn. Eye movement recording. *Vision Research: A Practical Guide to Laboratory Methods*, pages 245–285, 1999.

[Deng et al., 2005] Zhigang Deng, J. P. Lewis, and U. Neumann. Automated eye motion using texture synthesis. *IEEE CG&A*, 25(2):24–30, 2005.

[Dodge and Cline, 1901] R. Dodge and T. S. Cline. The angle velocity of eye movements. *Psychological Review*, 8:145–157, 1901.

[Eagle Jr, 1988] RC Eagle Jr. Iris pigmentation and pigmented lesions: an ultrastructural study. *Trans. of the American Ophthalmological Society*, 86:581, 1988.

[Efros and Freeman, 2001] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001.

[Efros and Leung, 1999] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *IEEE ICCV*, pages 1033–1038, 1999.

[Eggert, 2007] Thomas Eggert. Eye movement recordings: Methods. *Neuro-Ophthalmology*, 40:15–34, 2007.

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

[Flickr, 2006] Flickr. *Mohammed Alnaser - Mr. Falcon*. Creative Commons - Attribution 2.0, 2006. `https://www.flickr.com/photos/69er/324313066`.

[François et al., 2009] Guillaume François, Pascal Gautron, Gaspard Breton, and Kadi Bouatouch. Image-based modeling of the human eye. *IEEE TVCG*, 15(5):815–827, 2009.

[Fry and Hill, 1962] GA Fry and WW Hill. The center of rotation of the eye*. *Optometry & Vision Science*, 39(11):581–595, 1962.

[Fry and Hill, 1963] Glenn A Fry and WW Hill. The mechanics of elevating the eye*. *Optometry & Vision Science*, 40(12):707–716, 1963.

[Funkhouser et al., 2004] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. Modeling by example. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 652–663. ACM, 2004.

[Fyffe et al., 2011] Graham Fyffe, Tim Hawkins, Chris Watts, Wan-Chun Ma, and Paul Debevec. Comprehensive facial performance capture. In *Eurographics*, 2011.

[Fyffe et al., 2014] Graham Fyffe, Andrew Jones, Oleg Alexander, Ryosuke Ichikari, and Paul Debevec. Driving high-resolution facial scans with video performance capture. *ACM Trans. Graphics*, 34(1):8:1–8:14, 2014.

[Garrido et al., 2013] Pablo Garrido, Levi Valgaerts, Chenglei Wu, and Christian Theobalt. Reconstructing detailed dynamic face geometry from monocular video. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 32(6):158, 2013.

[Ghosh et al., 2011] Abhijeet Ghosh, Graham Fyffe, Borom Tunwattanapong, Jay Busch, Xueming Yu, and Paul Debevec. Multiview face capture using polarized spherical gradient illumination. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 30(6):129, 2011.

[Hachol et al., 2007] Andrzej Hachol, W Szczepanowska-Nowak, H Kasprzak, I Zawojska, A Dudzinski, R Kinasz, and D Wygledowska-Promienska. Measurement of pupil reactivity using fast pupillometry. *Physiological measurement*, 28(1):61, 2007.

[Haehnel et al., 2003] Dirk Haehnel, Sebastian Thrun, and Wolfram Burgard. An extension of the icp algorithm for modeling nonrigid objects with mobile robots. In *IJCAI*, volume 3, pages 915–920, 2003.

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

[Haralick, 1979] Robert M. Haralick. Statistical and structural approaches to texture. *Proc. IEEE*, 67(5):786–804, 1979.

[Hartridge and Thompson, 1948] H. Hartridge and L. C. Thompson. Methods of investigating eye movements. *British Journal of Ophthalmology*, 32:581–591, 1948.

[Hernández et al., 2008] Carlos Hernández, George Vogiatzis, and Roberto Cipolla. Multiview photometric stereo. *IEEE PAMI*, 30(3):548–554, 2008.

[Hogan et al., 1971] Michael J. Hogan, Jorge A. Alvarado, and Joan E. Weddell. Histology of the human eye: an atlas and textbook. *Philadelphia: Saunders*, pages 393–522, 1971.

[Huang et al., 1991] David Huang, Eric A Swanson, Charles P Lin, Joel S Schuman, William G Stinson, Warren Chang, Michael R Hee, Thomas Flotte, Kenton Gregory, Carmen A Puliafito, et al. Optical coherence tomography. *Science*, 254(5035):1178–1181, 1991.

[Ihrke et al., 2008] Ivo Ihrke, Kiriakos N Kutulakos, Hendrik PA Lensch, Marcus Magnor, and Wolfgang Heidrich. State of the art in transparent and specular object reconstruction. In *Eurographics 2008 - State of the Art Reports*, 2008.

[Ikemoto et al., 2003] Leslie Ikemoto, Natasha Gelfand, and Marc Levoy. A hierarchical method for aligning warped meshes. In *3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on*, pages 434–441. IEEE, 2003.

[Itti et al., 2003] Laurent Itti, Nitin Dhavale, and Frederic Pighin. 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.

[Jacob and Unser, 2004] Mathews Jacob and Michael Unser. Design of steerable filters for feature detection using canny-like criteria. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(8):1007–1019, 2004.

[Janik et al., 1978] S. W. Janik, A. R. Wellens, M. L. Goldberg, and L. F. Dell'Osso. Eyes as the center of focus in the visual examination of human faces. *Perceptual and Motor Skills*, 47(3):857–858, 1978.

[Judd et al., 1905] C. H. Judd, C. N. McAllister, and W. M. Steel. General introduction to a series of studies of eye movements by means of kinetoscopic photographs. *Psychological Review, Monograph Supplements*, 7:1–16, 1905.

[Kazhdan et al., 2004] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Shape matching and anisotropy. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 623–629. ACM, 2004.

[Kwatra et al., 2003] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: image and video synthesis using graph cuts. In *ACM Transactions on Graphics (ToG)*, volume 22, pages 277–286. ACM, 2003.

[Lam and Baranoski, 2006] M. W. Y. Lam and G. V. G. Baranoski. A predictive light transport model for the human iris. In *Computer Graphics Forum*, volume 25, pages 359–368, 2006.

[Le et al., 2012] B. H. Le, X. Ma, and Z. Deng. Live speech driven head-and-eye motion generators. *IEEE TVCG*, 18(11), 2012.

[Lee et al., 2002] Sooha Park Lee, Jeremy B. Badler, and Norman I. Badler. Eyes alive. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 21(3):637–644, 2002.

[Lefohn et al., 2003] Aaron Lefohn, Brian Budge, Peter Shirley, Richard Caruso, and Erik Reinhard. An ocularist's approach to human iris synthesis. *IEEE CG&A*, 23(6):70–75, 2003.

[LeGrand and ElHage, 2013] Yves LeGrand and Sami G ElHage. *Physiological optics*, volume 13. Springer, 2013.

[Levoy and Whitaker, 1990] M. Levoy and R. Whitaker. Gaze-directed volume rendering. In *Symposium on Interactive 3D Graphics*, pages 217–223, 1990.

[Li et al., 2008] Hao Li, Robert W Sumner, and Mark Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Computer graphics forum*, volume 27, pages 1421–1430. Wiley Online Library, 2008.

[Li et al., 2015] Jun Li, Weiwei Xu, Zhiquan Cheng, Kai Xu, and Reinhard Klein. Lightweight wrinkle synthesis for 3d facial modeling and animation. *Computer-Aided Design*, 58:117–122, 2015.

[Liang et al., 2001] Lin Liang, Ce Liu, Ying-Qing Xu, Baining Guo, and Heung-Yeung Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics (ToG)*, 20(3):127–150, 2001.

[Ma and Deng, 2009] Xiaohan Ma and Zhigang Deng. Natural eye motion synthesis by modeling gaze-head coupling. In *Proc. IEEE VR*, pages 143–150, 2009.

[Ma et al., 2007] Wan-Chun Ma, Tim Hawkins, Pieter Peers, Charles-Felix Chabert, Malte Weiss, and Paul Debevec. Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. In *Proc. Rendering Techniques*, pages 183–194, 2007.

[Mackey et al., 2011] David A Mackey, Colleen H Wilkinson, Lisa S Kearns, and Alex W Hewitt. Classification of iris colour: review and refinement of a classification schema. *Clinical & experimental ophthalmology*, 39(5):462–471, 2011.

[Mackworth and Thomas, 1962] N. H. Mackworth and E. L. Thomas. Head-mounted eye-marker camera. *Journal of the Optical Society of America*, 52:713–716, 1962.

[Marsella et al., 2013] Stacy Marsella, Yuyu Xu, Margaux Lhommet, Andrew Feng, Stefan Scherer, and Ari Shapiro. Virtual character performance from speech. In *Proc. SCA*, pages 25–35, 2013.

[Mohammed et al., 2009] Umar Mohammed, Simon JD Prince, and Jan Kautz. Visio-lization: generating novel facial images. *ACM Transactions on Graphics (TOG)*, 28(3):57, 2009.

[Nishino and Nayar, 2004] Ko Nishino and Shree K Nayar. Eyes for relighting. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 704–711. ACM, 2004.

[Palubicki et al., 2009] Wojciech Palubicki, Kipp Horel, Steven Longay, Adam Runions, Brendan Lane, Radomír Měch, and Przemyslaw Prusinkiewicz. Self-organizing tree models for image synthesis. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 28(3):58, 2009.

[Pamplona et al., 2009] Vitor F Pamplona, Manuel M Oliveira, and Gladimir VG Baranoski. Photorealistic models for pupil light reflex and iridal pattern deformation. *ACM Trans. Graphics (TOG)*, 28(4):106, 2009.

[Pejsa et al., 2016] Tomislav Pejsa, Daniel Rakita, Bilge Mutlu, and Michael Gleicher. Authoring directed gaze for full-body motion capture. *ACM Trans. Graphics (Proc. SIGGRAPH Asia*, 35(6), 2016.

[Pérez et al., 2003] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 22(3):313–318, 2003.

[Piñero, 2013] David P Piñero. Technologies for anatomical and geometric characterization of the corneal structure and anterior segment: a review. In *Seminars in Ophthalmology*, pages 1–10, 2013.

[Pinskiy and Miller, 2009] Dmitriy Pinskiy and Erick Miller. Realistic eye motion using procedural geometric methods. In *SIGGRAPH 2009: Talks*, page 75. ACM, 2009.

[Praun et al., 2000] Emil Praun, Adam Finkelstein, and Hugues Hoppe. Lapped textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 465–470. ACM Press/Addison-Wesley Publishing Co., 2000.

[Ramanarayanan and Bala, 2007] Ganesh Ramanarayanan and Kavita Bala. Constrained texture synthesis via energy minimization. *IEEE TVCG*, 13(1), 2007.

[Rio-Cristobal and Martin, 2014] Ana Rio-Cristobal and Raul Martin. Corneal assessment technologies: Current status. *Survey of Ophthalmology*, 2014.

[Rozenberg and Salomaa, 1976] Grzegorz Rozenberg and Arto Salomaa. *The mathematical theory of L systems*. Springer, 1976.

[Ruhland et al., 2014] 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 Reports*, pages 69–91, 2014.

[Sagar et al., 1994] Mark A. Sagar, David Bullivant, Gordon D. Mallinson, and Peter J. Hunter. A virtual environment and model of the eye for surgical simulation. In *Proceedings of Computer Graphics and Interactive Techniques*, pages 205–212, 1994.

[Seitz et al., 2006] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE CVPR*, volume 1, pages 519–528, 2006.

[Shen and Cai, 2009] Hui-Liang Shen and Qing-Yuan Cai. Simple and efficient method for specularity removal in an image. *Applied optics*, 48(14):2711–2719, 2009.

[Smolek and Klyce, 2003] Michael K Smolek and Stephen D Klyce. Zernike polynomial fitting fails to represent all visually significant corneal aberrations. *Investigative ophthalmology & visual science*, 44(11):4676–4681, 2003.

[Sorkine et al., 2004] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184. ACM, 2004.

[Sugano et al., 2014] Y. Sugano, Y. Matsushita, and Y. Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *IEEE CVPR*, 2014.

[Suwajanakorn et al., 2014] Supasorn Suwajanakorn, Ira Kemelmacher-Shlizerman, and Steven M Seitz. Total moving face reconstruction. In *Computer Vision–ECCV 2014*, pages 796–812. Springer, 2014.

[Tai et al., 2010] Yu-Wing Tai, Shuaicheng Liu, Michael S. Brown, and Stephen Lin. Super resolution using edge prior and single image detail synthesis. In *CVPR*, 2010.

[Valgaerts et al., 2012] Levi Valgaerts, Chenglei Wu, Andrés Bruhn, Hans-Peter Seidel, and Christian Theobalt. Lightweight binocular facial performance capture under uncontrolled lighting. *ACM Trans. Graph.*, 31(6):187, 2012.

[Van Run and Van den Berg, 1993] LJ Van Run and AV Van den Berg. Binocular eye orientation during fixations: Listing's law extended to include eye vergence. *Vision research*, 33(5):691–708, 1993.

[Vertegaal et al., 2001] Roel Vertegaal, Robert Slagter, Gerrit van der Veer, and Anton Nijholt. Eye gaze patterns in conversations: There is more to conversa-

tional agents than meets the eyes. In *Proc. Human Factors in Computing Systems*, pages 301–308, 2001.

[Vivino et al., 1993] M.A. Vivino, S. Chintalagiri, B. Trus, and M. Datiles. Development of a scheimpflug slit lamp camera system for quantitative densitometric analysis. *Eye*, 7(6):791–798, 1993.

[Vlasic et al., 2005] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Face transfer with multilinear models. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 426–433. ACM, 2005.

[Wang et al., 2015] Jianzhong Wang, Guangyue Zhang, and Jiadong Shi. Pupil and glint detection using wearable camera sensor and near-infrared led array. *Sensors*, 15(12):30126–30141, 2015.

[Wang et al., 2016] Congyi Wang, Fuhao Shi, Shihong Xia, and Jinxiang Chai. Realtime 3d eye gaze animation using a single rgb camera. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 35(4), 2016.

[Wei et al., 2009] Li-Yi Wei, Sylvain Lefebvre, Vivek Kwatra, and Greg Turk. State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR*, pages 93–117. Eurographics Association, 2009.

[Weissenfeld et al., 2010] Axel Weissenfeld, Kang Liu, and Jörn Ostermann. Video-realistic image-based eye animation via statistically driven state machines. *Vis. Comput.*, 26(9):1201–1216, 2010.

[Wen et al., 2017a] Quan Wen, Feng Xu, Ming Lu, and Yong Jun-Hai. Real-time 3d eyelids tracking from semantic edges. *ACM Transactions on Graphics (TOG)*, 2017.

[Wen et al., 2017b] Quan Wen, Feng Xu, and Jun-Hai Yong. Real-time 3d eye performance reconstruction for rgbd cameras. *IEEE transactions on visualization and computer graphics*, 23(12):2586–2598, 2017.

[Wikimedia Commons, 1485] Wikimedia Commons. *Sandro Botticelli - The Birth of the Venus*. Public Domain, 1485. `https://commons.wikimedia.org/wiki/File:Venus_botticelli_detail.jpg`.

[Wikimedia Commons, 1887] Wikimedia Commons. *Vincent Van Gogh - Self-Portrait*. Public Domain, 1887. `https://commons.wikimedia.org/wiki/File:VanGogh_1887_Selbstbildnis.jpg`.

[Wikimedia Commons, 2006] Wikimedia Commons. *Blue Eye Image*. GNU Free Documentation License Version 1.2, 2006. `https://commons.wikimedia.org/wiki/File:Blueye.JPG`.

[Wood et al., 2015] E. Wood, T. Baltrusaitis, X. Zhang, Y. Sugano, P. Robinson, and A. Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *IEEE ICCV*, 2015.

[Wood et al., 2016a] E. Wood, T. Baltrusaitis, L. P. Morency, P. Robinson, and A. Bulling. A 3d morphable eye region model for gaze estimation. In *ECCV*, 2016.

[Wood et al., 2016b] E. Wood, T. Baltrusaitis, L. P. Morency, P. Robinson, and A. Bulling. Learning an appearance-based gaze estimator from one million synthesized images. In *ETRA*, 2016.

[Zhai et al., 1999] S. Zhai, C. Morimoto, and S Ihde. Manual and gaze input cascaded (magic) pointing. In *Proc. of the ACM CHI Human Factors in Computing Systems Conference*, pages 246–253, 1999.

[Zhang et al., 2015] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Appearance-based gaze estimation in the wild. In *IEEE CVPR*, 2015.

[Zoric et al., 2011] Goranka Zoric, Rober Forchheimer, and Igor S Pandzic. On creating multimodal virtual humans—real time speech driven facial gesturing. *Multimedia tools and applications*, 54(1):165–179, 2011.