Detailed Spatio-Temporal Reconstruction of Eyelids

Amit Bermano[1,2] Thabo Beeler[1] Yeara Kozlov[1,2] Derek Bradley[1] Bernd Bickel[1,3] Markus Gross[1,2]
1) Disney Research Zurich 2) ETH Zurich 3) IST Austria

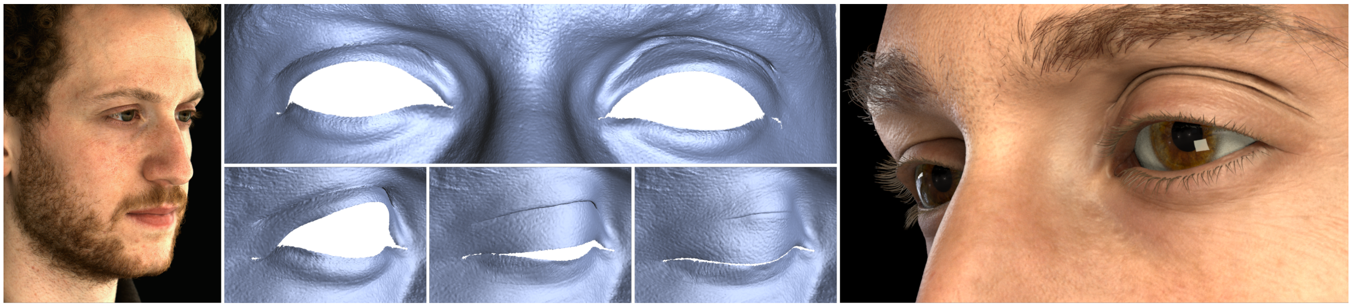


Figure 1: *Our method extends high-resolution facial performance capture with a reconstruction approach that targets eyelids. We produce detailed, spatio-temporal eyelid reconstructions, even during complex deformation and folding that occur in the eye region. The result can be used to create high-fidelity digital doubles, as shown on the right.*

Abstract

In recent years we have seen numerous improvements on 3D scanning and tracking of human faces, greatly advancing the creation of digital doubles for film and video games. However, despite the high-resolution quality of the reconstruction approaches available, current methods are unable to capture one of the most important regions of the face - the eye region. In this work we present the first method for detailed spatio-temporal reconstruction of eyelids. Tracking and reconstructing eyelids is extremely challenging, as this region exhibits very complex and unique skin deformation where skin is folded under while opening the eye. Furthermore, eyelids are often only partially visible and obstructed due to self-occlusion and eyelashes. Our approach is to combine a geometric deformation model with image data, leveraging multi-view stereo, optical flow, contour tracking and wrinkle detection from local skin appearance. Our deformation model serves as a prior that enables reconstruction of eyelids even under strong self-occlusions caused by rolling and folding skin as the eye opens and closes. The output is a person-specific, time-varying eyelid reconstruction with anatomically plausible deformations. Our high-resolution detailed eyelids couple naturally with current facial performance capture approaches. As a result, our method can largely increase the fidelity of facial capture and the creation of digital doubles.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Digitizing and scanning;

Keywords: Eyelid reconstruction, facial performance capture, eyelid modeling.

1 Introduction

The human face is the most important part of a person for conveying identity and emotion and therefore of central interest when creating realistic digital humans for computer games and films. Even subtle face motions can reveal information about the internal state of a person, where their attention is focused, the intention of actions, and interpretation of (non-)verbal communication. In recent years, we have witnessed tremendous progress in facial scanning, performance capture, and skin rendering. Stunning attempts to achieving a photorealistic actor such as *The Curious Case of Benjamin Button* or *The Digital Ira Project* [Alexander et al. 2013] are examples that high-resolution face scanning processes are key to cross the ”uncanny valley” that divides a synthetic-looking face from a photo-realistic virtual person.

For identifying emotions, humans mainly use a consistent selective sampling of visual information from the eye region and, to a lesser extent, the mouth region [Smith et al. 2005; Peterson and Eckstein 2012]. Subtle details such as the twitch of an eyelid and the formation of small wrinkles significantly contribute to the realism of human faces and the perception of emotions. However, despite the important role of the eye region, existing capture technology is usually not able to provide an adequate level of geometric detail and motion to reproduce these subtleties. In practice, achieving realistic eyelid motions and skin deformation of the surrounding area requires significant manual modeling efforts by highly skilled artists.

Acquiring this region is extremely challenging due to several reasons. In an expressive performance, eyelids undergo extreme deformations and wrinkling. The skin rolls and folds inward when the eye opens, and stretches over the eyeball when the eye is shut. Due to concavities and eyelashes, there is significant self-shadowing, inter-reflections, and partial occlusions. Even worse, in many facial expressions a significant part of the eyelid is folded in and not visible at all. We desire an accurate performance capture that delivers consistent geometry in correspondence over time whenever visible, and deforms non-visible parts in a plausible way. Unfortunately, existing dense performance capture approaches cannot handle these extreme deformations and occlusions.

We address this problem by introducing a novel reconstruction and tracking scheme that combines a geometric deformation model with image-based data. The model is motivated by the physiological behavior of the eye - skin interface and constrains the reconstruction to anatomically plausible motions. This prior is required due to noise and missing data in the depth information, inherently caused by eyelashes and self-occlusions. In addition, we observe that wrinkles greatly change the local appearance of the skin, and thus their location can be accurately determined from images and local motion. This combination of anatomically motivated priors, depth information, and image-based data, makes detailed eyelid reconstruction feasible.

Such an approach has several advantages. We obtain a single, consistent mesh over time, which allows our result to be directly combined with existing facial performance capture approaches, or to be used to create a data-driven blendshape rig. Our method provides plausible deformations even for regions that are occluded. We are able to capture the dynamic effects of eyelids, which is important because the location and shape of wrinkles is not only dependent on the current state of the eye region but also its history, a phenomenon referred to as hysteresis. As our method is agnostic to the capture approach, it can be easily integrated into any performance capture pipeline, be it passive or active, that records sufficiently high-resolution footage of the eye region. As we demonstrate with several results, our system allows the reconstruction of an expressive, dynamic model of the eye region at a quality level that has never before been possible, increasing the fidelity of this very important facial component in the creation of digital doubles.

2 Related Work

Our work on capturing eyelids falls into the general category of 3D face capture, and it is particularly related to other techniques that are tailored for reconstructing or modeling the eye region. At a higher level, our approach is akin to methods that model or generate wrinkle geometry for faces and clothing. In the following we discuss previous work in these areas.

Face Capture. Several methods have been proposed for high-quality static face capture, including both active [Ma et al. 2007; Ghosh et al. 2011] and passive [Beeler et al. 2010] approaches. Dynamic facial performance capture has also received a lot of attention in recent years [Bradley et al. 2010; Beeler et al. 2011; Huang et al. 2011; Klaudiny and Hilton 2012]. While these advances have resulted in large steps towards creating realistic digital faces, no method has yet succeeded to reconstruct the complex shape and motion of eyelids.

Reduced hardware approaches such as binocular [Valgaerts et al. 2012] and monocular [Garrido et al. 2013; Suwajanakorn et al. 2014; Shi et al. 2014] facial capture methods often use shape from shading approaches [Wu et al. 2011] to recover fine-scale facial details, however these methods are only able to approximate eyelids as simple skin folds without strong occlusions.

Real-time facial animation methods sacrifice quality in favor of reconstruction speed, typically through the use of generic facial priors [Weise et al. 2011; Rhee et al. 2011; Cao et al. 2013; Bouaziz et al. 2013; Li et al. 2013; Cao et al. 2014]. These techniques are able to animate generic eyelid shapes that are part of the shape priors, but are unable to recover person-specific eyelid details.

Eyes and Eyelids. The important role that eyes play in computer graphics applications has led to a number of research topics including eye motion and blink animation, iris and eyelid modeling, and high-quality eye capture. A detailed survey of eye and gaze animation methods was recently presented by Ruhland et al. [2014]. In the following we summarize the most related methods.

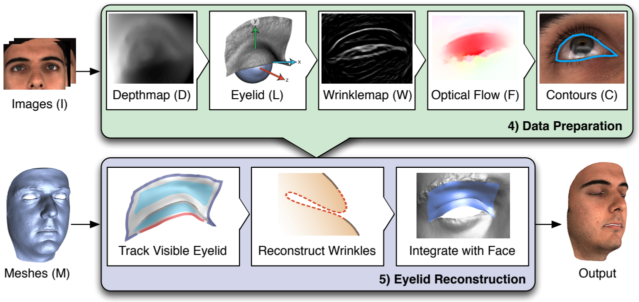


Figure 2: *Overview of the system. Starting from passively acquired image data, we compute for each frame depth information, optical flow, track the contours of the eye-skin interface, and the probability where wrinkles are forming (Section 4). Based on this data, our tailored deformation model deforms the eyelids over time, accurately tracking the actors performance (Section 5). Finally, the eyelid meshes are integrated with the reconstructed facial performance to provide a complete face model.*

Since the seminal work of Lee et al. [2002] on keeping a character's eyes "alive", several models of eye motion and blinks have been derived from motion capture and video data [Deng et al. 2005; Weissenfeld et al. 2010; Trutoiu et al. 2011; Le et al. 2012]. However, these approaches do not focus on detailed eyelid reconstructions that would provide person-specific eyelid wrinkle formation.

On the topic of reconstruction, François et al. [2009] estimate the multilayered shape and approximate scattering properties of the iris from a single camera image. More complete and detailed reconstructions were recently shown by Bérard et al. [2014], who use a multi-camera and multi-light setup to capture all the visible components of the eye in very high-resolution. Note that these methods are complementary to ours since they focus exclusively on the eyes and we are considered with the skin surrounding it.

Our work is not the first to model eyelids. A method for modeling the eye region has been presented by Sagar et al. [1994] in the context of surgical simulation. The eyelids are modeled by a simple NURBS surface that is fit to a single face scan and then manipulated by hand. In contrast, we reconstruct temporally-varying high-resolution geometry, capturing the unique wrinkling behavior of each individual eyelid.

Wrinkle Modeling. One of the defining characteristics of eyelids is their natural self-folding behavior, which can be considered an extreme case of wrinkling. Bickel et al. [2007] decompose the face into multiple scales in order to enhance low-resolution marker-based motion capture with fine-scale wrinkles captured on the forehead and under the eyes, and then learn the correspondence of skin-strain to wrinkle formation for real-time editing and wrinkle transfer to virtual characters [Bickel et al. 2008]. Dynamic wrinkles are also captured by Ma et al. [Ma et al. 2008] and modeled as a polynomial displacement map on top of a low-resolution face model. This approach of modeling wrinkles as a displacement map layer is very common in facial animation [Kähler et al. 2002; Dutreve et al. 2011; Li et al. 2015]. Skin wrinkles can also be generated through physically-based simulation of the face if the anatomy is sufficiently modeled [Magnenat-Thalmann et al. 2002; Zhang et al. 2005; Warburton and Maddock 2014]. Other techniques to enhance low-resolution facial performances with high-resolution wrinkles include mapping expressions into a common shape space and trans-

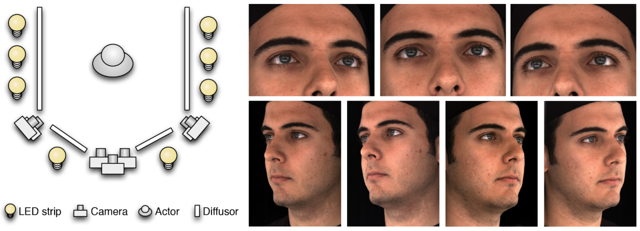


Figure 3: *The setup (left) consists of seven cameras, where the three central ones are zoomed in. LED strips mounted around the actor and diffused by frosted paper provide a flat illumination. An exemplary dataset for one frame is shown on the right.*

ferring the high-frequency details [Bermano et al. 2014], or manually specifying wrinkle curves with an artistic tool [Bando et al. 2002; Larboulette and Cani 2004]. While these methods target wrinkle modeling on the face, they do not address the complex wrinkling behavior of eyelids.

For clothing, a similar trend has emerged to enhance low-resolution cloth models with previously-generated fine-scale wrinkles [Müller and Chentanez 2010; Feng et al. 2010; Wang et al. 2010; Rohmer et al. 2010; Seiler et al. 2012; Kim et al. 2013; Zurdo et al. 2013]. These methods typically assume a coarse cloth simulation is available, which provides the underlying motion of the surface, and thus are not well-suited for capture scenarios. On the other hand, Popa et al. [2009] procedurally generate wrinkles for captured garments. Starting with a reconstruction of the low-frequency garment shape [Bradley et al. 2008], they introduce temporally coherent high-frequency wrinkles that correspond to detected edges in the capture images. Such an approach could also improve the shape of wrinkles on a captured face, however the method does not consider wrinkles with strong self-occlusions such as eyelids.

In contrast to existing work, we present the first method specifically designed to capture person-specific eyelids, including the complex temporal behavior and self-folding that occurs during opening and closing of the eyes. Our work naturally complements existing techniques for face and eye capture, forming a more complete and realistic digital face.

3 Method Overview

Our system, schematically depicted in Figure 2, starts by capturing a performance of the eyes using off-the-shelf cameras, as described in Section 4.1. The images are then analyzed to remove eyelashes and generate a spatiotemporal reconstruction of the face shape along with per frame depths maps (Section 4.2). Optical flow computed frame by frame is misled by the skin wrinkling and needs to be corrected (Section 4.5) using wrinkle probability maps which indicate where wrinkles are most likely to form (Section 4.4). Finally, we also track accurate eyelid contours over time (Section 4.6) to ensure faithful reconstruction of the visually important interface between the eyelid and the eye.

For all four eyelids, we manually create template meshes (Section 4.3) which will be continuously deformed from frame to frame. As shown in the lower part of Figure 2, eyelid tracking starts by deforming the visible part of the eyelid using constraints from optical flow, tracked eyelid contours and the surrounding face mesh (Section 5.1). We then reconstruct the parts of the eyelid which were subject to wrinkling and are thus not visible. The reconstruction produces plausible wrinkles that are visually pleasing (Section 5.2). Lastly, we use the eyelid templates as control meshes to deform the face mesh (Section 5.3) resulting in a complete facial performance with accurately tracked eyelids as shown in the bottom right corner of Figure 2.

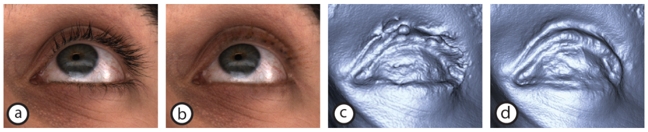


Figure 4: *Eyelashes pose problems for the reconstruction as they occlude the underlying skin (a) and confuse stereo methods, leading to noisy geometry (c). We adopt the inpainting approach proposed by Beeler et al. [2012a] to remove the eyelashes (b), which improves the reconstructed geometry substantially (d).*

4 Data Preparation

In this section we describe how to generate and prepare the input data required for eyelid reconstruction.

4.1 Data Acquisition

The image data is acquired using a multi-view setup consisting of seven synchronized video cameras, each providing roughly 40 frames per second at about 1MP. Three cameras are zoomed in to get higher resolution on the eye region and the other four are split in pairs of two to capture the full face. As illumination, we mount LED strips on a cage around the actor and diffuse them with frosted paper. Figure 3 shows the setup on the left and the images captured at one point in time on the right.

4.2 Face Mesh Reconstruction

From the acquired images, we reconstruct the spatio-temporal shape of the face using the method of Beeler et al. [2011], which provides high-resolution per-frame tracked meshes in dense correspondence. In addition to the tracked meshes, we also compute per-frame depth maps D using Beeler et al. [2010], which contain information in areas not covered by the tracked meshes, such as the eyes. One major problem for stereo-based reconstruction methods are the eyelashes, which occlude the underlying skin and confuse stereo matching, causing considerable artifacts in the reconstructed depth maps. Therefore we adopt the inpainting approach proposed by Beeler et al. [2012] to remove the eyelashes from the input images before reconstruction, which greatly improves the reconstruction quality as can be seen in Figure 4.

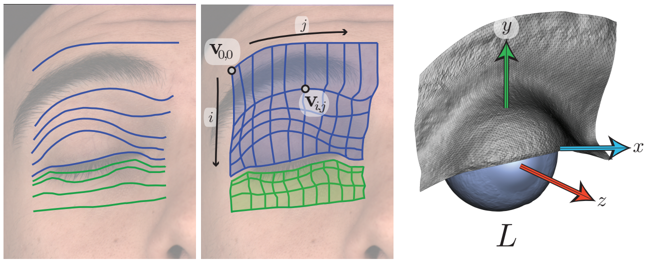


Figure 5: *The eyelid mesh is generated by manually drawing a few curves on the rest pose frame (left). From these contours a 2D grid is created (center) with the origin $\mathbf{v}_{0,0}$ in the top left corner, rows i running down and columns j to the right. From this grid the 3D eyelid mesh L is created using the depth maps and a reference coordinate frame is established (right).*

4.3 Eyelid Initialization

We manually create a template mesh for each of the four eyelids (upper and lower, left and right) once per actor. Per Section 5, the template meshes should consist of a regular grid of vertices $\mathbf{v}_{i,j}$, in which the rows i are aligned with the predominant wrinkle orientation and the columns j run orthogonal across the wrinkles. This structure allows to efficiently process the eyelid area to detect and reconstruct wrinkles (Section 5.2). One way to create the template meshes would be to model them in 3D. This, however, would require the user to be familiar with 3D modelling. Instead we propose a simpler means to generate the template meshes by drawing a few curves on a closed-eye image of the actor (see Figure 5, left). From these curves a 2D regular grid is created (center) and lifted to 3D using the computed depth maps D to generate the eyelid (right). In our experiments, we have traced 10 curves for each eye, loosely approximating wrinkle flow lines, from which a grid of size 120×100 vertices is generated. We found that the best expression to use is one where the eyes are closed and the eyebrows are raised, since the entire eyelid is visible and the skin is least compressed. We will refer to this as the rest pose later on.

The user also initializes a reference coordinate frame, approximately in the center of the eye socket, with the z-axis pointing forward and the y-axis up (Figure 5, right). This coordinate frame follows the rigid head motion computed by rigid stabilization [Beeler and Bradley 2014] and is used both to reconstruct the eyelid wrinkles (Section 5.2) and to compute the wrinkle probability map described in the next section.

4.4 Wrinkle Probability Map

The wrinkle probability map encodes the likelihood that a pixel is part of a wrinkle, and is computed for each frame from the inpainted and histogram normalized images using oriented kernels. Specifically we employ anisotropic difference of Gaussians $\mathcal{N}(\sigma_x, \sigma_y, \theta) - \mathcal{N}(\sigma_x)$ for seven different orientations θ in the range of $\pm 20°$, where we set $\sigma_x = 8$ and $\sigma_y = 0.1\sigma_x$, and record the maximum response in the wrinkle map. Other oriented kernels, such as Gabor, could also be applied. While this identifies wrinkles it also captures a lot of noise caused by areas of similar appearance. To improve the signal-to-noise ratio we propose the following three steps. First, since the wrinkles we are interested in tend to form concentrically around the center of the eyes in the images, we rotate the oriented kernel based on the relative position to the closest eye center. Second, we employ spatio-temporal hysteresis [Canny 1986], which keeps only pixels whose probability is either higher than a given threshold ξ_u, or which are connected to such pixels in space or time via some other pixels with probabilities no lower than ξ_l. We use $\xi_u = 0.05$, $\xi_l = 0.01$ for all results. Third, since the inpainting might have missed a few eyelashes, which can happen if they cluster, we consolidate wrinkle maps from multiple views and filter wrinkle probabilities where the views do not agree.

4.5 Optical Flow

To be able to track the eyelid over time, we compute optical flow F from one frame to the next using the method of Brox et al. [2004] on the inpainted images. A source-sink map S encodes the density of the optical flow and is computed by accumulating the inbound flow vectors for every pixel. Areas where the flow vectors diverge are considered a source and appear dark in the visualization, and areas where they converge are considered a sink and they appear bright (see Figure 6). While generally very reliable, optical flow performs poorly at the wrinkles. Despite the motion of the eyelid surface during wrinkle formation, the appearance around the wrinkle remains similar due to shading, and this can confuse the flow computation. The incorrect flow vectors become very apparent when inspecting the source-sink map (Figure 6, top row). As we can see, the flow compresses on both sides of the wrinkle and not inside of it, which would be the correct sink.

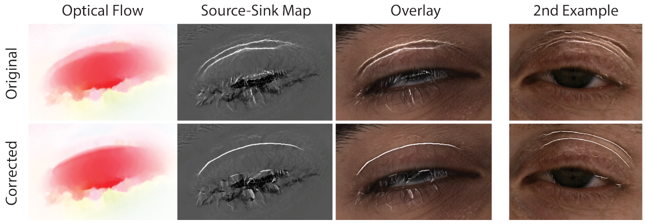


Figure 6: *Wrinkling poses problems for optical flow since the appearance changes and parts become occluded. The original flow shown in the top row is inaccurate around the wrinkle and compresses on both sides of the wrinkle, as shown in the source-sink map. Correcting the flow provides the desired behavior where the flow converges into the wrinkle instead. A second example with a double wrinkle is shown in the last column.*

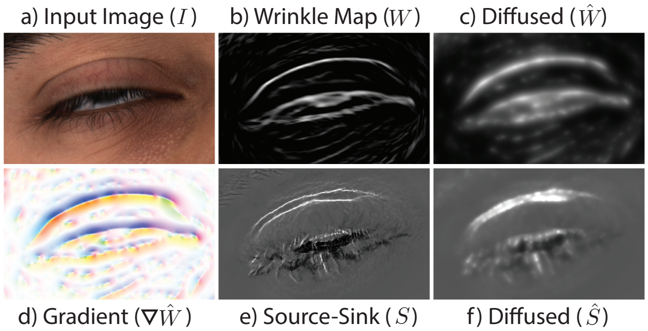
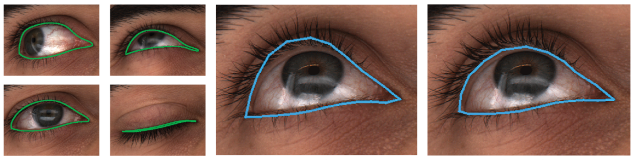


Figure 7: *To correct the flow maps we employ different variants of diffusion (Section 4.5). First the wrinkle probability map (b) is diffused isotropically with retention (c). The gradient of the diffused map (d) encodes the direction to the closest wrinkle and is employed to diffuse the source-sink map (e) anisotropically with retention (f) and finally the flow as shown in Figure 6.*

We devise a method to correct the optical flow. Using guidance from the source-sink map S, the wrinkle probability map W (Section 4.4) is diffused smoothly to spread out the probabilities. As we do not intend to reduce existing probability but just spread it out, we iteratively update the probability map using

$$\hat{W}^{k+1} = \max\left(\hat{W}^k, \mathcal{N}\left(\hat{W}^k\right)\right), \tag{1}$$

where $\hat{W}^k$ denotes the diffused wrinkle probability map at the k-th iteration, $\hat{W}^0 = W$ and $\mathcal{N}$ is a Gaussian filter of size 7×7 (and an eye region is approximately 400 pixels wide in the image). The number of iterations required is determined by the distance of the sink to the true wrinkle location, which we found to be consistent in our examples and thus the same number of iterations (30) were applied to all frames, leading to a smoothed map as shown in Figure 7.c. The gradient of this map $\nabla\hat{W}$ (Figure 7.d) indicates the direction towards the closest wrinkle and will be used to correct the flow in a two-step process. First, we diffuse the source-sink map S (Figure 7.e) towards the wrinkle center, as this will determine the area in which the flow needs to be corrected. To do so we employ a variant of anisotropic diffusion [Perona and Malik 1990]:



(a) Input contours (b) Initial result (c) Refined contour

Figure 8: *Eyelid contour tracking pipeline: (a) subset of reference contours used for training, (b) initial tracking produces an estimate of the contour shape, but is not accurate enough for good localization, (c) the reference contour closest to the initial estimate is deformed using optical flow to refine the initial estimate.*

$$\hat{S}^{k+1} = \hat{S}^k + \lambda\psi\left(c\left(\nabla\hat{W}\right)\nabla\hat{S}^k\right), \qquad (2)$$

where $\hat{S}^k$ denotes the diffused source-sink map at the k-th iteration and $\hat{S}^0 = S$. Instead of preventing smoothing along the gradient as was the goal in Perona and Malik [1990] we control the diffusion to spread predominately in the positive direction of the gradient. As diffusion coefficient c, we therefore choose

$$c\left(\nabla\hat{W}\right) = \left| e^{-\left(\frac{\nabla\hat{W}}{\kappa}+1\right)} \right|_{0,1}, \qquad (3)$$

where κ (0.01) controls the sensitivity and $|\cdot|_{0,1}$ clamps to the range of $[0, 1]$ to warrant the maximum principle. The retaining function $\psi(x)$ attenuates the decay by multiplying x with a user given parameter (0.1) whenever $x < 0$, thus spreading this information to a larger region. The timestep λ was set to $1/8$ and the diffusion is run for 60 iterations leading to the result shown in Figure 7.f.

We employ $\hat{S}$ to attenuate diffusion of the flow field outside of the wrinkle neighborhood by including it in the diffusion coefficient as

$$c\left(\hat{S}, \nabla\hat{W}\right) = \left| \hat{S}e^{-\left(\frac{\nabla\hat{W}}{\kappa}+1\right)} \right|_{0,1} \qquad (4)$$

and then diffuse the flow field F using Equation 2. To prevent flow vectors from overshooting the wrinkle location, we only update them if the wrinkle probability gradients at the origin and destination of the flow vector point in the same direction, i.e. the flow remains on the same side of the wrinkle. Figure 6 shows how the original flow and source-sink map (top row) are corrected by this approach (bottom row).

4.6 Eyelid Contours

The time-varying 2D eyelid contours are invaluable constraints for reconstructing accurate eyelid deformation. For this reason we also pre-compute contour curves for each frame. The contours are tracked in image space from a single front view using a two step method. First, we compute an initial contour shape estimate using the regression framework proposed by Cao et al. [2012]. This framework has shown to work well on related problems [Cao et al. 2013; Cao et al. 2014] but any similar system, such as active appearance trackers [Cootes et al. 2001], may be employed. We then refine the contour position in image space using optical flow.

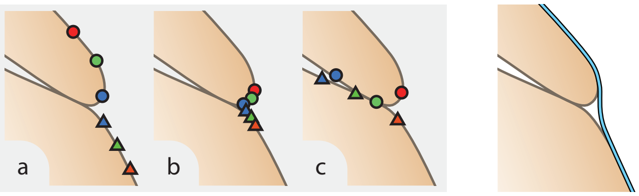


Figure 9: Left: *During wrinkling vertices are compressed from their initial position (a) into the wrinkle location (b) since they become occluded. Section 5.2 describes how the proposed deformation model moves them into the wrinkle in an anatomically plausible way (c).* Right: *Estimated depth (blue line) is inaccurate at the wrinkle location since the multi-view stereo method cannot resolve small scale details given the input image resolution.*

For each actor we choose a small set of frames in which we manually trace the eyelid contour (Figure 8.a). Each of these reference contours is represented by a set of landmarks, placed equidistantly along the contour from the inner to the outer eye corner. From these samples we then train an eye-specific contour tracker. In our experiments, we used a reference set of $20 - 30$ contours, and 20 landmarks. For convenience, the reference set was constructed by starting with an initial blink sequence, and was then iteratively expanded by adding frames that caused tracking failures (up to at most 3 iterations).

To track the contours over the sequence, we apply the contour tracker on the frames taken from the same view. Each frame's tracking result is used to initialize tracking in the next frame. The tracking results provide a good initial estimate of the contour shape and position (Figure 8.b), but are not sufficient to accurately constrain the eyelid reconstruction, and thus need to be further refined. For each frame we retrieve the most similar reference frame by comparing the shape of the predicted contour to the reference contours. We then compute optical flow [Brox et al. 2004] between the reference image and the current image and use the flow vectors to deform the reference contour into the current frame yielding subpixel-accurate registration to the reference frame (Figure 8.c).

However, sequential frames may be matched to different reference frames, which could lead to temporal jitter since the reference contours themselves exhibit some inaccuracies as they are hand-drawn. We thus smooth the contours temporally over the entire sequence using optical flow computed between frames to produce accurate, temporally smooth eyelid contours.

5 Eyelid Reconstruction

In this section, we describe our eyelid deformation model and our method for robust eyelid reconstruction. Our goal is to evolve the eyelid created in Section 4.3 over time t. The eyelid is represented by a template mesh L, which consists of regularly sampled vertices $\mathbf{v}_{i,j}$ along the horizontal direction j corresponding to the dominant main wrinkle orientation and the orthogonal vertical direction i, as explained in Section 4.3 and illustrated in Figure 5. Our deformation model follows a two-step process: first, as described in detail below, we deform the skin based on optical flow and tracked contour data. This provides the desired behavior in areas visible both in the current and previous frames and undergoing moderate deformation, but optical flow is unreliable in more challenging cases. Even though the flow correction described in Section 4.5 improves the reliability in regions visible in both frames, it is unable to guide the deformation of the occluded mesh regions. Thus, for newly occluded regions, vertices will be compressed at the wrinkle location (see Figure 9 (left)). To address this challenge we identify wrinkle regions and propose a dedicated wrinkle model that is parameterized with a small set of distinctive feature points. These feature points can be efficiently estimated from the acquired data and allow plausible reconstruction even in regions with extreme skin deformations.

5.1 Visible Skin Deformation

The first step of our deformation model is driven by the visible areas of the skin - we deform the eyelid using optical flow where it can be trusted, while respecting tracked boundary conditions from the surrounding face. The extreme deformations occurring around the wrinkle areas are handled in a second step, described in Section 5.2.

Close inspection of the eyelid reveals that the eye-eyelid interface transforms mostly rigidly, as it fits tightly around the eye shape while sliding over it. Consequently, this area preserves its shape based on the underlying eye and mostly just rotates when the lid opens – unlike the rest of the eyelid, which undergoes strong wrinkling. To reflect this we combine two linear thin-shell energies [Botsch and Sorkine 2008] to deform the eyelid (E_S, E_I), guided by three different data terms (E_C, E_B, E_F). Figure 10 illustrates the spatial distribution of these energies on the mesh. The first energy regularizes the deformation based on the shape of the lid at the previous frame L^{t-1} and is given as

$$E_S = \sum_{V} \left\| \Delta_{L^{t-1}} (\mathbf{v}_{i,j}^t - \mathbf{v}_{i,j}^{t-1}) \right\|^2 , \tag{5}$$

where $\Delta_{L^{t-1}}$ is the discrete Laplace-Beltrami operator for the eyelid mesh L^{t-1} and V denotes all vertices of the mesh. The second energy reflects the deformation driven by the rigidly transforming region at the eye-eyelid interface V^I. It seeks to deform this region to match the rest pose, up to a global rotation R^t:

$$E_I = \sum_{V^I} \left\| \Delta_{L^0} (R^t \mathbf{v}_{i,j}^t - \mathbf{v}_{i,j}^0) \right\|^2 , \tag{6}$$

where Δ_{L^0} is again the discrete Laplace-Beltrami operator, this time computed from the rest pose. Note that the two Laplace-Beltrami operators only differ in the cotangent weights, which is required to account for skin compression while the lid opens. Figure 10.a depicts the regions regulated by these energies.

The eye-eyelid interface itself does not transform purely rigidly, but undergoes some deformation due the shape of the underlying eye. We account for this by incorporating the contours computed in Section 4.6 as an energy term

$$E_C = \sum_{V^C} \| P(\mathbf{v}_{i,j}^t), C^t \|_\ell^2 , \tag{7}$$

where $\|\cdot,\cdot\|_\ell$ denotes the point-line distance in image space computed by projecting $\mathbf{v}_{i,j}^t$ into the camera image using the camera projection matrix P. The contour imposes constraints on the vertices V^C at the eye-eyelid interface of the eyelid mesh L. The remaining boundary should deform such that it is compatible with the face mesh M to alleviate integration later on (Section 5.3). For the vertices V^B in the outer two rings at these boundaries, we wish to constrain the motion to be similar to the motion of the corresponding points C^B on the face mesh. A correspondence $\mathbf{c}_{i,j}^B \in C^B$ is computed as the closest point in rest pose to the eyelid vertex $\mathbf{v}_{i,j}^0 \in V^B$. Encoding the correspondence in barycentric coordinates of the adjacent triangle allows to propagate it in time consistently with the face mesh. The boundary energy term is then

$$E_B = \sum_{V^B} \left\| \mathbf{v}_{i,j}^t - \mathbf{c}_{i,j}^{B,t} \right\|^2 , \tag{8}$$

The valid vertices in the interior of the eyelid $\mathbf{v}_{i,j}^t \in V^{F,t}$ at time t are constrained by optical flow. We compute positional constraints $\mathbf{c}_{i,j}^{F,t} \in C^{F,t}$ for these vertices by projecting them into the main camera's image plane, advecting them using the optical flow F and elevating them back into 3D using the depth maps D. A vertex is considered to be valid if it is (1) visible from the main camera, (2) does not exhibit a high enough wrinkle probability, and (3) is sufficiently far from the boundary (we use a 5-ring margin from the border in all our experiments). The flow energy term

$$E_F = \sum_{V^{F,t}} \gamma_{i,j} \left\| \mathbf{v}_{i,j}^t - \mathbf{c}_{i,j}^{F,t} \right\|^2 \tag{9}$$

where $\gamma_{i,j}$ is a confidence weight indicating how much the constraint can be trusted. The confidence is provided by the multi-view geometry reconstruction method and is a measure of how similar the neighborhood of this vertex looks in the different views. This helps overcome outliers caused, for example, by occluding eyelashes. Note that the vertex set $V^{F,t}$ associated with E_F may change over time. The vertex sets associated with all other energy terms remain constant throughout the sequence. Figure 10.b illustrates the mesh regions contributing to each data term.

Combining the individual terms together yields the total energy

$$E = \lambda_F E_F + \lambda_B E_B + \lambda_C E_C + E_I + E_S , \tag{10}$$

where $\lambda_{F,B,C}$ are user parameters. In our experiments we fixed $\lambda_F = \lambda_B = 30$ and $\lambda_C = 300$.

Unfortunately, the rotation R^t used in E_I (Equation 6) is also unknown and needs to be estimated as well. Following Sorkine and Alexa [2007] we interleave estimation of deformation and rotation and iterate both of them three times, starting with estimating the deformation.

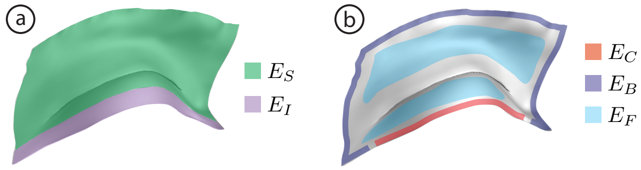


Figure 10: *(a) The visible skin deformation is regulated by two thin-shell energies: E_S regulates most of the eyelid for temporal smoothness, and E_I regulates the eye-eyelid interface for rigidity relative to the rest pose. (b) Regions contributing to each data term: tracked contours contribute to E_C, the interface with the face mesh to E_B, and visible interior regions contribute to the flow term E_F.*

5.2 Wrinkle Reconstruction

After deforming the visible parts of the eyelid, we process the hidden and newly occluded areas. For these areas flow computation is not possible and the best the flow correction (Section 4.5) can do is to compress the problematic vertices into the wrinkle area. Figure 9.b depicts schematically how the vertices from Figure 9.a are aggregated at the wrinkle location. The problem is further aggravated since the multi-view stereo method cannot accurately reconstruct small scale details (Figure 9. (right)) and as a consequence the geometry in the wrinkle area cannot be trusted. In this section we describe how we create an anatomically plausible wrinkle shape and move these vertices into the wrinkle valley in a physically meaningful manner (Figure 9.c).

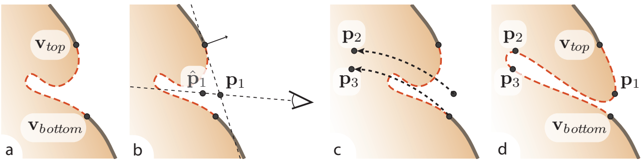


Figure 11: *(a) The top and bottom vertices bounding a wrinkle area in the cross section are defined as $\mathbf{v}_{top,bottom}$. These are the last vertices where we rely on the result from Section 5. (b) The front-buckle $\mathbf{p}_1$ is computed by projecting the weighted average $\hat{\mathbf{p}}_1$ of all visible vertices in the wrinkle area along the ray to the main camera onto the extension from $\mathbf{v}_{top}$. (c) The two back-buckle points $\mathbf{p}_{2,3}$ are computed by rotating $\mathbf{p}_1$ and $\mathbf{v}_{bottom}$ around the eye center. (d) The wrinkle is constructed as membrane from these feature points allowing the vertices in the wrinkle area to relax into the wrinkle.*

We start by projecting the wrinkle probability map W^t onto the eyelid L^t and assign a wrinkle probability $w^t_{i,j}$ to every vertex. The eyelid mesh has been designed such that it allows to efficiently identify distinctive wrinkle feature points from which we can construct the hidden part of the wrinkle. Figure 11 illustrates the extracted feature points schematically. Traversing a wrinkle cross-section from top to bottom will sequentially produce the *top of wrinkle* ($\mathbf{v}_{top}$), the *front-buckle* ($\mathbf{p}_1$), the *back-buckle top* ($\mathbf{p}_2$), the *back-buckle bottom* ($\mathbf{p}_3$) and the *bottom of wrinkle* ($\mathbf{v}_{bottom}$). Note that $\mathbf{v}_{top}$ and $\mathbf{v}_{bottom}$ correspond to actual vertices of the mesh, where $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ are points in space. The individual feature points are computed as follows:

1. *Top of wrinkle* ($\mathbf{v}_{top}$): First vertex with wrinkle probability $w^t_{i,j} > \xi$ when traversing the vertices $\mathbf{v}^t_{i,j}$ of a column j from top to bottom. We set $\xi = 0.1$ for all our results.

2. *Bottom of wrinkle* ($\mathbf{v}_{bottom}$): Last vertex with wrinkle probability $w^t_{i,j} > \xi$. The vertices between $\mathbf{v}_{top}$ and $\mathbf{v}_{bottom}$ denote the wrinkle segment ($\mathbf{v}_{top}$ to $\mathbf{v}_{bottom}$) (Figure 11.a).

3. *Front-buckle* ($\mathbf{p}_1$): Computed by projecting the weighted average $\hat{\mathbf{p}}_1$ of all visible vertex positions in the wrinkle segment onto the extended plane from $\mathbf{v}_{top}$ (Figure 11.b). The weighted average is computed as $\hat{\mathbf{p}}_1 = \sum w^t_{i,j}\mathbf{v}^t_{i,j} / \sum w^t_{i,j}$ over all vertices in the wrinkle segment, where $w^t_{i,j}$ is the wrinkle probability associated with vertex $\mathbf{v}^t_{i,j}$.

4. *Back-buckle top* ($\mathbf{p}_2$): From human anatomy it is reasonable to assume the wrinkle folds inwards on an orbit around the eye. To compute $\mathbf{p}_2$ we thus rotate $\mathbf{p}_1$ inwards around the eye center by half the geodesic distance from $\mathbf{v}_{bottom}$ to the vertex closest to $\mathbf{p}_1$ computed in the rest pose. See Figure 11.c for a schematic depiction. The skin compresses due to micro-wrinkles. We account for this by adjusting the rotation magnitude by the area ratio between neighboring visible skin in the current frame and in the rest pose.

5. *Back-buckle bottom* ($\mathbf{p}_3$): Computed analogously to $\mathbf{p}_2$ by rotating $\mathbf{v}_{bottom}$ instead of $\mathbf{p}_1$.

Next we want to create the wrinkle as a membrane (Figure 11.d) that smoothly transitions into the visible part of the lid defined by the two vertices $\mathbf{v}_{top,bottom}$ and also closely approximates the three inner feature points ($\mathbf{p}_{1,2,3}$) defined above. To resolve the compression problem shown in Figure 9.b, vertices are allowed to move freely on the membrane surface, in order to relax and slide into the wrinkle. We achieve this by alternating between two stages. The first stage relaxes the vertices in the wrinkle area by applying one iteration of Laplacian smoothing, which optimizes the surface to reduce stretching. This moves vertices into the wrinkle, but also potentially pulls them away from the intended wrinkle shape. We therefore apply a second stage, where we find the nearest vertex on the membrane for every feature point and constrain their positions to the feature points, while again solving for the membrane energy. This second step pulls the surface back towards the desired shape. For both stages we use Neumann boundary conditions at the border of the wrinkle area to ensure a smooth transition into the visible part of the eyelid. We repeat the two stages six times after which we found the vertices to have relaxed inside the wrinkle.

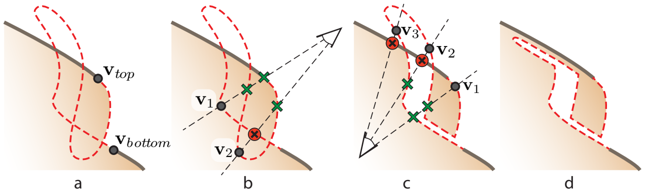


Figure 12: *The constructed wrinkle is not guaranteed to be free of self-intersections (a). The proposed method resolves self-intersections leveraging the fact that it is always the upper part of the lid folding over the lower. Thus we traverse the wrinkle area from bottom to top testing for occlusions by lower parts (b). We then move the occluded vertices in front of the occluding surface and reverse the procedure from top top bottom testing for occlusions with respect to the eye (c). Alternating these steps several times produces an intersection free wrinkle (d).*

Self-Intersection Handling. The aforementioned process is not guaranteed to be free of self-intersections. Specifically, the formed wrinkle might protrude out from the visible part of the eyelid, or the smoothing might cause the wrinkle to intersect with itself. Figure 12.a shows an extreme case for illustration. Our main concern is to prevent any visually distracting artifacts and thus we wish to resolve self-intersections that are visible from the outside. To efficiently test for and correct self-collisions we can leverage the anatomy of the eyelid. The wrinkles form in such a way that skin farther away folds over skin closer to the eye. In terms of our eyelid model this means that vertices further down a cross-section should never occlude vertices which are higher up. Our algorithm sequentially traverses the vertices in the wrinkle area from $\mathbf{v}_{bottom}$ to $\mathbf{v}_{top}$, where the vertex indices decrease from *bottom* to *top* ($bottom > i > top$). For every vertex $\mathbf{v}_i$, the method tests if $\mathbf{v}_i$ is occluded by a lower part of the eyelid (i.e. a triangle that contains at least one vertex $\mathbf{v}_{k,j}$ with $k > i$). If the vertex is occluded, it is moved in front of the occluding triangle. Visibility is determined from the point of view of the main camera. Figure 12.b depicts two possible scenarios: while the method will not report a self-intersection for $\mathbf{v}_1$ since it is only occluded by higher up parts, it will correctly identify and correct $\mathbf{v}_2$, which is occluded by lower parts. Once all vertices of the eyelid have been processed, we reverse the order and evaluate the vertices relative to the center of the eye (Figure 12.c). The vertices are now traversed from $\mathbf{v}_{top}$ to $\mathbf{v}_{bottom}$ and the method checks if a vertex is occluded by parts higher up. We alternate between these two steps until no more occluded vertices detected, which is typically within 3 iterations. The resulting wrinkle is now guaranteed to be behind the visible surface (Figure 12.d).

5.3 Integration

Finally, we integrate the tracked eyelid with the full face, which is provided by Beeler et al. [2011]. Their method uses the concept of *anchor frames*, which states that during a facial performance certain

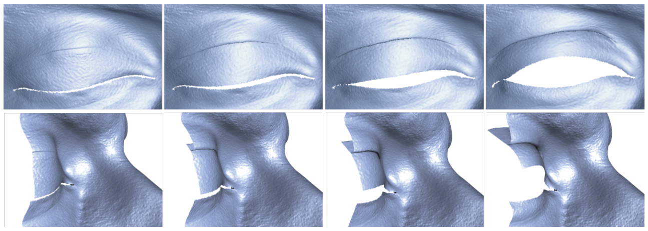


Figure 14: *Looking closely at the formation of a wrinkle we see the complex temporal dynamics of an eyelid. Our method is able to capture the skin folding under and creates a plausible eyelid shape, as seen from front (top) and from a side view cut-away (bottom).*

expressions will re-occur and they thus propose to pick a reference frame that is similar enough to the anchor frames to be able to track directly to them. This concept is also very useful in our scenario, as we found that tracking from closed eyelids is preferable. We therefore pick a frame with a neutral expression and closed eyelids as reference frame and construct the eyelid mesh from this frame to facilitate the integration with the face mesh (Figure 5). The lid is naturally aligned to the face mesh and we can establish dense correspondences between the two. We then use the eyelid to drive the deformation of the face mesh in this area. Since we made sure that the boundary of our eyelid deforms in a compatible manner to the face mesh (Section 5.1) the integration is seamless.

As there are many eyelid wrinkles at the micro- and mesoscopic scales during deformation, we apply mesoscopic optimization and temporal smoothing following Beeler et al. [2011] to produce temporally consistent high frequency details seamlessly across the full face including the eyelids.

6 Results

Eyelids are extremely unique and can produce extremely different wrinkles. This variance is not only visible from person to person but the shape and temporal deformation of the eyelids also differ substantially between the left and right eye of the same person. To demonstrate this variance we captured both left and right eyelids of three subjects. We show a selection of wrinkle reconstructions in Figure 13, which includes both single and double wrinkles of varying intensity. The shape of the eyelid does not just differ due to wrinkling but also depends on the underlying eyeball, as can be seen in Figure 13.e, where the corneal bulge of the eye is visible on the eyelid, even though the eye is fully closed. We further demonstrate the variation of eyelids within the same person in Figure 15. Notice for both actors B and C that one eyelid has two wrinkles while the other has only one. Additionally we illustrate in Figure 15 that our eyelid reconstructions naturally complement high-resolution facial capture methods, as the eyelids fit seamlessly into the face, increasing the reconstruction fidelity.

In addition to the intricate shape details of static eyelids, eyelid wrinkles also exhibit strong variation in their temporal formation. Figure 14 shows how a wrinkle is formed over time. During wrinkling, skin is folded in a rolling manner, which can be best seen in the accompanying video.

The accurate location where wrinkles form is essential for faithful reproduction. Figure 16 shows an overlay of the eyelid onto the input image and demonstrates how well the formed wrinkles coincide with the captured data.

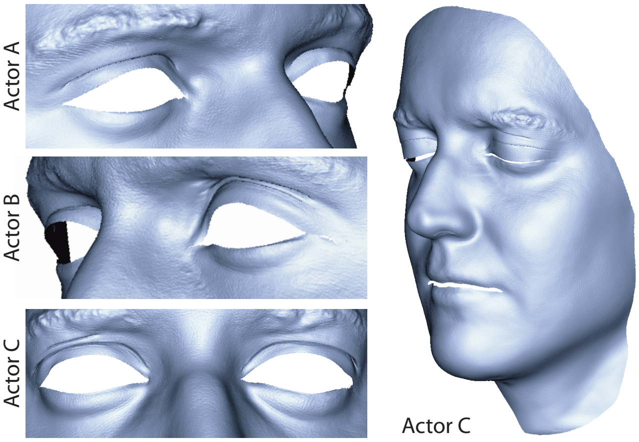


Figure 15: Left: *Eyelids can vary in shape quite substantially between people, as well as within the same person - notice the double wrinkles in one eye and single wrinkle in the other for both actor B and C.* Right: *Our reconstructed eyelids blend seamlessly with high-resolution captured faces.*

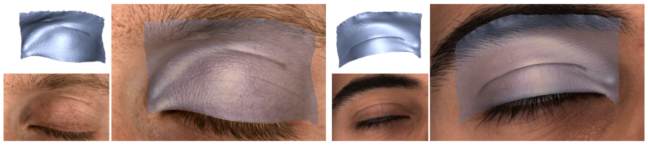


Figure 16: *We demonstrate the accuracy of our reconstructed eyelids by overlaying the mesh on an input image. The alignment of the wrinkles indicates the accuracy of the results.*

As a last example, we demonstrate how the captured eyelids may be used in the creation of a digital double for an actor. The result of our system is combined with the eyes of the actor provided by Bérard et al. [2014] and we manually complete the model by sculpting the interface between the eyelid and eye as well as adding eyelashes and eyebrows. The renders shown in Figure 17 were created using Renderman with built-in shaders.

Our experiments were run on a Windows $i7$ machine with 32GB RAM and input images of 1176×864, an eyelid template mesh of approximately 12,000 vertices and a face mesh of roughly 1 million vertices. For this setting, our average computation times per frame were 24.7 seconds per camera for the wrinkle probability map creation, 26.8 seconds per camera for flow correction, 5.2 seconds per eye for contour tracking, 11.3 seconds per eyelid for the deforma-

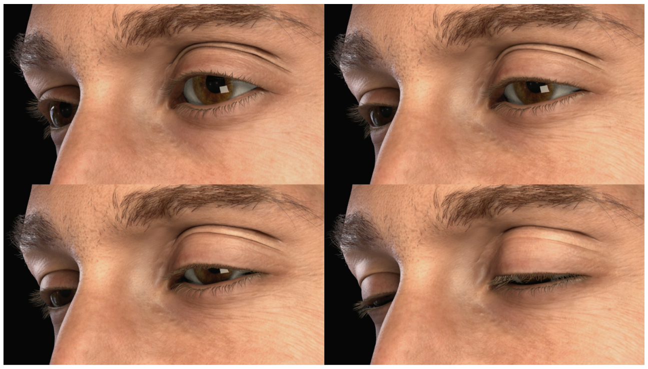


Figure 17: *Our eyelid reconstructions can be used to make high-fidelity digital doubles. Here we sculpted the thin interface to the eye, and added eyelashes, eyebrows, and eyes.*

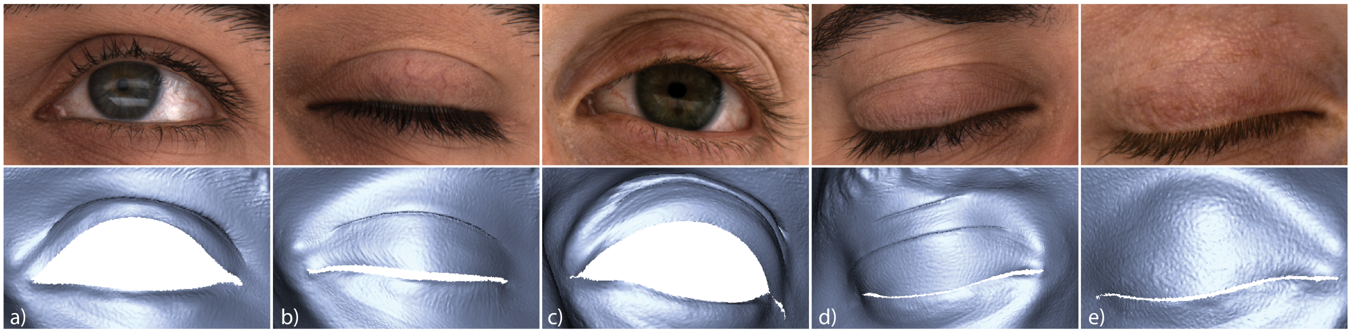


Figure 13: *We are able to reconstruct complex eyelids including a) thick wrinkles, b) thin wrinkles, c) double wrinkles close together, d) multiple distant wrinkles, and e) as an eye closes and wrinkles disappear completely, notice the subtle bulge on the lid caused by the cornea.*

tion and 24.5 seconds per eyelid for the integration step.

7 Conclusion

We have presented the first method for detailed spatio-temporal reconstruction of eyelids. Our approach combines a geometric deformation model with image data, leveraging multi-view stereo, optical flow, contour tracking and wrinkle detection from local skin appearance. Our results demonstrate that the model is able to provide a high-resolution mesh that deforms over time, reflecting detailed dynamic skin features and plausible deformations even for regions that are occluded or undergo extreme deformations. As the eye region is essential for conveying emotions, we believe that our method is an important step towards capturing expressive facial performances and the creation of realistic digital doubles.

Limitations and Future Work. Currently, our pipeline is not fully automatic and relies on a few manual steps, such as initializing the contour tracker with a few hand-drawn contours, and specifying the principal direction of the wrinkles when creating the eyelid mesh. While these manual steps can be done in a few minutes and do not require artistic skills, we plan a fully automatic pipeline for the future. By design, we can only reconstruct wrinkles that are identified by the wrinkle probability map, which in turn depends on the underlying image quality. Low resolution, motion blur or low contrast can cause detection to fail and a more sophisticated means of computing and extracting the wrinkles would be required.

Some expressions such as extreme grinning or squinting can cause wrinkle formations that our method does not handle well. For example, wrinkles in the radial direction may be filtered out by our wrinkle detection scheme (Figure 18.a). Figure 18.b depicts a case in which the skin under the wrinkle is compressed and bulges outwards rather than inwards, contradicting the assumptions of our model. In the future, we would like to extend our model to handle such cases. The ability to separate wrinkles depends on resolution, both of the wrinkle map and the proxy geometry. Figure 18.c demonstrates how very close wrinkles may be incorrectly merged if insufficient resolution is used.

Furthermore, as we compute several data terms, such as the eyelid contours, relative to the front camera, we can only handle minor head rotations. While this is sufficient for many capture scenarios, such as helmet cameras, extending the method to allow for large head rotations could be an interesting avenue for future research. Finally, our system so far focuses on performance capture and replay. For future work, an interesting avenue would be to add animation control, use our data to automatically create convincing eyelid rigs, and investigate performance and detail transfer of the eye region to virtual characters different than the actor, thereby bringing the expressiveness of virtual characters to a new level.

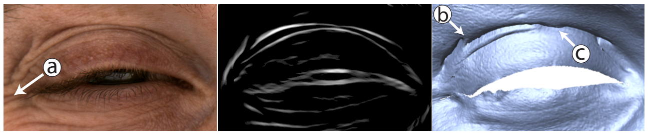


Figure 18: *A challenging grinning expression (left), the corresponding wrinkle map (center) and the reconstructed geometry (right). This expression pushes the limits of our method as radial wrinkles (a) are filtered out during the wrinkle map extraction, a skin crease is incorrectly modeled as an eyelid wrinkle (b) and very close wrinkles are merged during the geometry reconstruction (c).*

Acknowledgements

We wish to thank our performers, Maya Sigron, Kaan Yücer, and Pascal Bérard. We would also like to thank Maurizio Nitti and Alessia Marra for designing the eyelashes and eyebrows and their help with rendering, as well as Pascal Bérard again for helping with the supplementary video.

References

Alexander, O., Fyffe, G., Busch, J., Yu, X., Ichikari, R., Jones, A., Debevec, P., Jimenez, J., Danvoye, E., Antionazzi, B., et al. 2013. Digital ira: creating a real-time photoreal digital actor. In *ACM SIGGRAPH 2013 Posters*.

Bando, Y., Kuratate, T., and Nishita, T. 2002. A simple method for modeling wrinkles on human skin. In *Proc. Pacific Graphics*.

Beeler, T., and Bradley, D. 2014. Rigid stabilization of facial expressions. *ACM Trans. Graph. 33*, 4 (July), 44:1–44:9.

Beeler, T., Bickel, B., Sumner, R., Beardsley, P., and Gross, M. 2010. High-quality single-shot capture of facial geometry. *ACM Trans. Graphics (Proc. SIGGRAPH)*.

Beeler, T., Hahn, F., Bradley, D., Bickel, B., Beardsley, P., Gotsman, C., Sumner, R. W., and Gross, M. 2011. High-quality passive facial performance capture using anchor frames. *ACM Trans. Graphics (Proc. SIGGRAPH) 30*, 75:1–75:10.

Beeler, T., Bickel, B., Noris, G., Beardsley, P., Marschner, S., Sumner, R. W., and Gross, M. 2012.

Coupled 3d reconstruction of sparse facial hair and skin. *ACM Trans. Graph. 31*, 4 (July), 117:1–117:10.

Bérard, P., Bradley, D., Nitti, M., Beeler, T., and Gross, M. 2014. High-quality capture of eyes. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 33*, 6.

Bermano, A. H., Bradley, D., Beeler, T., Zund, F., Nowrouzezahrai, D., Baran, I., Sorkine-Hornung, O., Pfister, H., Sumner, R. W., Bickel, B., and Gross, M. 2014. Facial performance enhancement using dynamic shape space analysis. *ACM Trans. Graphics 33*, 2.

Bickel, B., Botsch, M., Angst, R., Matusik, W., Otaduy, M., Pfister, H., and Gross, M. 2007. Multi-scale capture of facial geometry and motion. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 33.

Bickel, B., Lang, M., Botsch, M., Otaduy, M. A., and Gross, M. 2008. Pose-space animation and transfer of facial details. In *Proc. SCA*, 57–66.

Botsch, M., and Sorkine, O. 2008. On linear variational surface deformation methods. *Visualization and Computer Graphics, IEEE Transactions on 14*, 1, 213–230.

Bouaziz, S., Wang, Y., and Pauly, M. 2013. Online modeling for realtime facial animation. *ACM Trans. Graphics (Proc. SIGGRAPH) 32*, 4, 40:1–40:10.

Bradley, D., Popa, T., Sheffer, A., Heidrich, W., and Boubekeur, T. 2008. Markerless garment capture. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 99.

Bradley, D., Heidrich, W., Popa, T., and Sheffer, A. 2010. High resolution passive facial performance capture. *ACM Trans. Graphics (Proc. SIGGRAPH) 29*, 41:1–41:10.

Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. 2004. High accuracy optical flow estimation based on a theory for warping. In *ECCV*. Springer, 25–36.

Canny, J. 1986. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 6, 679–698.

Cao, X., Wei, Y., Wen, F., and Sun, J. 2012. Face alignment by explicit shape regression. In *IEEE CVPR*, 2887–2894.

Cao, C., Weng, Y., Lin, S., and Zhou, K. 2013. 3d shape regression for real-time facial animation. *ACM Trans. Graphics (Proc. SIGGRAPH) 32*, 4, 41:1–41:10.

Cao, C., Hou, Q., and Zhou, K. 2014. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Trans. Graphics (Proc. SIGGRAPH) 33*, 4, 43:1–43:10.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. 2001. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence 23*, 6, 681–685.

Deng, Z., Lewis, J., and Neumann, U. 2005. Automated eye motion using texture synthesis. *CGA 25*, 2.

Dutreve, L., Meyer, A., and Bouakaz, S. 2011. Easy acquisition and real-time animation of facial wrinkles. *Comput. Animat. Virtual Worlds 22*, 2-3, 169–176.

Feng, W.-W., Yu, Y., and Kim, B.-U. 2010. A deformation transformer for real-time cloth animation. *ACM Trans. Graphics (Proc. SIGGRAPH) 29*, 4, 108:1–108:9.

François, G., Gautron, P., Breton, G., and Bouatouch, K. 2009. Image-based modeling of the human eye. *IEEE TVCG 15*, 5, 815–827.

Garrido, P., Valgaerts, L., Wu, C., and Theobalt, C. 2013. Reconstructing detailed dynamic face geometry from monocular video. In *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, vol. 32, 158:1–158:10.

Ghosh, A., Fyffe, G., Tunwattanapong, B., Busch, J., Yu, X., and Debevec, P. 2011. Multiview face capture using polarized spherical gradient illumination. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 30*, 6, 129:1–129:10.

Huang, H., Chai, J., Tong, X., and Wu, H.-T. 2011. Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition. *ACM Trans. Graphics (Proc. SIGGRAPH) 30*, 4, 74:1–74:10.

Kähler, K., Haber, J., Yamauchi, H., and Seidel, H.-P. 2002. Head shop: Generating animated head models with anatomical structure. In *Proc. SCA*, 55–63.

Kim, D., Koh, W., Narain, R., Fatahalian, K., Treuille, A., and O'Brien, J. F. 2013. Near-exhaustive precomputation of secondary cloth effects. *ACM Trans. Graphics (Proc. SIGGRAPH) 32*, 4, 87:1–87:8.

Klaudiny, M., and Hilton, A. 2012. High-detail 3d capture and non-sequential alignment of facial performance. In *3DIM-PVT*.

Larboulette, C., and Cani, M.-P. 2004. Real-time dynamic wrinkles. In *Proc. Computer Graphics Int.*, 522–525.

Le, B. H., Ma, X., and Deng, Z. 2012. Live speech driven head-and-eye motion generators. *IEEE TVCG 18*, 11, 1902–1914.

Lee, S. P., Badler, J. B., and Badler, N. I. 2002. Eyes alive. *ACM Trans. Graphics (Proc. SIGGRAPH) 21*, 3, 637–644.

Li, H., Yu, J., Ye, Y., and Bregler, C. 2013. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graphics (Proc. SIGGRAPH) 32*, 4, 42:1–42:10.

Li, J., Xu, W., Cheng, Z., Xu, K., and Klein, R. 2015. Lightweight wrinkle synthesis for 3d facial modeling and animation. *Computer-Aided Design 58*, 0, 117–122.

Ma, W.-C., Hawkins, T., Peers, P., Chabert, C.-F., Weiss, M., and Debevec, P. 2007. Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. In *Eurographics Symposium on Rendering*, 183–194.

Ma, W.-C., Jones, A., Chiang, J.-Y., Hawkins, T., Frederiksen, S., Peers, P., Vukovic, M., Ouhyoung, M., and Debevec, P. 2008. Facial performance synthesis using deformation-driven polynomial displacement maps. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 27*, 5, 121.

Magnenat-Thalmann, N., Kalra, P., Luc Leveque, J., Bazin, R., Batisse, D., and Querleux, B. 2002. A computational skin model: Fold and wrinkle formation. *Trans. Info. Tech. Biomed. 6*, 4, 317–323.

Müller, M., and Chentanez, N. 2010. Wrinkle meshes. In *Proc. SCA*, 85–92.

Perona, P., and Malik, J. 1990. Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 12*, 7 (Jul), 629–639.

PETERSON, M. F., AND ECKSTEIN, M. P. 2012. Looking just below the eyes is optimal across face recognition tasks. *Proceedings of the National Academy of Sciences 109*, 48.

POPA, T., ZHOU, Q., BRADLEY, D., KRAEVOY, V., FU, H., SHEFFER, A., AND HEIDRICH, W. 2009. Wrinkling captured garments using space-time data-driven deformation. *Computer Graphics Forum (Proc. Eurographics) 28*, 2, 427–435.

RHEE, T., HWANG, Y., KIM, J. D., AND KIM, C. 2011. Real-time facial animation from live video tracking. In *Proc. SCA*.

ROHMER, D., POPA, T., CANI, M.-P., HAHMANN, S., AND SHEFFER, A. 2010. Animation wrinkling: Augmenting coarse cloth simulations with realistic-looking wrinkles. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 29*, 6, 157:1–157:8.

RUHLAND, K., ANDRIST, S., BADLER, J., PETERS, C., BADLER, N., GLEICHER, M., MUTLU, B., AND MCDONNELL, R. 2014. 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*, 69–91.

SAGAR, M. A., BULLIVANT, D., MALLINSON, G. D., AND HUNTER, P. J. 1994. A virtual environment and model of the eye for surgical simulation. In *Proceedings of Computer Graphics and Interactive Techniques*, 205–212.

SEILER, M., SPILLMANN, J., AND HARDERS, M. 2012. Enriching coarse interactive elastic objects with high-resolution data-driven deformations. In *Proc. SCA*, 9–17.

SHI, F., WU, H.-T., TONG, X., AND CHAI, J. 2014. Automatic acquisition of high-fidelity facial performances using monocular videos. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 33*.

SMITH, M. L., COTTRELL, G. W., GOSSELIN, F., AND SCHYNS, P. G. 2005. Transmitting and decoding facial expressions. *Psychological Science 16*, 3, 184–189.

SORKINE, O., AND ALEXA, M. 2007. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, vol. 4.

SUWAJANAKORN, S., KEMELMACHER-SHLIZERMAN, I., AND SEITZ, S. M. 2014. Total moving face reconstruction. In *ECCV*.

TRUTOIU, L. C., CARTER, E. J., MATTHEWS, I., AND HODGINS, J. K. 2011. Modeling and animating eye blinks. *ACM Trans. Appl. Percept. 8*, 3.

VALGAERTS, L., WU, C., BRUHN, A., SEIDEL, H.-P., AND THEOBALT, C. 2012. Lightweight binocular facial performance capture under uncontrolled lighting. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 31*, 6.

WANG, H., HECHT, F., RAMAMOORTHI, R., AND O'BRIEN, J. F. 2010. Example-based wrinkle synthesis for clothing animation. *ACM Trans. Graphics (Proc. SIGGRAPH) 29*, 4, 107:1–107:8.

WARBURTON, M., AND MADDOCK, S. 2014. Physically-based forehead animation including wrinkles. *Comput. Animat. Virtual Worlds*.

WEISE, T., BOUAZIZ, S., LI, H., AND PAULY, M. 2011. Real-time performance-based facial animation. *ACM Trans. Graphics (Proc. SIGGRAPH) 30*, 4, 77:1–77:10.

WEISSENFELD, A., LIU, K., AND OSTERMANN, J. 2010. Video-realistic image-based eye animation via statistically driven state machines. *The Visual Computer 26*, 9, 1201–1216.

WU, C., VARANASI, K., LIU, Y., SEIDEL, H.-P., AND THEOBALT, C. 2011. Shading-based dynamic shape refinement from multi-view video under general illumination. In *ICCV*.

ZHANG, Y., SIM, T., AND TAN, C. 2005. Simulating wrinkles in facial expressions on an anatomy-based face. In *Proc. ICCS*, 207–215.

ZURDO, J. S., BRITO, J. P., AND OTADUY, M. A. 2013. Animating wrinkles by example on non-skinned cloth. *IEEE TVCG 19*, 1, 149–158.