

Interactive Generation of Realistic Facial Wrinkles from Sketchy Drawings

Hyeon-Joong Kim^{1,3}, A. Cengiz Öztireli², Il-Kyu Shin¹, Markus Gross², Soo-Mi Choi^{†1}

¹ Sejong University, Korea ² ETH Zurich, Switzerland ³ 3D Systems, Korea

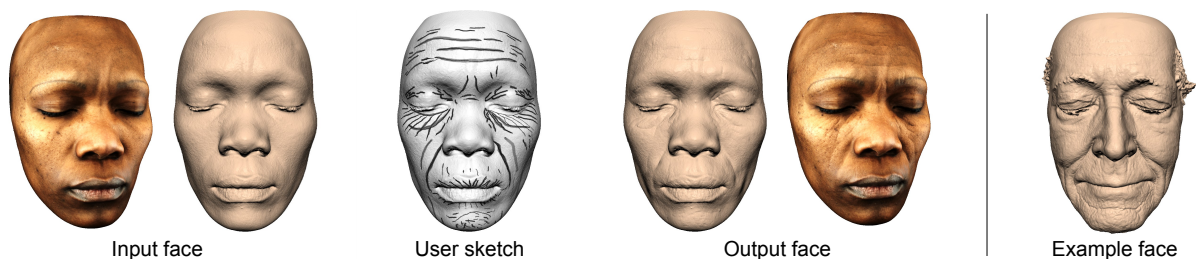


Figure 1: We use statistics extracted from example faces to augment interactively drawn concept sketches for synthesizing realistic facial wrinkles.

Abstract

Synthesizing facial wrinkles has been tackled either by a long process of manual sculpting on 3D models, or using automatic methods that do not allow for user interaction or artistic expression. In this paper, we propose a method that accepts interactive sketchy drawings depicting wrinkle patterns, and synthesizes realistic looking wrinkles on faces. The method inherits the simplicity of sketching, making it possible for artists as well as novice users to generate realistic facial detail very efficiently, allowing fast preview for physical makeup, or aging simulations for fun and professional applications. All strokes are used to infer the wrinkles, retaining the expressiveness of the sketches and realism of the final result at the same time. This is achieved by designing novel multi-scale statistics tailored to the wrinkle geometry and coupled to the sketch interpretation method. The statistics capture the cross-sectional profiles of wrinkles at different scales and parts of a face. The strokes are augmented with the statistics extracted from given example face models, and applied to an input face model interactively. The interface gives the user control over the shapes and scales of wrinkles via sketching while adding extra details required for realism automatically.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Shape Modeling—Realistic Face Modeling

1. Introduction

Modeling human faces has been the focus of many works in various fields. In graphics, it is particularly important to model small scale details on face models for a wide range of applications. Understanding and synthesizing details such as wrinkles on faces is a very challenging task

due to the extreme sensitivity of the human perception for human faces as well as the complex physiological, anatomical, and environmental factors involved in the process of creation of such facial detail [SEL06, FGH10]. When modeling aging faces, these factors along with subjective considerations about the history and character of the face should be taken into account [Deb09]. Hence, synthesizing realistic facial details requires an interactive approach where the user is free to express her/his view of the facial features. How-

[†] Corresponding author (smchoi@sejong.ac.kr)

ever, most methods in graphics have focused on synthesizing or transferring such details automatically based on 2D pictures or 3D face scans [GMP*06, BBA*07, FGH10, LLLC11, GTB*13].

So far, the options for interactive creation of facial details such as wrinkles have been either going through a tedious process of silicon based physical makeup [Deb09], or using a generic virtual sculpting tool. Both of these solutions, however, require considerable effort and several iterations to achieve a realistic result. In contrast, sketching is a notably effective way to describe wrinkle patterns, and is widely used as the initial step for generating such details in many fields such as forensic art and special makeup communities [Man10, Deb09]. In films and television productions, special makeup artists begin by drawing sketches to design aging makeup looks for a specific face depending on the character being portrayed [Deb09]. In forensic science, sketch artists interview witnesses and victims of crimes and create sketches that match their descriptions, or draw sketches of missing persons, showing how they might have aged [Man10]. The sketches drawn for these purposes often only depict an abstraction of the modeled face, and are not suitable for the generation of realistic looking face models with the existing tools. In general, it is a highly challenging and ill-posed problem to infer meaningful information from such sketchy drawings [NSS*12]. It is even more difficult to synthesize wrinkles given such drawings, and methods so far rely on extracted clean curves from images and predefined functions [LLC11].

In this paper, we present a method that allows the generation of realistic looking wrinkles on 3D face models with interactive sketchy drawings. Our tool provides an intuitive and efficient way for fast and vivid preview of wrinkle patterns implied by such drawings, and hence can be used as the initial planning step for professional applications such as physical or virtual aging makeup, as well as for generating aged looks of people for entertainment by novice users. The method allows the user to sketch freely, and utilizes all the information in the sketch to create a set of wrinkles on an input face model with desired detail characteristics extracted from example models. This is made possible by designing multi-scale statistical measures tailored to the facial wrinkle geometry that capture the cross-sectional profiles of wrinkles at different scales and parts of a face. The statistics are extracted from given face model(s) with desired details. Once they are computed, the user can interactively draw strokes with varying thickness and alter an input face model. The algorithm automatically augments the strokes with the computed statistics to generate realistic looking wrinkles from any complex drawings.

2. Related Work

Our work relates to wrinkle and detail generation for faces, especially with adult aging [FGH10], and sketch-based and interactive modeling.

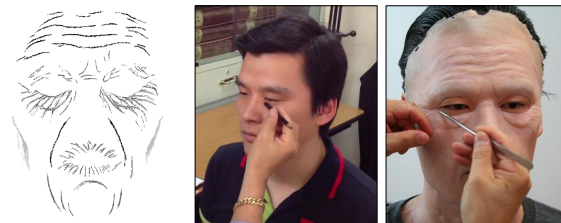


Figure 2: (Left) A concept sketch drawn by a professional makeup artist. The locations of strokes indicate center-lines of wrinkles and thickness indicates scale of deformation due to wrinkles. This sketch is used for planning for the physical makeup process (right).

Physically based facial wrinkle generation Wrinkles are closely related to the physical layers such as the elastic fibers in dermis and facial muscles [SEL06, INN07, Deb09]. Hence, several methods have been proposed to infer wrinkles and other face and expression specific deformations from simplified physical models of the facial structures. Some of these aim at accurate reconstruction of age related [BKTK00, MTKLL*02] general [LK14] skin wrinkles, while others target animating expression specific details on faces [TW90, WMTT94, Wat87, ZPS01, ZPS04, ZSTS06]. Although physically accurate models result in realistic wrinkling, these methods do not take factors beyond the facial structure into account. Furthermore, it is difficult to shape or edit the generated wrinkles.

Geometric wrinkle generation Another approach for constructing wrinkles is defining functions or energies that govern the wrinkle shapes, and controlling these by a few parameters, nodes, or curves. Some methods depend on energies controllable by curves/parameters [VLR05, WWY06, RPC*10], while others allow more direct control over the wrinkles via sketched or extracted curves along with functions that define the wrinkle shapes [BKN02, LC04, ZS05, CGW*05, CGW*07, LXZ07, LLLC11]. However, it is challenging and unintuitive to estimate the parameters for users to achieve the realism and expressiveness desired for facial models [LLC11]. Instead of specifying curves and parameters, with our method, the user can draw freely to create realistic wrinkles matching her/his strokes.

Statistical and data-driven face models It has been observed that 3D morphable models can be used to accurately approximate face geometry [BV99, PKA*09]. This idea has also been utilized to simulate large scale aging effects [SSSB07] or generate 3D face models from sketches [STDN10]. Similarly, existing facial deformations can be used to learn models in order to guide new facial deformations [FKY08, MLD09]. However, in the mentioned methods, the resulting faces mostly capture large scale features and not the details important for realism. To capture and synthesize such details and wrinkles, localized statistics of example face surfaces [GMP*06], learned re-

lations of details to skin strain [BLB*08] or motion capture data [MJC*08], facial rigs with high resolution geometry [ARL*10], or high quality dynamic facial performance capture [BBB*14] can be utilized. Although these methods allow for face-aware transfer or synthesis of details, they do not offer control of the synthesized wrinkles and hence are not suitable when art-directability is required. In contrast, our novel statistical measures are carefully designed for synthesizing realistic wrinkles from complex sketchy drawings.

Sketch-based modeling Sketching is a very intuitive way of modeling objects from scratch or deforming or augmenting them with additional structures [OSSJ09]. Hence, many methods have been developed to create 3D models via strokes [AHKS94, ZHH06, IMT99, GIZ09], or to modify and augment models according to the strokes [CGW*05, OSSJ05, ZHS*05, NISA07, KSvdP09, Pix13]. However, these methods are designed for general modeling, and do not allow for easy and efficient generation of facial wrinkle patterns. The mentioned methods usually require clean sketches with clear curves and hence a sketch cleaning step is needed to extract relevant strokes [OSSJ09]. However, it is a difficult and ill-posed problem to detect implied curves or other semantic information from a complex real world sketchy drawing [NSS*12]. Consequently, all sketching systems for wrinkles so far assume simple curves are drawn to define the locations of the wrinkles [CGW*05, WWY06, CGW*07]. Even under the assumption of clean sketches with simple curves, generating wrinkle patterns with realistic details is challenging for the user [LLLC11]. In contrast, our method accepts sketchy drawings and still generates realistic looking wrinkles by utilizing specialized statistics. An example input sketch can be seen in Figure 1.

3. Overview

Our aim is to allow the user to freely produce a sketchy drawing interactively, and augment it with statistics extracted from given example face model(s) to produce realistic looking wrinkles on an input face automatically in real time. In order to achieve this, we first make the observation that there are essentially two types of information present in sketches that depict wrinkles: a stroke marks the *center-line* of a wrinkle, and the thickness of the stroke is used to indicate the *scale of deformation* due to the drawn wrinkle. This is illustrated for a sketch drawn by a professional artist for planning aging makeup in Figure 2. Following this observation, we design statistics that capture the shape of deformation around center-lines (furrows) of wrinkles at multiple scales in Section 4. These statistics extend the commonly used directionality based facial detail statistics in the medical sciences [ZH06] and previous works for transferring facial detail in graphics [GMP*06]. We present methods to compute these statistics from given example face models. The statistics then allow us to map strokes to wrinkle center-lines, and

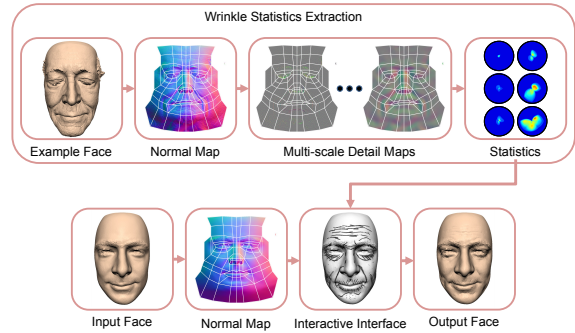


Figure 3: Overview of our method. After wrinkle statistics are extracted from an example face in an offline step, the normal map of an input model is modulated interactively with user strokes augmented with the statistics to generate the output face model.

thickness of strokes to deformations at different scales in Section 5. The main steps of our technique are illustrated in Figure 3.

4. Wrinkle Statistics

In this section, we develop statistics that measure how the facial surface deforms around the center-lines at multiple scales, for representing region dependent details due to wrinkles on a given face.

4.1. Parametrization and Alignment

For efficient and robust processing of facial geometry, we work with maps that correspond to a consistent parametrization of faces. Hence, we first parametrize the input model I , and the example model E over an ellipsoid (Figure 4 (b)), by picking a few predefined points on each face for the alignment of the face with the ellipsoidal model. Then, normal maps \mathbf{n} represented as images over this domain are computed (Figure 4 (c)). Next, a generic face template is laid down on the generated normal maps. The user can then adjust some vertices of the face template if they do not align with the facial features. The aligned face template assigns structurally different regions to different patches in accordance with the facial anatomy to group similar wrinkle patterns together, and also provides a natural patch-wise correspondence between faces (Figure 4 (d)). We mask out the patches that correspond to the eye and nose regions, which are not relevant for wrinkle synthesis (they are depicted as the black patches in Figure 4 (d)). Note that a precise alignment is not needed for our purposes, since we average quantities on patches to compute patch-wise statistics, which are then interpolated throughout the face (please also see Figure 18). In practice, only a few adjustments of the face template vertices were needed, making the overall interaction time negligible. This simple alignment strategy was thus sufficient for our purposes.

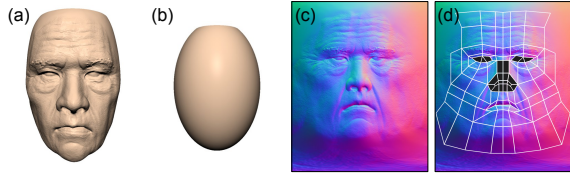


Figure 4: From left to right: A model, the parametrization domain, normal map image, alignment with the face template.

4.2. Multi-scale Facial Detail Representation

It has been observed [GMP*06] that details due to aging can be captured by computing an approximation of an anti-aged version of a facial model by a smoothing method, and using a notion of difference between the original and smoothed facial geometry. We utilize this observation as a starting point of our facial detail representation. Instead of point locations, we work with normal maps that are well suited to represent geometric details and transfer them without increasing the geometric complexity of the output model. To capture wrinkles at different detail levels, multi-scale detail maps are computed by generating normal maps of smoothed versions of the example face and subtracting them at consecutive scales. Although smoothing does not necessarily generate the actual anti-aged version of a face, the detail maps still accurately capture wrinkle shapes sufficient to produce statistical measures for realistic synthesis.

We thus first construct smoothed versions \mathbf{n}_s of the normal map \mathbf{n} of E for $s = 1 \dots m$ by convolving the parametrized E with a Gaussian filter of increasing standard deviations σ_s (Figure 5, first row). To isolate the details at each scale, we then compute the difference between the normal maps \mathbf{n}_s and rotate the difference such that the normals from the smoother map are aligned with the z axis:

$$\Delta \mathbf{n}_s(i, j) = \mathbf{R}_s(i, j) (\mathbf{n}_{s-1}(i, j) - \mathbf{n}_s(i, j)), \quad (1)$$

where $\mathbf{n}_s(i, j)$ represents the normal vector stored at the location (i, j) on the normal map \mathbf{n}_s , $\mathbf{R}_s(i, j)$ is the rotation matrix that aligns the normal $\mathbf{n}_s(i, j)$ with the z axis, and $\mathbf{n}_0 = \mathbf{n}$ (Figure 5, second row). Finally, images \mathbf{v}_s that contain the two-dimensional vector of the normalized projection of $\Delta \mathbf{n}_s(i, j)$ onto the xy plane, and further images Θ_s that store the angle between $\mathbf{n}_s(i, j)$ and $\mathbf{n}_{s-1}(i, j)$ are computed. If the projection is zero, a vector of zero length is stored. For the rest of the paper, we will use the continuous notation $\Theta_s(\mathbf{x})$, $\mathbf{v}_s(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^2$.

We show these steps and the resulting images in Figure 5. The maps $\Delta \mathbf{n}_s$ capture details of certain frequencies in the normal map \mathbf{n} , which essentially contain the changes due to wrinkles. The angle $\Theta_s(\mathbf{x})$ gives us how much the normal at \mathbf{x} is rotated at scale s and hence how the facial surface is altered due to the wrinkles at that scale. $\Theta_s(\mathbf{x})$ is small in areas not affected by the wrinkles and along the center-lines of the wrinkles, as can be observed in the images on the

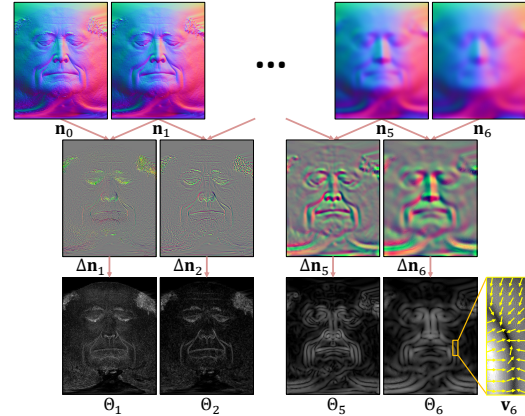


Figure 5: Normal maps of smoothed versions of the example face (top row) are subtracted and rotated to get $\Delta \mathbf{n}_s$ (middle row), which represent details at different scales. These are then used to compute the derived maps Θ_s that contain the angles between the normals at different scales (bottom row), and 2D vector fields \mathbf{v}_s that point towards the center-lines of the wrinkles (bottom row, rightmost, some of the vectors are plotted with contrast-enhanced Θ_s in the background).

bottom row of Figure 5. This results from the shape of the wrinkles with a narrow flat valley along the center-lines and bulges around the center-lines [BKN02, LLC11]. Hence, the maps Θ_s provide a representation of the cross-sectional profiles of the wrinkles. The vector $\mathbf{v}_s(\mathbf{x})$ gives the rotated direction of the normal difference at \mathbf{x} . This vector always points towards the center-lines of the wrinkles, as illustrated for a small patch in the rightmost figure on the bottom row of Figure 5. This is again a result of the characteristic shape of wrinkles, where the normals change along lines orthogonal to the wrinkle center-lines [BKN02, LLC11]. We will use these maps to extract center-lines and boundaries, and trace the angles along cross-sectional profiles in the next section.

4.3. Wrinkle Shape Function

As discussed in the last section and illustrated in Figure 5, the vectors in \mathbf{v}_s tend to be oriented orthogonal to the center-lines of wrinkles, and the angles in Θ_s capture the deformations along the cross sections. We will utilize these properties to compute a function that describes cross-sectional profiles of wrinkles. In contrast to previous works that compute parametric profile shapes [BKN02, ZS05, LXZ07, LLC11], this construction allows us to capture realistic data-driven wrinkle shapes localized to different regions of a face from given example faces.

Specifically, we would like to estimate the function $\Theta_s(\mathbf{x}, \mathbf{d}, r) : \mathbb{R}^2 \times \mathbb{S}^2 \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ defined on the domain of the normal maps, where \mathbf{x} is a point on the center-line of a wrinkle, $\mathbf{d} \in \mathbb{S}^2$ is a unit vector orthogonal to the local wrinkle segment, and r is the distance to the center-line (see Figure 6 for examples). This function is computed for each

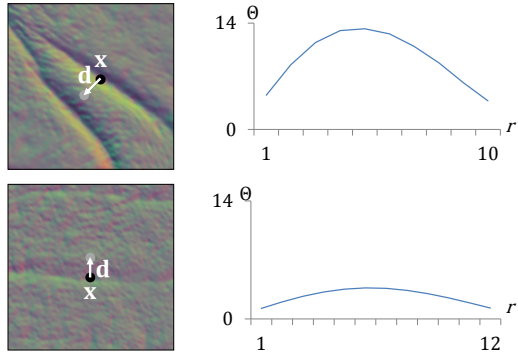


Figure 6: Wrinkle shape functions for fixed \mathbf{x} and \mathbf{d} 's.

point on the center-lines of all wrinkles. On all other points, $\theta_s(\mathbf{x}, \mathbf{d}, r)$ is defined to be 0.

Center-line and boundary extraction The first step of computing the function $\theta_s(\mathbf{x}, \mathbf{d}, r)$ is extracting wrinkle center-lines and influence boundaries. The center-line extraction algorithm starts from each pixel \mathbf{x}_0 with an angle $\Theta_s(\mathbf{x}_0) > \epsilon$ for a small constant ϵ to avoid areas without bulging due to wrinkles. It then follows the vector $\mathbf{v}_s(\mathbf{x})$ for a pixel center \mathbf{x} on the path to go to the next pixel towards the center-line of a wrinkle. This is iterated until a pixel with $\Theta_s(\mathbf{x}) < \epsilon$ or $\mathbf{v}_s(\mathbf{x})^T \mathbf{v}_s(\mathbf{x}_0) < \epsilon$ is found. At such an \mathbf{x} , the angle difference is small or $\mathbf{v}_s(\mathbf{x})$ changes its direction significantly, indicating that \mathbf{x} is on the center-line of a wrinkle. In addition to center-lines, we also extract boundaries of wrinkles, which mark the end of the influence of a wrinkle. The algorithm for their extraction is the same as above, except that it now follows the negative direction $-\mathbf{v}_s(\mathbf{x})$ to move away from the center-lines.

These algorithms are run starting from each pixel of the maps and with $\epsilon = 10^{-6}$ to avoid artifacts due to numerical precision. The resulting center-lines (red curves) and boundaries (blue curves) are plotted for different scales in Figure 7. The red curves agree with the apparent center-lines of wrinkles and the blue curves mark the end of the bulging due to wrinkles. The cross-sectional profiles of the wrinkles thus stay between these two types of curves.

Tracing along cross sections Once center-lines and boundaries are extracted, cross sections can be traced to get samples from the function $\theta_s(\mathbf{x}, \mathbf{d}, r)$. Starting from each pixel \mathbf{x} , the tracing algorithm moves along the line $\mathbf{x} + t\mathbf{v}_s(\mathbf{x})$ until a red \mathbf{x}_0 and a blue \mathbf{x}_1 pixel are hit. The line between \mathbf{x}_0 and \mathbf{x}_1 given by $\mathbf{x}_0 + t\mathbf{d}$ with $\mathbf{d} = \mathbf{x}_1 - \mathbf{x}_0 / \|\mathbf{x}_1 - \mathbf{x}_0\|$ can be traced on Θ_s to get samples for the wrinkle shape function: $\theta_s(\mathbf{x}_0, \mathbf{d}, r_i) = \Theta_s(\mathbf{x}_i)$, where $r_i = \|\mathbf{x}_i - \mathbf{x}_0\|$ for points \mathbf{x}_i (pixel centers) on the traced line. Resulting sampled functions for two examples are shown in Figure 6. In practice, we omit the lines with number of pixels larger than 100 for efficient augmentation of the strokes in the later stages. This value is sufficient to capture the deformations due to wrinkles for all scales we use.

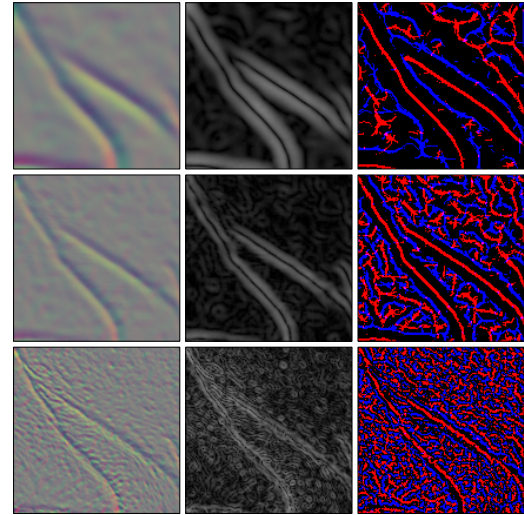


Figure 7: Example patches of normal difference maps (left column) and angle maps (middle column), and (right column) extracted wrinkle center-lines (red curves) and boundaries (blue curves) for different scales.

Discussion Previous works also represent shapes of wrinkles by developing models for their cross-sectional profiles. However, they rely on parametric models for the wrinkle shapes. The parameters are often set globally and it is difficult to control them for realistic wrinkle synthesis when augmenting complex user sketches. Example results with a state-of-the-art approach [LLLC11] are shown in Figure 8, top. Learning the wrinkle functions from data gives us the distinct advantage that the generated wrinkles always respect the facial geometry, as shown on the bottom row of the figure.

Note that our definition of the wrinkle shape function differs from the ones used in previous works in two aspects. First, the function θ_s does not depend on the distance of a point on a wrinkle center-line to the endpoints of the center-line. This dependency was included in previous works [LLLC11] to reflect the property of the real wrinkles that the deformation due to a wrinkle becomes milder as the endpoints are approached. However, in our framework, this dependency is easily and more intuitively controlled by the user by changing the thickness of strokes via the pressure of the stylus. Hence, we observed that including this in the statistics is actually counter-productive as it makes the domain of the wrinkle function higher dimensional, degrading the sampling and hence the quality of the resulting wrinkles. Second, the wrinkle functions do not cover the full bulge around the center-lines of the wrinkles, since the detected boundaries of the wrinkles are located on the peaks of the bulges as illustrated by the blue curves in Figure 7. We observed that the areas of the bulges past the peaks can be regarded as resulting from near-by wrinkles, and are indeed implied by additional strokes that depict near-by wrinkles

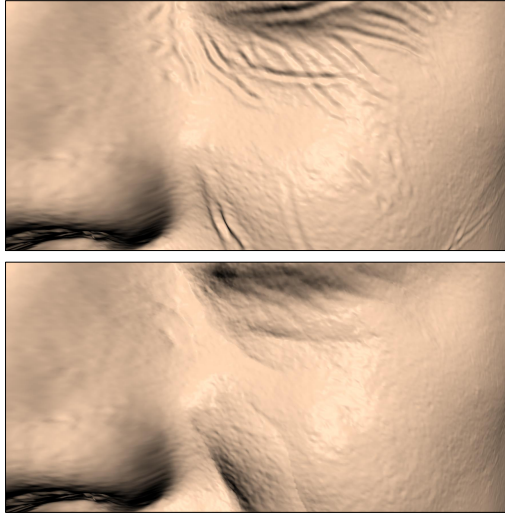


Figure 8: Existing wrinkle synthesis methods rely on pre-defined wrinkle functions to deform the facial geometry (top), resulting in coarse deformations when used with user sketches. Utilizing extracted wrinkle statistics that capture wrinkle details for different scales and parts of a face, our method can generate realistic wrinkles from the same strokes (bottom).

in a drawing, as illustrated for a simple case in Figure 10. Hence, we avoided including these areas in the statistics in order to prevent build-up of unintended deformations.

4.4. Statistics for Face Template Patches

The output of the extraction and tracing steps as explained in the last section are samples from the functions θ_s . These samples provide a very sparse sampling of the high dimensional domain of these functions. For a given point \mathbf{x} , the samples lie only along the directions orthogonal to the wrinkle center-line at that point on the example face, while a user sketch can contain strokes in any direction at any point. In general, the wrinkles on the example model E and the ones implied by the user sketch on the input model I can have different locations and orientations. In order to augment user strokes with the wrinkle shapes as stored by the samples, we thus need a way to construct statistical measures that capture the overall details due to the wrinkles in different parts of E .

We use the structural segmentation given by the face template that groups wrinkles with uniform characteristics to compute patch-wise statistics by aggregating the wrinkle shape functions $\theta_s(\mathbf{x}, \mathbf{d}, r)$. The aggregation for a patch P is performed by averaging the functions for which the locations \mathbf{x} fall into the patch P :

$$\theta_s^P(\mathbf{d}, r) = \frac{1}{|P|} \int_{\mathbf{x} \in P} \theta_s(\mathbf{x}, \mathbf{d}, r) d\mathbf{x}, \quad (2)$$

with $|P|$ denoting the area of the patch P . In practice, we

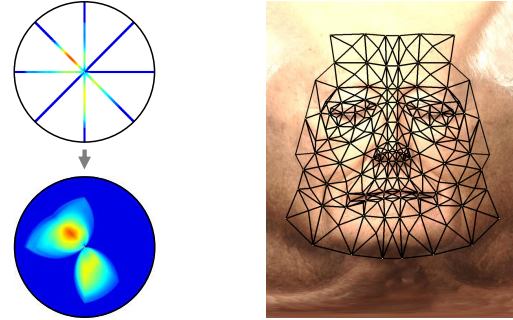


Figure 9: (Left) Patch-wise statistics computed as functions in eight directions are interpolated in polar coordinates to get the final statistics for each patch. (Right) For any point on the image plane, patch-wise statistics are interpolated on an extended dual graph of the face template to avoid discontinuities in the synthesized wrinkles.

approximate the functions $\theta_s(\mathbf{x}, \mathbf{d}, r)$ and the integral numerically by using the computed samples. For each sample $\theta_s(\mathbf{x}_i, \mathbf{d}_i, r_i)$ with $\mathbf{x}_i \in P$, the direction \mathbf{d}_i is snapped to the nearest direction among the eight directions that partition the unit circle to further reduce the sparsity of the sampling. Then, the samples are averaged to approximate the integral in Equation 2. For a given patch, the result is a function on the lines in the eight directions, as illustrated in Figure 9, top-left. Then, we interpolate these values by a linear interpolation in the polar coordinates to get the final estimation of $\theta_s^P(\mathbf{d}, r)$ (Figure 9, bottom-left). Statistics for some patches of the face template are shown in Figure 11. As illustrated, $\theta_s^P(\mathbf{d}, r)$ accurately captures profiles of wrinkles in different regions of a face at multiple scales.

In order to avoid discontinuities around the patch boundaries, the statistics θ_s^P are bi-linearly interpolated on an extension of the dual graph of the face template at runtime, as illustrated in Figure 9, right. We denote the interpolated statistics at a point \mathbf{x} with $\theta_s^{P(\mathbf{x})}(\mathbf{d}, r)$.

5. Interpreting and Augmenting User Sketches

Once the statistics are extracted from the example model E , the interface becomes ready to accept user strokes. We also record the pressure values from the stylus and use them to determine the thickness of the strokes. We do not make any assumptions on how sketchy the drawings can be as attempt-



Figure 10: Deformations due to single strokes (left and middle) combine to form a bulge (right) when the strokes are drawn together.

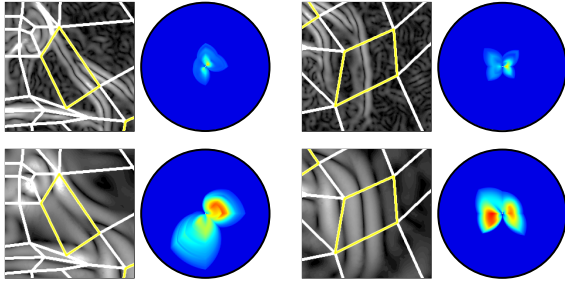


Figure 11: Statistics for two different face template patches at two different scales. Contrast-enhanced Θ_s are shown in the background.

ing to clean the sketches can reduce the degree of expressiveness and skew user intention. Instead, the statistics derived in the previous section are used to ensure realistic output wrinkles from the set of the strokes at each step.

As detailed in Section 4, the statistics for each patch encode the average cross-sectional profile of the wrinkles in that patch for different scales, directions, and distances from the center-lines. In user sketches, each stroke mark the center-line of a wrinkle, while the thickness of the stroke indicates the scale of the expected deformation around the center-line. By mapping the local orientation and thickness of a stroke to the direction and scale in the statistics, it is possible to modulate the surface normals of the input face according to the statistics. Below we explain how the strokes are represented and these mappings are computed.

5.1. Stroke Definition

Each stroke drawn via the interactive interface is projected onto the input model I , and the corresponding points on the normal map are then computed. This defines a stroke in the two-dimensional domain of the normal maps. We denote the resulting stroke with $h(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^2$ denoting a pixel center. In addition to the locations of the points along the stroke, we also get a pressure value for each point. The pressure value for a point \mathbf{x} is linearly mapped to an integer $s(\mathbf{x})$ in the range $\{1, m\}$ (m is the number of scales, see Section 4.2). Hence, for each point on h , we obtain the corresponding scale implied by the stroke at that point. Finally, the stroke is segmented according to the computed scales such that we get a set of m segments $h_s(\mathbf{x}) = \{\mathbf{x} | s(\mathbf{x}) = s\}$, as illustrated in Figure 12, top row. Each segment h_s thus represents a part of the stroke that should be mapped to a wrinkle at scale s .

5.2. Augmenting Strokes with Statistics

The goal of stroke augmentation is determining how a surface normal $\mathbf{n}(\mathbf{y})$ at a point \mathbf{y} on the normal map of the input model I will change due to the given stroke h . Each segment h_s of the stroke implies a change to $\mathbf{n}(\mathbf{y})$ at a different scale. Hence, for each segment h_s , the statistics at scale s are used

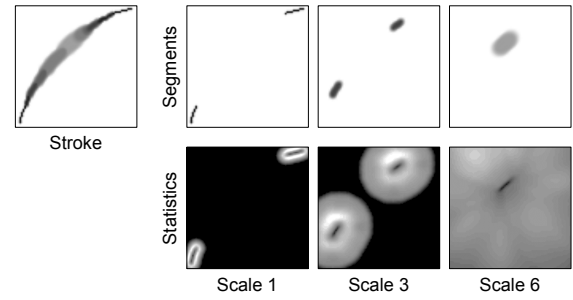


Figure 12: A user stroke (leftmost) is segmented into different parts based on local thickness (top row). Each segment is then augmented with the statistics of the corresponding scale (bottom row).

to compute the change in $\mathbf{n}(\mathbf{y})$ due to that segment. These are then averaged to get the final change.

Approximating distance and direction In order to augment h_s with the statistics, we need to approximate the direction $\mathbf{d}(\mathbf{y})$ and distance $r(\mathbf{y})$ of \mathbf{y} to h_s , which are then used to fetch the value at $\theta_s^{P(\mathbf{y})}(\mathbf{d}(\mathbf{y}), r(\mathbf{y}))$. The direction and distance can be estimated with various fitting or smoothing methods. We found out that a simple estimation that averages on the locations of points on h_s is sufficient for our purposes. Specifically, the direction and distance are given by: $\mathbf{d}(\mathbf{y}) = \sum_{\mathbf{x}_i \in h_s} \mathbf{d}_i(\mathbf{y}) w_i(\mathbf{y})$, $r(\mathbf{y}) = \sum_{\mathbf{x}_i \in h_s} r_i(\mathbf{y}) w_i(\mathbf{y})$ with $\mathbf{d}_i(\mathbf{y}) = \mathbf{y} - \mathbf{x}_i / \|\mathbf{y} - \mathbf{x}_i\|$ and $r_i(\mathbf{y}) = \|\mathbf{y} - \mathbf{x}_i\|$. For the weights w_i , we use a normalized Gaussian kernel: $w_i(\mathbf{y}) = g_i(\mathbf{y}) / \sum_{\mathbf{x}_j \in h_s} g_j(\mathbf{y})$, where $g_i(\mathbf{y}) = e^{-r_i^2(\mathbf{y}) / \sigma^2}$ with cutoff at 3σ , and $\sigma = 0.5$ (units are measured in pixel coordinates such that the distance between two pixel centers is 1). The σ effectively controls smoothing of the user strokes. For our setup of the interactive interface with a tablet, the value 0.5 provided enough smoothing for generating pleasing curves. It can be adjusted if less/extra smoothing is required for other setups.

Normal change at scale s The computed $r(\mathbf{y})$ and $\mathbf{d}(\mathbf{y})$ are used to retrieve the value stored at $\theta_s^{P(\mathbf{y})}(\mathbf{d}(\mathbf{y}), r(\mathbf{y}))$ (the retrieved values for segments of a stroke at three different scales are shown in Figure 12, bottom row). Finally, the retrieved value and the direction are used to form the quaternion $\mathbf{q}_s(\mathbf{y})$ with the rotation axis $\mathbf{d}(\mathbf{y})$ and angle $\theta_s^{P(\mathbf{y})}(\mathbf{d}(\mathbf{y}), r(\mathbf{y}))$. This quaternion represents the rotation of the normal $\mathbf{n}(\mathbf{y})$ of the input face model due to the stroke segment h_s . Note that since the statistics have a finite extent (see Section 4.3), only the points \mathbf{y} in a neighborhood of h_s need to be considered.

Combining normal changes at different scales Combining normal changes due to stroke segments is simply performed by averaging the quaternions at different scales: $\mathbf{q}(\mathbf{y}) = 1/m \sum_{s=1}^m \mathbf{q}_s(\mathbf{y})$, and normalizing the result. This

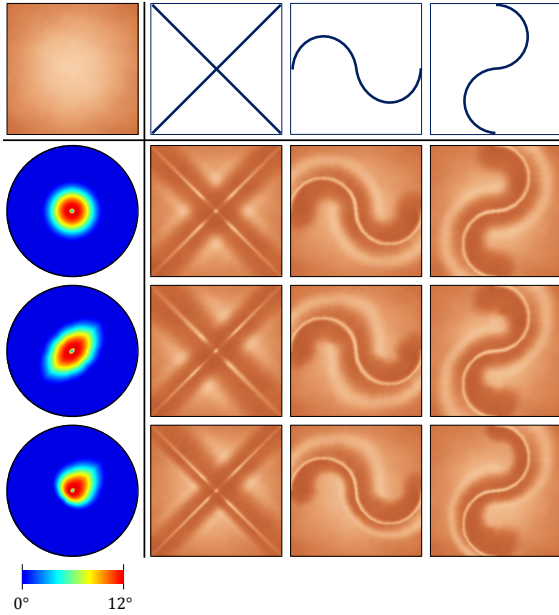


Figure 13: Simple statistics and strokes are combined to produce the illustrated wrinkles on the planar surface shown in top-left. Anisotropy in statistics is directly reflected to the shape of the resulting wrinkles.

quaternion is used to rotate the original normal of the input face to generate the new normal.

We illustrate example normal changes on a planar surface patch with simple statistics and strokes in Figure 13. For simplicity, the strokes contain only a single scale. The deformations clearly reflect the shapes of the statistics with anisotropy in similar directions. We present more complex examples with multiple scales in the next section.

6. Results and Analysis

In order to study the utility of our method in practice, we performed extensive tests with many sketches, input and example faces. The sketches are drawn interactively on the input face models.

6.1. Implementation

The interactive interface We implemented an interactive application where the users can draw strokes on an input face with a pressure-sensitive stylus. The pressure values are mapped to thicknesses of the strokes. The strokes are rendered with a color tone that depends on the local thickness in order to help the user better distinguish complex areas of a sketch with overlapping strokes (see Figures 12, and 14). The users get immediate feedback after the completion of each stroke (please also refer to the accompanying video).

Parameters The most important parameters are the σ_s 's that control the scales of the statistics. In the limit, an infinite number of scales with small increments can be selected to cover the full range of stroke thicknesses continuously. As explained in Section 5, in practice, it is sufficient to select a number of scales and interpolate the resulting normal changes from different stroke segments. For all results in this paper, we used $m = 6$ different scales with 2, 4, 6, 10, 18, 32 as the σ_s 's. This was sufficient to cover the range of details present in the used face models with approximately 1 million vertices (see Figure 5 for examples of details captured at different scales). As long as a reasonable range of scales is covered, the exact choice of σ_s 's does not alter the synthesized wrinkles significantly, as illustrated in Figure 15, where we shift all σ_s 's by a constant factor.

Apart from these, the method uses a parameter to control the maximum extent of the statistics (that determines the radius of the circular statistics figures we show throughout the paper) as explained in Section 4.3, and the standard deviation of the Gaussian for computing the weights w_i in Section 5.2. Effects of these parameters are discussed in the respective sections.

6.2. Wrinkle Synthesis Results

We used eight different sketches drawn on seven different input face models with statistics extracted from two different faces. The input and example faces are selected to cover different face types, genders, and races. The drawings exhibit different levels of complexity from simple ones containing a few mostly non-overlapping strokes to very sketchy ones. We present some of the results in Figures 1 and 14. Please also refer to the supplementary videos for additional results totaling around 170 output faces. The method can handle vastly different faces and transfer details between different face types. The results also illustrate that the synthesized wrinkles stay realistic for sketches of different complexity. The statistics accurately capture wrinkle details while respecting the user strokes and the characteristics of the input face. Each individual stroke, regardless of its interac-

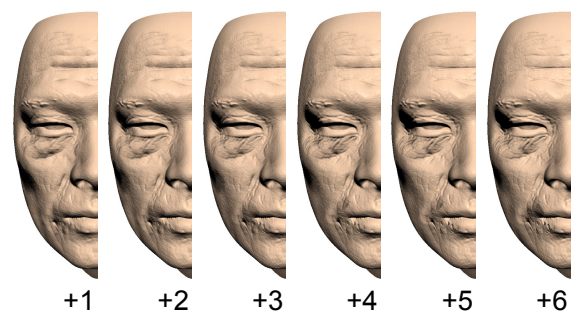


Figure 15: Altering the scales, such as by shifting σ_s 's by a constant factor, does not influence the synthesized wrinkles significantly.

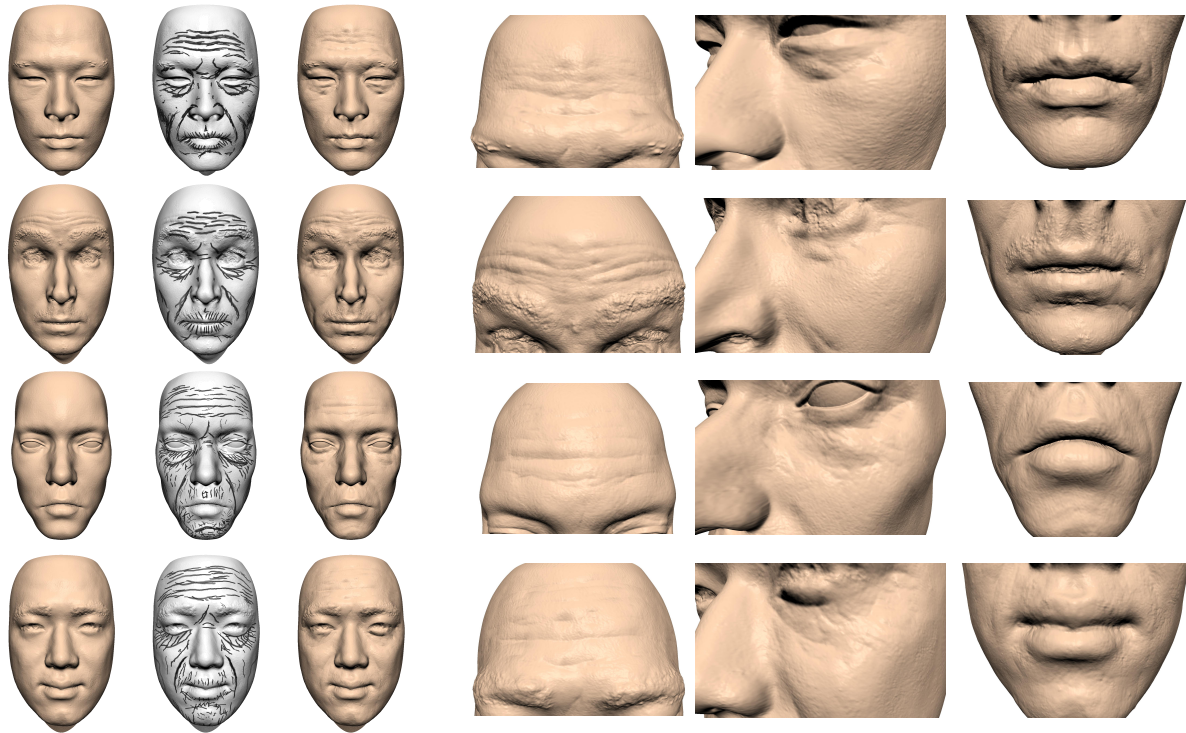


Figure 14: From left to right: input models, user sketches, output models, close-ups from the output models. The first two rows are generated with the statistics from Example 1 and the last two with those from Example 2 shown in Figure 16.

tions with the others, results in a face region specific realistic wrinkle.

Effect of example models We further study the effect of using different example faces on the results in Figure 16. Statistics extracted from three different example faces are used for augmenting the same user sketch on a given input face. For the forehead region, the synthesized wrinkles with the second example model are the strongest. The eye region is also slightly more curved with the second example face. The chin region, on the other hand, contains much stronger wrinkles with the first example face. These characteristics can also be observed on the example face models. Note that although the general behavior of the wrinkles depend on the chosen example model, the wrinkle shapes are determined by user strokes in all cases.

Combining example models For some cases, the users want to utilize wrinkle characteristics from different faces or amplify some of them according to their needs. Our method allows these modifications by simple linear operators on patch-wise statistics. We illustrate applying such operations to the statistics in Figure 17. In the top panel of the figure, the statistics in the eye region are amplified by a factor of 1.5, the middle panel illustrates linearly combining statistics from two example faces (rightmost and leftmost faces), and the bottom panel shows a case where statistics from two dif-

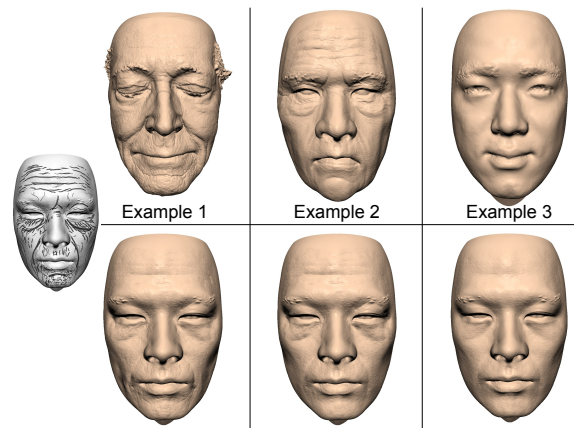


Figure 16: Statistics extracted from different faces (top row) are used to synthesize wrinkles on an input model. The synthesized details reflect the overall details of the wrinkles in different regions of the example models.

ferent faces are used for the upper and lower parts of the input model.

Effect of misalignment of faces As explained in Section 4.1, our method relies on an alignment of faces with a face template that also provides natural patch-wise corre-

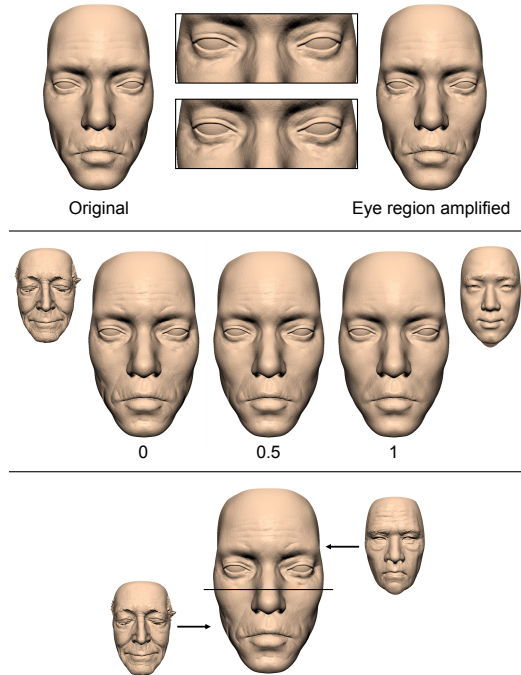


Figure 17: The figure illustrates some of the possible ways to alter or combine statistics if desired. Top panel: amplifying the statistics of the eye region by a factor of 1.5, middle panel: linearly interpolating the statistics of two given example faces shown on the left and right, bottom panel: taking statistics from two different example faces for the upper and lower parts of the input face.

spondences among different faces. Since we use averaged wrinkle shape functions, the actual alignments need not be precise. We illustrate this in Figure 18, where we apply different amounts of noise to the vertices of the face template for the input model (top) and example model (bottom). The resulting wrinkles on the input face stay almost the same for all cases.

Comparison with existing methods Existing methods for wrinkle generation from curves rely on parametric functions defining the shapes of the wrinkle profiles. Although they work well for sets of simple curves, they are not suited for realistic wrinkle synthesis from complex user sketches. This is illustrated in Figure 8 (top) for a state-of-the-art method [LLLC11]. For this method, we had to tune the parameters carefully to get the best result. With the help of the region specific statistics, our method can generate a more realistic result for the same sketch, as shown in Figure 8 (bottom), without having to specify any parameters.

Statistics from images The wrinkle statistics depend on rotated normal difference maps at different scales and hence are not affected by the overall shape of the facial surface, i.e. the lowest frequency components of the facial geome-

try. This property allows us to use images as a source of these maps and thus the statistics. We show an example image along with some angle maps and generated output faces in Figure 19. We chose the scales of the extracted normal maps such that they roughly match the ones we extracted from 3D scans. The resulting output faces (Figure 19, bottom row) contain the details of the characteristic wrinkles in the image, such as the ones around the eyes. Please refer to the accompanying video and images for more results with this set of statistics.

6.3. Informal User Study

We performed a qualitative user study with eight people, two of which are professional artists from the makeup and animation communities. All the non-artist participants have experience with 3D modeling tools.

The participants commented that due to the freedom of sketching overlapping strokes with varying thickness, and the immediate feedback, the interface feels natural to use, very similar to sketching with paper and pencil. Only the non-artist users with no experience with the stylus had difficulty controlling the thickness at first. But all participants were able to achieve the target look of an old person in a photo we presented in less than five minutes after a few practice strokes. Some of the non-artist participants informed us that they had bad sketching skills on paper, but enjoyed that they could sketch realistic wrinkles easily with the tool.

The animation artist further commented that the tool is particularly useful for drawing small scale wrinkles and associated details, which is normally very time consuming for him. The makeup artist was delighted about the ability to see the synthesized wrinkles out of his concept sketch, which he routinely uses for planning the actual makeup. The artists were also pleased that the sketched strokes can be stored and reused for other models and tasks, and increasing the complexity of sketches or the details synthesized does not affect the speed of the tool.

6.4. Discussion and Limitations

Although we mainly target small scale age related wrinkles, it is also possible to generate very old looks by introducing larger scale wrinkles as illustrated in Figure 20, where we used statistics from example face 1 in Figure 16 amplified by 1.5. We observed that in some regions (Figure 20, right, bottom), due to the smoothing introduced when computing the statistics, the full depth and scale of the bulging intended by the user cannot be reproduced, although in others (Figure 20, right, top) the method works well.

Normal maps can represent fine details without increasing mesh resolution, which is important for our target devices with low resources such as tablets. Normal maps also allow us to extract statistics from simple photos as illustrated in Figure 19. This is important for practical applica-

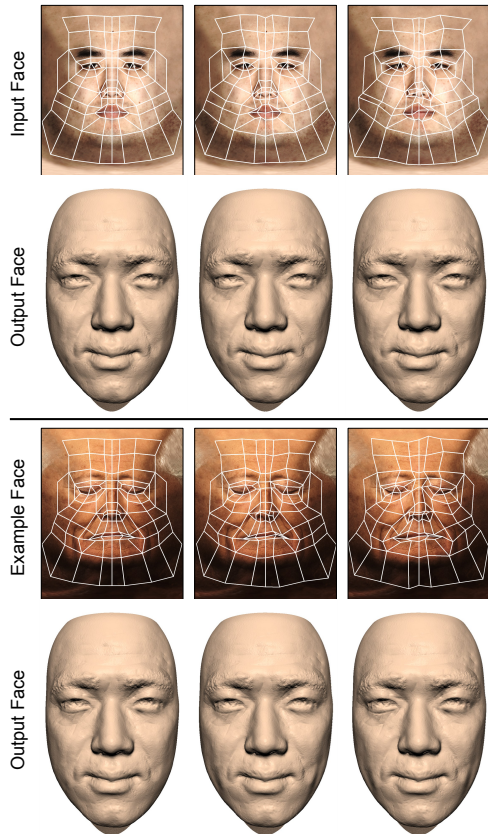


Figure 18: Different amounts of noise are applied to the input (top panel) and example (bottom panel) face models and the resulting output faces are shown. The synthesized wrinkles are robust to such misalignment of the face template.

tions since obtaining the required details by 3D scanning requires advanced scanning techniques. The downside is that well-known limitations of normal maps such as the lack of self shadowing and wrong silhouettes are inherited. To avoid the limitations associated with normal maps, it is possible to bake the changes into the geometry. We implemented a recent method [DM11] for this conversion, which gives very accurate results as we illustrate in Figure 21.

Our method relies on locally integrated statistics specialized for wrinkle geometry. These statistics extend previous representations for analysis and synthesis of facial detail [GMP*06, ZH06]. In common with these works, the statistics lead to synthesis results that respect shapes of target faces. This is essential for our purposes since the example and target faces can be significantly different. Hence, direct transfer of wrinkles with, e.g. centerlines deformed according to the strokes, cannot be used. Furthermore, in contrast to the previous works, the proposed statistics allow interactive stroke augmentation respecting user sketches while synthesizing small scale wrinkle details that are hard to be

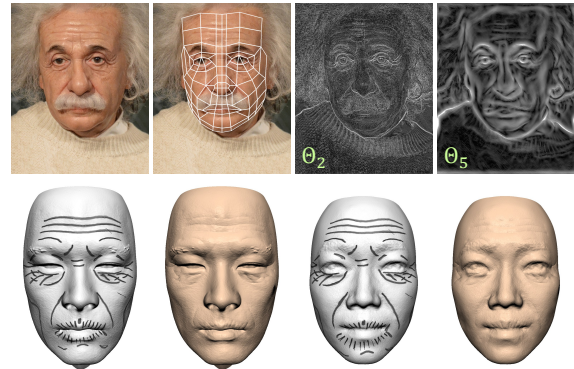


Figure 19: (Top row, from left to right): An image, the image with the face template, the maps Θ_2 and Θ_5 . We show two output faces generated with the extracted statistics in the bottom row.

sketched. Finally, the statistics can be easily manipulated with linear operators as discussed in Figure 17.

7. Conclusions

We presented an interactive method for synthesizing realistic wrinkles on faces with sketches. The method accepts expressive and sketchy drawings as input and automatically generates realistic wrinkles in real-time in an interactive interface. We mainly aim at providing a fast and realistic preview of wrinkle patterns from sketches commonly used for planning purposes. The generated models can be also used for other applications such as 3D games, digital storytelling, or makeup simulation.

Future directions

- The statistics and sketch interpretation techniques we presented can be extended by incorporating models of the skin texture and color. The realism of the results can be further increased by utilizing high resolution scanning systems [BBB*10] and realistic skin rendering techniques [JSB*10].
- In addition to synthesis, aging analysis can be performed using derived quantities from the statistics.
- The synthesized geometry can be used to print 3D silicon patches and 2D drawing patterns to be used for physical makeup.
- The synthesized wrinkles can be used to enhance the realism of animations in real-time applications [Oat07, JEOG11].
- Multi-scale detail map extraction from images can be improved by incorporating more advanced techniques (e.g. [BBZG12]).

Finally, we believe that developing specialized statistics coupled to user drawings to augment raw sketches can also be useful in many other synthesis-from-sketch tasks [OSSJ09] as well as in non-photo realistic rendering.

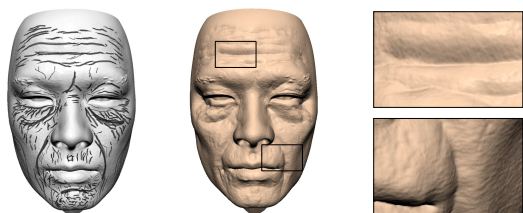


Figure 20: Our method can be used to generate very old looks. However, due to the smoothness introduced in the statistics, some of the bulging such as due to the large wrinkle depicted in right, bottom cannot be synthesized in full scale.

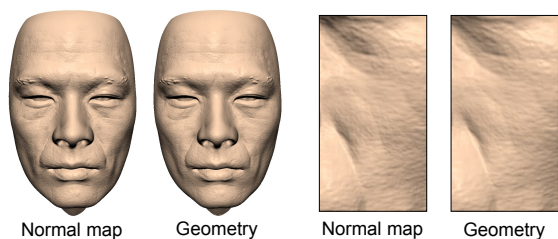


Figure 21: The generated normal maps can be converted into geometry changes with high accuracy.

The main challenge for such extensions will be developing appropriate measures for the particular application and sketch type targeted.

Acknowledgements We thank Dr. Daeyoung Kang, professional makeup artist of the Korea Makeup Studio, for his sketches and useful discussions. This work was supported in part by Korean NRF and Swiss SER under the Korean-Swiss Cooperative Program (2013K1A3A1A14055180) and in part by the NRF grant (2014R1A2A1A11053135).

References

[AHKS94] AKEO M., HASHIMOTO H., KOBAYASHI T., SHIBUSAWA T.: Computer graphics system for reproducing three-dimensional shape from idea sketch. *Comput. Graph. Forum* 13, 3 (1994), 477–488. 3

[ARL*10] ALEXANDER O., ROGERS M., LAMBETH W., CHIANG J.-Y., MA W.-C., WANG C.-C., DEBEVEC P.: The digital emily project: Achieving a photorealistic digital actor. *IEEE Comput. Graph. & App.* 30, 4 (2010), 20–31. 3

[BBA*07] BICKEL B., BOTSCH M., ANGST R., MATUSIK W., OTADUY M., PFISTER H., GROSS M.: Multi-scale capture of facial geometry and motion. *ACM Trans. Graph.* 26, 3 (July 2007), 33:1–33:10. 2

[BBB*10] BEELER T., BICKEL B., BEARDSLEY P., SUMNER B., GROSS M.: High-quality single-shot capture of facial geometry. *ACM Trans. Graph. (Proc. SIGGRAPH)* 29, 3 (2010), 40:1–40:9. 11

[BBB*14] BERMANO A. H., BRADLEY D., BEELER T., ZUND F., NOWROUZSAHRAI D., BARAN L., SORKINE-HORNUNG O., PFISTER H., SUMNER R. W., BICKEL B., GROSS M.: Facial performance enhancement using dynamic shape space analysis. *ACM Trans. Graph.* 33, 2 (Apr. 2014), 13:1–13:12. 3

[BBZG12] BEELER T., BRADLEY D., ZIMMER H., GROSS M.: Improved reconstruction of deforming surfaces by cancelling ambient occlusion. In *Proc. ECCV 2012* (Berlin, Heidelberg, 2012), ECCV'12, Springer-Verlag, pp. 30–43. 11

[BKN02] BANDO Y., KURATATE T., NISHITA T.: A simple method for modeling wrinkles on human skin. In *Proc. Pacific Graphics '02* (Washington, DC, USA, 2002), PG '02, IEEE Computer Society, pp. 166–175. 2, 4

[BKTK00] BOISSIEUX L., KISS G., THALMANN N., KALRA P.: Simulation of skin aging and wrinkles with cosmetics insight. In *Computer Animation and Simulation 2000* (2000), Magnat-Thalmann N., Thalmann D., Arnaldi B., (Eds.), Eurographics, Springer Vienna, pp. 15–27. 2

[BLB*08] BICKEL B., LANG M., BOTSCH M., OTADUY M. A., GROSS M.: Pose-space animation and transfer of facial details. In *Proc. SCA 2008* (Aire-la-Ville, Switzerland, 2008), SCA '08, Eurographics Association, pp. 57–66. 3

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99* (New York, NY, USA, 1999), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., pp. 187–194. 2

[CGW*05] CUTLER L. D., GERSHBEIN R., WANG X. C., CURTIS C., MAIGRET E., PRASSO L., FARSON P.: An art-directed wrinkle system for cg character clothing. In *Proc. SCA 2005* (New York, NY, USA, 2005), SCA '05, ACM, pp. 117–125. 2, 3

[CGW*07] CUTLER L. D., GERSHBEIN R., WANG X. C., CURTIS C., MAIGRET E., PRASSO L., FARSON P.: An art-directed wrinkle system for cg character clothing and skin. *Graphical Models* 69, 5-6 (2007), 219–230. 2, 3

[Deb09] DEBRECENI T.: *Special Makeup Effects for Stage and Screen*. Elsevier, 2009. 1, 2

[DM11] DMITRIEV K., MAKAROV E.: Generating displacement from normal map for use in 3d games. In *ACM SIGGRAPH 2011 Talks* (New York, NY, USA, 2011), SIGGRAPH '11, ACM, pp. 9:1–9:1. 11

[FGH10] FU Y., GUO G., HUANG T. S.: Age synthesis and estimation via faces: A survey. *IEEE Trans. Pattern Analysis and Machine Intelligence* 32, 11 (2010), 1955–1976. 1, 2

[FKY08] FENG W.-W., KIM B.-U., YU Y.: Real-time data driven deformation using kernel canonical correlation analysis. *ACM Trans. Graph.* 27, 3 (Aug. 2008), 91:1–91:9. 2

[GIZ09] GINGOLD Y., IGARASHI T., ZORIN D.: Structured annotations for 2d-to-3d modeling. *ACM Trans. Graph.* 28, 5 (December 2009), 148:1–148:9. 3

[GMP*06] GOLOVINSKIY A., MATUSIK W., PFISTER H., RUSINKIEWICZ S., FUNKHOUSER T.: A statistical model for synthesis of detailed facial geometry. In *ACM SIGGRAPH 2006 Papers* (New York, NY, USA, 2006), SIGGRAPH '06, ACM, pp. 1025–1034. 2, 3, 4, 11

[GTB*13] GRAHAM P., TUNWATTANAPONG B., BUSCH J., YU X., JONES A., DEBEVEC P., GHOSH A.: Measurement-based synthesis of facial microgeometry. *Comput. Graph. Forum* 32, 2 (2013), 335–344. 2

[IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: a sketching interface for 3d freeform design. In *Proc. SIGGRAPH '99* (New York, NY, USA, 1999), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., pp. 409–416. 3

[INN07] IGARASHI T., NISHINO K., NAYAR S. K.: The appearance of human skin: A survey. *Found. Trends. Comput. Graph. Vis.* 3, 1 (Jan. 2007), 1–95. 2

[JEOG11] JIMENEZ J., ECHEVARRIA J. I., OAT C., GUTIERREZ D.: *GPU Pro 2*. AK Peters Ltd., 2011, ch. Practical and Realistic Facial Wrinkles Animation, pp. 15–27. 11

- [JSB*10] JIMENEZ J., SCULLY T., BARBOSA N., DONNER C., ALVAREZ X., VIEIRA T., MATTS P., ORVALHO V., GUTIERREZ D., WEYRICH T.: A practical appearance model for dynamic facial color. *ACM Trans. Graph.* 29, 6 (Dec. 2010), 141:1–141:10. 11
- [KSvdP09] KRAEVOY V., SHEFFER A., VAN DE PANNE M.: Modeling from contour drawings. In *Proc. SBIM '09* (New York, NY, USA, 2009), SBIM '09, ACM, pp. 37–44. 3
- [LC04] LARBOULETTE C., CANI M.-P.: Real-time dynamic wrinkles. In *Proc. CGI '04* (Washington, DC, USA, 2004), CGI '04, IEEE Computer Society, pp. 522–525. 2
- [LK14] LI P., KRY P. G.: Multi-layer skin simulation with adaptive constraints. In *Proc. MIG '04* (New York, NY, USA, 2014), MIG '14, ACM, pp. 171–176. 2
- [LLC11] LI L., LIU F., LI C., CHEN G.: Realistic wrinkle generation for 3d face modeling based on automatically extracted curves and improved shape control functions. *Comput. & Graph.* 35, 1 (2011), 175 – 184. Extended Papers from NPAR 2010. 2, 3, 4, 5, 10
- [LXZ07] LI Y.-B., XIAO H., ZHANG S.-Y.: The wrinkle generation method for facial reconstruction based on extraction of partition wrinkle line features and fractal interpolation. In *Proc. ICIG '07* (Washington, DC, USA, 2007), ICIG '07, IEEE Computer Society, pp. 933–937. 2, 4
- [Man10] MANCUSI S.: *The Police Composite Sketch*. Springer, 2010. 2
- [MJC*08] MA W.-C., JONES A., CHIANG J.-Y., HAWKINS T., FREDERIKSEN S., PEERS P., VUKOVIC M., OUYOUNG M., DEBEVEC P.: Facial performance synthesis using deformation-driven polynomial displacement maps. *ACM Trans. Graph.* 27, 5 (Dec. 2008), 121:1–121:10. 3
- [MLD09] MA X., LE B. H., DENG Z.: Style learning and transferring for facial animation editing. In *SCA* (2009), Fellner D. W., Spencer S. N., (Eds.), ACM, pp. 123–132. 2
- [MTKLL*02] MAGNENAT-THALMANN N., KALRA P., LUC LEVEQUE J., BAZIN R., BATISSE D., QUERLEUX B.: A computational skin model: fold and wrinkle formation. *IEEE Trans. Inf. Technol. Biomed* 6, 4 (2002), 317–323. 2
- [NISA07] NEALEN A., IGARASHI T., SORKINE O., ALEXA M.: Fibermesh: designing freeform surfaces with 3d curves. *ACM Trans. Graph.* 26, 3 (July 2007), 41:1–41:10. 3
- [NSS*12] NORIS G., SYKORA D., SHAMIR A., COROS S., WHITED B., SIMMONS M., HORNUNG A., GROSS M., SUMNER R.: Smart scribbles for sketch segmentation. *Comput. Graph. Forum* 31, 8 (2012), 2516–2527. 2, 3
- [Oat07] OAT C.: Animated wrinkle maps. In *ACM SIGGRAPH 2007 Courses* (New York, NY, USA, 2007), SIGGRAPH '07, ACM, pp. 33–37. 11
- [OSSJ05] OLSEN L., SAMAVATI F. F., SOUSA M. C., JORGE J. A.: Sketch-based mesh augmentation. In *Proc. SBIM '05* (Aug 28-29 2005), Eurographics, pp. 43–52. 3
- [OSSJ09] OLSEN L., SAMAVATI F. F., SOUSA M. C., JORGE J. A.: Sketch-based modeling: A survey. *Comput. & Graph.* 33, 1 (2009), 85 – 103. 3, 11
- [Pix13] PIXOLOGIC: Zbrush, 2013. 3
- [PKA*09] PAYSAN P., KNOTHE R., AMBERG B., ROMDHANI S., VETTER T.: A 3d face model for pose and illumination invariant face recognition. In *AVSS '09* (Sept 2009), pp. 296–301. 2
- [RPC*10] ROHMER D., POPA T., CANI M.-P., HAHMANN S., SHEFFER A.: Animation wrinkling: augmenting coarse cloth simulations with realistic-looking wrinkles. In *ACM SIGGRAPH Asia 2010 papers* (New York, NY, USA, 2010), SIGGRAPH ASIA '10, ACM, pp. 157:1–157:8. 2
- [SEL06] SERUP J., E. J. G. B., L. G. G.: *Handbook of Non-Invasive Methods and the Skin*, 2 ed. Taylor and Francis, 2006. 1, 2
- [SSSB07] SCHERBAUM K., SUNKEL M., SEIDEL H.-P., BLANZ V.: Prediction of individual non-linear aging trajectories of faces. In *EUROGRAPHICS 2007* (Prague, Czech Republic, 2007), vol. 26 of *Computer Graphics Forum*, The European Association for Computer Graphics, Blackwell, pp. 285–294. 2
- [STDN10] SUCONTPHUNT T., TUNWATTANAPONG B., DENG Z., NEUMANN U.: Crafting 3d faces using free form portrait sketching and plausible texture inference. In *Proc. GI 2010* (Toronto, Ont., Canada, 2010), GI '10, Canadian Information Processing Society, pp. 209–216. 2
- [TW90] TERZOPOULOS D., WATERS K.: Physically-based facial modelling, analysis, and animation. *The Journ. Visualization and Comput. Animat.* 1, 2 (1990), 73–80. 2
- [VLR05] VENKATARAMAN K., LODHA S., RAGHAVAN R.: A kinematic-variational model for animating skin with wrinkles. *Comput. & Graph.* 29, 5 (2005), 756–770. 2
- [Wat87] WATERS K.: A muscle model for animation three-dimensional facial expression. In *Proc. SIGGRAPH '87* (New York, NY, USA, 1987), SIGGRAPH '87, ACM, pp. 17–24. 2
- [WMTT94] WU Y., MAGNENAT THALMANN N., THALMANN D.: A plastic-visco-elastic model for wrinkles in facial animation and skin aging. In *Proc. PG '94* (River Edge, NJ, USA, 1994), Pacific Graphics '94, World Scientific Publishing Co., Inc., pp. 201–213. 2
- [WYW06] WANG Y., WANG C. C., YUEN M. M.: Fast energy-based surface wrinkle modeling. *Comput. & Graph.* 30, 1 (2006), 111–125. 2, 3
- [ZH06] ZAHOUANI H., HUMBERT P.: *Handbook of Non-Invasive Methods and the Skin, Second Edition*. Taylor & Francis, 2006, ch. The Morphological Tree of the Cutaneous Network of Lines. 3, 11
- [ZHH06] ZELEZNIK R. C., HERNDON K. P., HUGHES J. F.: Sketch: An interface for sketching 3d scenes. In *ACM SIGGRAPH 2006 Courses* (New York, NY, USA, 2006), SIGGRAPH '06, ACM. 3
- [ZHS*05] ZHOU K., HUANG J., SNYDER J., LIU X., BAO H., GUO B., SHUM H.-Y.: Large mesh deformation using the volumetric graph laplacian. *ACM Trans. Graph.* 24, 3 (July 2005), 496–503. 3
- [ZPS01] ZHANG Y., PRAKASH E., SUNG E.: Real-time physically-based facial expression animation using mass-spring system. In *Proc. CGI '01* (Washington, DC, USA, 2001), IEEE Computer Society, pp. 347–350. 2
- [ZPS04] ZHANG Y., PRAKASH E. C., SUNG E.: A new physical model with multilayer architecture for facial expression animation using dynamic adaptive mesh. *IEEE Trans. Visualization and Comput. Graph.* 10, 3 (May 2004), 339–352. 2
- [ZS05] ZHANG Y., SIM T.: Realistic and efficient wrinkle simulation using an anatomy-based face model with adaptive refinement. In *Proc. CGI '05* (Washington, DC, USA, 2005), CGI '05, IEEE Computer Society, pp. 3–10. 2, 4
- [ZSTS06] ZHANG Y., SIM T., TAN C. L., SUNG E.: Anatomy-based face reconstruction for animation using multi-layer deformation. *Journ. Visual Languages & Computing* 17, 2 (2006), 126 – 160. 2