Eurographics Symposium on Geometry Processing 2023
P. Memari and J. Solomon
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 5

Graph-Based Synthesis for Skin Micro Wrinkles

S. Weiss[1] and J. Moulin[2] and P. Chandran[1] and G. Zoss[1] and P. Gotardo[1] and D. Bradley[1]

[1] DisneyResearch|Studios [2] Industrial Light & Magic

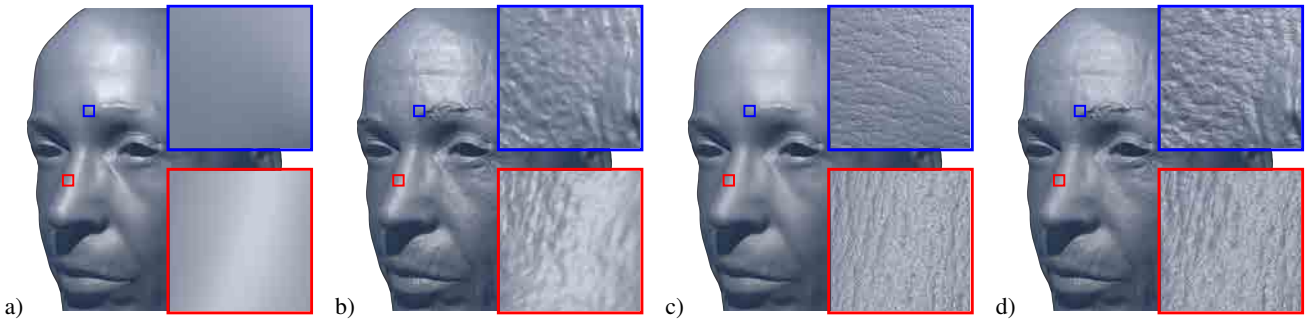

Figure 1: *Current facial scanning techniques can accurately capture the base mesh (a) and the macro-scale wrinkles (b), but struggle with the fine micro-wrinkles. We propose a procedural wrinkle simulation targeted for these micro wrinkles (c). The best result is achieved when the coarse wrinkles from the 3D reconstruction and the generated fine wrinkles are blended together (d).*

Abstract

We present a novel graph-based simulation approach for generating micro wrinkle geometry on human skin, which can easily scale up to the micro-meter range and millions of wrinkles. The simulation first samples pores on the skin and treats them as nodes in a graph. These nodes are then connected and the resulting edges become candidate wrinkles. An iterative optimization inspired by pedestrian trail formation is then used to assign weights to those edges, i.e., to carve out the wrinkles. Finally, we convert the graph to a detailed skin displacement map using novel shape functions implemented in graphics shaders. Our simulation and displacement map creation steps expose fine controls over the appearance at real-time framerates suitable for interactive exploration and design. We demonstrate the effectiveness of the generated wrinkles by enhancing state-of-art 3D reconstructions of real human subjects with simulated micro wrinkles, and furthermore propose an artist-driven design flow for adding micro wrinkles to fictional characters.

1. Introduction

The creation of high quality digital characters is very common in many application scenarios, such as VFX for feature films, video games, and other virtual environments like VR and telepresence. A crucial step in producing realistic digital characters is the accurate representation of skin geometry. In particular, human skin exhibits micro-details that break up specular reflectance and lead to an organic (less plastic-like) appearance (Fig. 1). Obtaining these micro-structures using traditional 3D scanning techniques is extremely challenging, requiring specialized hardware and dedicated reconstruction methods, and modeling them by hand would be nearly impossible in a reasonable amount of time.

An attractive alternative is to *simulate* skin micro-geometry, but the challenges lie in 1) producing high-quality and realistic wrinkle details that complement the underlying coarse geometry; and 2) allowing intuitive artistic control over the synthesis process. In this work, we present a new simulation method for skin micro wrinkle synthesis that overcomes both of these challenges.

Before proceeding, we feel it is important to define the different scales of skin wrinkles that we will refer to throughout this work:

- Macro scale (Fig. 1 b): Pimples and wrinkles in the scale of multiple millimeters. These wrinkles include the expression wrinkles created by skin compression in various facial expressions, as well as permanent wrinkles due to skin sagging with age. Wrinkles of this scale are usually captured directly in reconstructed triangle meshes [RGB*20] or modeled by artists.
- Micro scale (Fig. 1 c): pores (sweat glands and hair follicles) and wrinkles in the scale of tens to hundreds of micrometers between

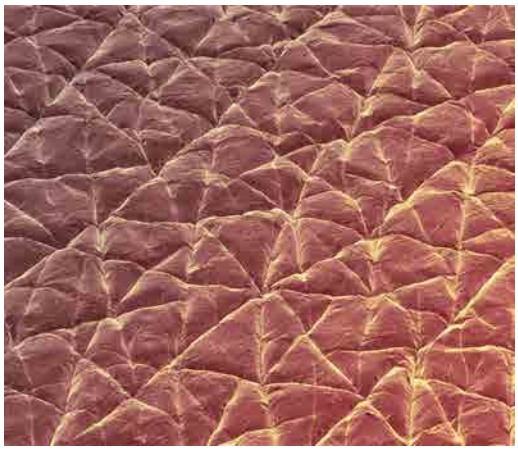

Figure 2: *A scanning electron micrograph (SEM) of the surface of real human skin. The patch width is approximately 3.4mm. Note how micro wrinkles tend to begin and end at pore locations (Image courtesy of Science Photo Library).*

neighboring wrinkles (see Fig. 2). At this range, traditional image-based reconstruction methods cannot recover detail.

The micro-scale wrinkles, sometimes called the microrelief, connect the pores and the hair follicles. The density distribution and the depth of the wrinkles varies from skin region to skin region [LC93, CCL84, BL11]. On the scalp, only 1-2 such wrinkles are observed in an area of 1mm width, on the forearm around 10. In regions with anisotropic stress, i.e. when compression and stretching happens predominantly in a single direction, the deepest wrinkles (primary wrinkles) form rectangles aligned with that direction. On the face, this primary orientation is also called the facial flow. Smaller secondary wrinkles criss-cross those primary wrinkles in a more random, diagonal pattern. In regions of uniform isotropic stress the wrinkles form relatively regular triangular patterns. These primary wrinkles are between 20 to 100 μm deep, depending on age, the secondary wrinkles between 5 to 40 μm deep.

We present a simulation approach that can procedurally create realistic wrinkles at the micro scale to enhance macro-level skin models obtained from 3D reconstruction techniques or from artistic sculpts. Our method can be applied to any part of the body, and we demonstrate results on both human faces and hands. Instead of utilizing neural networks for super-resolution (see Section 2), a graph-based simulation is proposed where the nodes correspond to the skin pores and the edges to the candidate wrinkles. Inspired by pedestrian simulations, edges are iteratively carved out following a pre-defined set of rules, e.g., prefer edges oriented along the facial flow lines (Section 3.1). The resulting nodes and edges are then drawn into a displacement map using wrinkle shape functions, giving rise to pores and micro wrinkles (Section 3.2).

To steer the simulation, we propose several control mechanisms. First, we create interactive brushes for pure artistic design targeted for fictional characters (Section 4.3). Second, we show that our micro-wrinkles can be integrated with existing macro-scales by estimating initial pore and wrinkle orientations using traditional computer vision algorithms (Section 4.1). Third, we develop an optimization scheme to reconstruct simulation parameters from high-resolution displacement map patches of the target person, if available (Section 4.2). The idea is that existing techniques capable of reconstructing micro wrinkles for small skin patches [GTB*13] can be applied sparsely and then our method can fit simulation parameters that can be used for the whole surface.

To summarize, in this work we propose

- a graph-based representation for skin micro wrinkles, with the skin pores as nodes and micro wrinkles as edges,
- a parallel simulation of wrinkle selection and carving inspired by pedestrian trail formation,
- a new wrinkle shape function targeted for efficient rendering and fine control,
- and integration into various editing workflows, from interactive artistic controls to parameter reconstruction from patches.

2. Related Work

We now present related work in 3D face reconstruction, patch-based skin capture, procedural wrinkle generation and skin super-resolution methods.

3D Face Reconstruction. As one of the first works to capture the whole face, Nahas *et al.* [NHRD90] propose a laser range scanner setup with a spatial sampling resolution of 0.6*mm* to 0.8*mm*. To capture the geometry at macro level, camera-based studio capture setups utilize marker-based mesh fitting with coarse geometry [BBA*07], multi-view stereo [BBB*10], or shape-from-specular with hundreds of individually controllable light sources [GFT*11]. These setups can reach impressive quality including dynamic albedo variations due to blood flow [GRB*18] or real-time reconstruction with coarse wrinkles [CBZB15], but require either complex expensive hardware or fail to recover dense micro wrinkle geometry. To remove the requirement of a studio setup with multiple calibrated cameras and light sources, several methods investigate geometry capture setups using commodity hardware. This includes material acquisition in-the-wild [RRFG17, NLG22, LLK*22], or using flash- and no-flash image pairs [CWS*20, PCF05], which are also applicable to human faces but limited to a macro scale. In a different thread, many neural networks have been proposed to estimate the 3D geometry directly from one or several 2D face images [TZK*17, GPKo19, DYX*19, GZY*20, FFBB21, DTA*21, WBH*22]. While these approaches are attractive in their ease of usability, and many can achieve real-time reconstruction, they target low resolution base-level geometry that contains neither macro nor micro details. In contrast to these existing 3D face capture methods, we target details at a micro level. For example, the best studio-based capture setups report displacement map resolutions up to $4k^2$ pixels (mean pixel size of 0.08*mm*) [RGB*20]. In order to fully resolve the micro-scale wrinkles, we target a resolution of $16k^2$, i.e. a mean pixel size of 0.02*mm*, in this work.

Patch-Based Skin Capture. Instead of capturing the whole face at once, several methods have been proposed that focus on small patches of the human skin instead. Haro *et al.* [HGE01] use silicon molds to capture a small piece of skin and then use photometric

stereo to reconstruct the displacements. Then, texture synthesis using vector quantization [WL00] is used to grow a displacement map spanning the whole face. Graham *et al.* [GTB*13] propose a scanning setup where the skin area is placed against a metal plate with a square hole. On the other side, a single camera and multiple active lights are used to reconstruct the surface geometry from shading, achieving a spatial resolution of $0.03mm$ for a patch of $24mm \times 16mm$. To extend the patch over the whole face, Image Analogies [HJO*01] was used. This scanning setup was then later extended by Nagano *et al.* [NFA*15] with a simulation of stretching and compression. By measuring the skin under different deformations, they derived a linear model on how the scale (depth + spatial frequency) of the wrinkles change under stretching and compression. To go beyond the micro scale and capture the cellular level, Balu *et al.* [BMH*16] present a microscope setup with a resolution of up to $0.5\mu m$ on an image area of $0.8 \times 0.8mm^2$. While patch-based skin capture allows for very realistic faces, it is not controllable or applicable to fictional characters where scans are impossible. This motivates a completely procedural, artist-driven wrinkle simulation approach, as we present in this work. Furthermore, we show how to integrate our simulation approach into existing scanning applications, i.e. by estimating simulation parameters from coarse scale displacement maps or high resolution skin patches.

Procedural Wrinkle Generation. Existing methods for procedural generation of skin wrinkles typically fall into one of two categories: medically-informed simulations that try to represent the actual skin as physically accurate as possible, and empirical-based simulations that use approximations to get plausible results quickly. On the medical side, Flynn and McCormack [FM10] perform a stress simulation using the finite element method (FEM) of a small skin patch in 3D. The elasticity parameters were altered to analyze their influence and to match them against measured displacements of actual human skin. A similar simulation is conducted by Limbert and Kuhl [LK18] with a focus on how skin wrinkles change from fine patterns seen in young skin towards deep, coarse creases seen in old skin due to an increase in the stiffness ratio. On the empirical side, early simulations use Voronoi cells [IYYT93] or Delaunay triangulation [WKT96] to form the base polygons. Those polygons are then embossed to form the wrinkles. This type of simulation is mostly appropriate for skin areas exhibiting isotropic stress, e.g. the forearm, less so for areas with anisotropic stress and thus dominating primary wrinkle directions. To model anisotropic areas, Bando *et al.* [BKN02] introduces a greedy method that draws lines following a user-control flow field. Further parameters control the spacing between the lines, thickness, and depth of the resulting wrinkles. Vanderfeesten and Bikker [VB18] later revise and refine this algorithm, by reducing the number of user-defined parameters and working in geometry-space instead of image-space to avoid the dependency on the texture resolution. Li *et al.* [LXZ07] build upon the drawing framework from Bando *et al.*, but extract the lines from given albedo images using image filters. This work was then extended with a more refined extraction algorithm of the coarse expression wrinkles from face photographs, and an extended mesh subdivision and displacement scheme [LLLC11]. While closely related to our method, the work of Bando *et al.*(and follow up studies) has two main disadvantages that are both solved by the proposed method. First, theirs is an inherent sequential algorithm where each wrinkle is drawn after the previous, which makes it difficult to interactively design the parameters. Second, wrinkles start and end at random locations and can have open ends, whereas fine wrinkles on the human skin always meet at pores (see Fig. 2). To already offer a comparison, Fig. 3 shows an example of a skin patch simulated with the method of Bando *et al.* versus our simulation, which more closely matches real skin (Fig. 2).

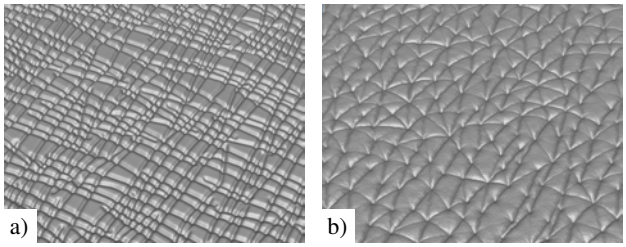

Figure 3: *The method of Bando* et al. *[BKN02] draws each wrinkle one after another while respecting certain distance constraints (a). The resulting wrinkles can thus begin and end at arbitrary positions. Our graph-based simulation enforces wrinkles to connect at sampled pores (b), creating a more realistic look.*

Super-Resolution Methods. If coarse skin geometry is available, super-resolution methods can infer missing higher frequency details. To this end, Saito *et al.* [SWH*17] built a database of detail skin patches and interpolated them linearly to match different regions of coarse face data. Yamaguchi *et al.* [YSN*18] show a simple convolutional network that is trained to perform 4x super-resolution on reprojected albedo and displacement maps, from 512^2 to 2048^2. A similar approach is taken by Huynh *et al.* [HCS*18], with the modification of using two different 4x upscaling paths, one for high-frequency displacements, the other for mid-frequency displacements. The network is applied patch-by-patch and then stitched together using Image Quilting [EF01] resulting in a final $4k^2$ displacement map. In our work, we also target the application of micro wrinkle simulation when coarse geometry is not necessarily available.

3. Simulating Skin Micro Wrinkles

Based on investigation of medical models [LC93, CCL84, BL11], skin pores and hair follicles act as initial skin irregularities and wrinkles form between them due to repeated stretch and compression of the skin in motion. Therefore, we present a procedural micro geometry method by simulating wrinkles and pores as a graph (Section 3.1). For every such generated pore and wrinkle, a geometry patch is instantiated and drawn in UV-space on a displacement map using a controllable shape function as the wrinkle profile (Section 3.2). The resulting displacement map can be rendered in real-time on a base mesh or stored for high quality offline rendering.

Skin pores and micro wrinkles exist at multiple fine resolutions. For simplicity, we will describe the simulation for a single level. To obtain the most realistic results, however, a multi-level approach is ultimately employed with repeatedly smaller distances between the pores. This allows to capture the different features at different scales, e.g. more directional at coarser scale and more regular at finer

scale. An example of the wrinkle generation over multiple levels is shown in the supplementary video. In the following, all user-defined parameters are denoted with α_{suffix}. These parameters are specified per simulation level and can be automatically optimized or interactively edited (Section 4). We will describe our method in the context of micro wrinkles on the face for a single level, but will also demonstrate generalization to other skin (e.g. hands) in Section 5.

3.1. Graph-based Simulation of Wrinkles

Throughout the following description, we assume we are given a UV-wrapped base face mesh. The simulation is then split into two stages, graph creation and iterative wrinkle simulation.

3.1.1. Graph Creation

During graph creation, first the set of nodes in UV-space is sampled, $V := \{v_i \in [0,1]^2, i = 1,...,N_V\}$. For the per-patch parameter fitting, see Section 4.2, the nodes are sampled using uniform Poisson disk sampling with a user-defined average node distance α_d. For simulations on the whole face, we utilize a spatially-varying minimal distance as input to the Poisson disk sampling to generate nodes that are approximately uniformly spaced apart on the 3D mesh after UV projection. First, a radius scaling map $\mathbf{R}$ is computed to compensate for UV stretching using an approximation of the geodesic distance [BWWM10]. Areas with $\mathbf{R} > 1$ indicate regions where the UV map is squashed together and nodes should therefore be spread out more, and vice versa. Then, this map $\mathbf{R}$ is multiplied by the average node distance α_d and passed to the Poisson disk sampling. The difference between uniform and geometry-aware sampling is visualized in Fig. 4. This process, however, assumes isotropic scale correction. For a more accurate anisotropic sampling, we refer to Li *et al.* [LWSF10]. To combine simulations over multiple levels, the graph at a finer level is initialized with the nodes from the previous, coarser level and new nodes are added in-between.

Next, for each node v_i, the k-nearest neighboring nodes v_j are computed and the edge $\{v_i, v_j\}$ added to the list of edges E. The resulting graph is denoted by $G = (V, E)$. Note that due to the graph being undirected, each node is connected to at least k neighbors, but typically a few more. The graph edges form candidate wrinkles that will be selected and carved out in the wrinkle simulation step.

For an efficient computation, the graph is stored as a half-edge data structure. Let H_v be the set of half-edges originating at node v. Each half-edge $h \in H_v$ stores the target node $t_h \in V$, the angle from the start to the target node θ_h, and, shared with the twin half-edge, the index of the edge w_h, the geodesic distance between the incident nodes d_h, and the list of all crossing edges $C_h \subset E$.

3.1.2. Iterative Wrinkle Simulation

Due to anisotropic stress on the skin caused by muscle deformations in predominant directions, the micro detail on the skin often shows a significant directional component. The direction of this major stress can be described as the *facial flow lines* [Lan61, AUM76]. For the simulation, this anisotropy is modeled as an orientation sampler providing the probability of selecting an outgoing graph edge at angle θ at the location of node v, $\mathbb{P}[\theta; v]$.

In this work, we model the angle probability using two spatially varying parameters, the primary angle $\alpha_{\hat{\theta}}$ and the base strength α_s. Then the probability is defined as

$$\mathbb{P}[\theta] \propto \alpha_s + \exp\left(-\frac{|\theta - \alpha_{\hat{\theta}}|_\pi^2}{\sigma^2}\right), \tag{1}$$

$$\text{with } |\theta_1 - \theta_2|_\pi := \min\{|\theta_1 - \theta_2|, |\theta_1 - \theta_2 - \pi|, |\theta_1 - \theta_2 + \pi|\}. \tag{2}$$

We use a Gaussian distribution here to control the spread, i.e., how sharp the fall-off of the probability from the target direction shall be. The variance is set to $\sigma = 0.3$ as default. Note, that the orientation sampler is undirected. A direction θ is equally likely as a direction $\theta + \pi$. The orientation probabilities will be used in the wrinkle simulation, described next. The influence of the base strength α_s (after simulation) is visualized in Fig. 5.

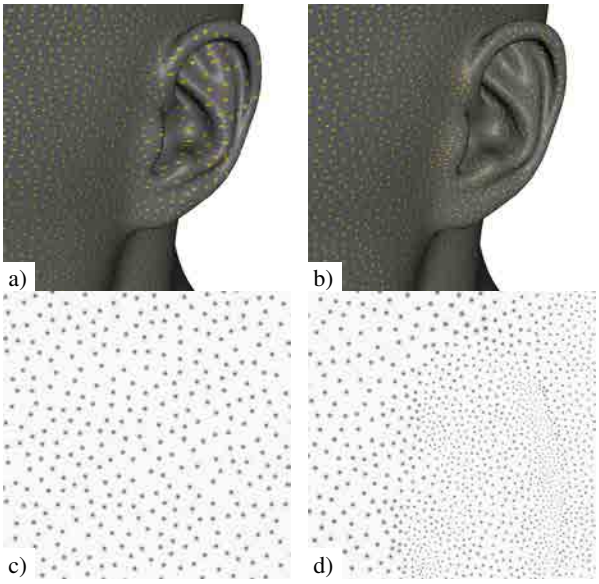

Figure 4: *Influence of the geometry-aware node sampling, without (a,c) and with (b,d) adaptive node distance. The top row shows the rendered face after UV mapping with the generated pores shaded in yellow, zoomed in on the ear. The bottom row shows the same part of the displacement map in UV space.*

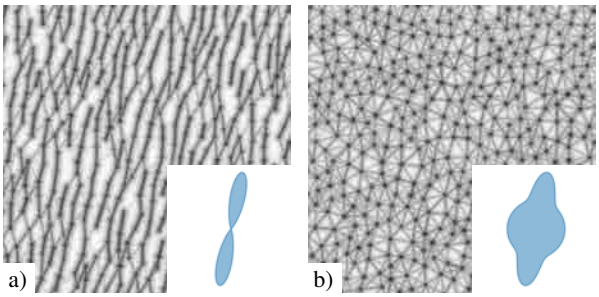

Figure 5: *Influence of the base strength α_s in the orientation sampling. In (a), $\alpha_s = 0$ and only the primary direction influences the wrinkles. In (b), $\alpha_s = 1.0$ and the wrinkles have uniform orientation.*

For the remaining simulation, the graph as described above is fixed in topology and node locations. Inspired by pedestrian simulations and trail formation [HMS98, HMFB01, HKM97], wrinkles are carved out by repeatedly walking from one node to another. Unlike pedestrian simulations that simulate trail formation in free space, we work purely on the graph, drastically reducing computation cost.

Our goal is to compute a vector of edge weights describing how deep each wrinkle is to be carved out. Let $W \in \mathbb{R}^{|E|}$ be this vector, which we initialize to zeros. As each half edge h stores the index of the edge w_h, we denote the weight of the half edge as $W_h := W_{w_h}$ for ease of notation. Then, for a randomly selected node v_i, we choose an incident edge $h \in H_{v_i}$ with probability $\mathbb{P}[h]$ proportional to

$$\mathbb{P}[h] \propto \max\left(0, \frac{1}{d_h^{\alpha_{\text{dist}}}}\left(p_{\text{cont}} + (1-\alpha_{\text{cont}})p_{\text{pref}} - \alpha_{\text{cross}}p_{\text{cross}}\right)\right), \tag{3}$$

$$p_{\text{cont}} := \sum_{h' \in H_{v_i}, h' \neq h} W_{h'}\left(\alpha_{\text{cont}} N(|\theta_h - (\theta_{h'} + \pi)|_{2\pi}; 0, \sigma_{\text{cspread}}) - \alpha_{\text{sim}} N(|\theta_h - \theta_{h'}|_{2\pi}; 0, \sigma_{\text{cspread}})\right), \tag{4}$$

$$p_{\text{pref}} := \mathbb{P}[\theta_h], \tag{5}$$

$$p_{\text{cross}} := \sum_{h' \in C_h} W_{h'}. \tag{6}$$

Here, α_{dist} weighs the distance between nodes, with a low value indicating low influence of the distance, and a high value leading to preferring the shortest connection. The preferred orientation due to the facial flow is included with p_{pref} and crossings between edges are penalized by p_{cross}. To steer the simulation towards generating longer connected wrinkles, p_{cont} both rewards edges that have a strong edge in the opposite direction to continue that wrinkle (α_{cont}) and penalizes T-crossings by weighting similar directions lower (α_{sim}). In Eq. 4, $|\theta_1 - \theta_2|_{2\pi}$ returns the difference between those two angles with a 360° wrap-around (see Eq. 2) and $N(x; 0, \sigma_{\text{cspread}})$ is the Gaussian distribution with standard variation $\sigma_{\text{cspread}} = 0.3$.

Once a half-edge h is sampled, the edge weight is increased to carve out a wrinkle. We adopt the equation proposed by Helbing *et al.* [HMFB01] to saturate the maximal wrinkle depth to one,

$$W_h \leftarrow W_h + \Delta t\, \alpha_{\text{deposit}}(1 - W_h), \tag{7}$$

where $\alpha_{\text{deposit}} \in [0.01, 1.0]$ is the deposit strength and $\Delta t = 0.03$ is a global time step. After N edges are simulated, a decay is applied that reduces the overall wrinkle depth globally, akin to trails disappearing over time in the trail formation analogy. The decay is realized as

$$W_i \leftarrow W_i(1 - \Delta t\, \alpha_{\text{decay}}\, N/|E|)\ \forall i. \tag{8}$$

We simulate $N = 50000$ edges in parallel using one CUDA kernel to fully utilize the GPU. The ratio between α_{deposit} and α_{decay} influences how saturated the edges are. Typically, we fix $\alpha_{\text{decay}} = 0.5$ and only vary α_{deposit}. All parameters are user-controllable, and we illustrate the effects of setting each parameter to its respective maximum value (as compared to default) in Fig. 6. Initially, the four reward and penalty parameters $\alpha_{\text{dist}}, \alpha_{\text{cont}}, \alpha_{\text{sim}}, \alpha_{\text{cross}}$ are set to zero and α_{deposit} is set to 0.25. Then the artist can interactively dial in the different reward and penalty terms to achieve the desired look. We refer to the supplementary material for a table of the parameter values used to generate Fig. 1 and Fig. 16. Finally, a global scaling factor α_{scale} is applied multiplicative to all displacements before drawing the wrinkles (as described next). This is especially useful to balance multiple layers of simulation. Examples of the achievable effects when the parameter values are randomly sampled are shown in the supplementary material.

3.2. Controllable Wrinkle Drawing

The simulation above describes the generated wrinkles in terms of edge weights $W \in \mathbb{R}^{|E|}$ on an undirected graph. To now create geometric wrinkles, we must convert the graph to a UV displacement map by "rendering" the wrinkles into a displacement texture space.

3.2.1. Shape Function

Following the ideas of Bando *et al.* [BKN02], Zhang and Sim [ZS05], and similar works, we draw wrinkles as lines with a given width and depth. For each pore and wrinkle, we instantiate a geometry patch that is then transformed in a vertex shader so that it starts and ends at the incident nodes of the current wrinkle and has a width proportional to the strength of the wrinkle scaled by the user-defined parameter $\alpha_{\text{wrinkle-width}}$, see Fig. 7a.

Then, in a fragment shader, the displacements applied are controlled by the so-called wrinkle shape function, a function depending on the distance x from the wrinkle center line. For each fragment, the shape function is queried and the displacements are accumulated into an offscreen displacement texture, scaled by the strength of the wrinkle W_i. We propose two wrinkle shape functions, defined as

$$\begin{aligned} S_{\text{ours, bulge}}(x) &:= (x-3)^2(x-1)/9,\ x \in [0,3] \\ S_{\text{ours, no bulge}}(x) &:= -(1-2x/3)^3,\ x \in [0,1.5], \end{aligned} \tag{9}$$

where the first option is a polynomial approximation of the shape function proposed by Bando *et al.* [BKN02] on the bounded domain of $3\times$ the initial width w of the wrinkle. Here, the interval $[0,1]$ represents the main wrinkle valley and $[1,3]$ the bulge. The second option is a simplification on half of the domain, thus requiring geometry that is only half as wide and improves performance by reducing overdraw. For a comparison with other shape functions proposed in literature, we refer to the supplementary material. In practice, we use "ours, no bulge" for the finest level only to keep the rendering time low and use "ours, bulge" for all other levels.

When rendering the wrinkles into the displacement map, a texel might be covered by multiple wrinkles, e.g. at the pores or with parallel wrinkles if the artist-defined width of the wrinkle is very large. Therefore, we utilize a variation of the MellowMax-function [AL17] to smoothly blend the different fragments together. Let $\hat{N}_f$ be the maximal number of fragments that are expected to be drawn on top of each other, we empirically set $\hat{N}_f = 16$, then the changed MellowMax function is defined as

$$\text{mellowmax}(\mathbf{x}) := \frac{1}{\beta}\log\left(\frac{1}{\hat{N}_f}\left(\hat{N}_f - N_f + \sum_{i=1}^{N_f}\exp(\beta x_i)\right)\right) \tag{10}$$

with $\beta = 20$ and $\mathbf{x} \in \mathbb{R}^{N_f}$ the values of the fragments written to the current pixel. In the rare case that N_f exceeds $\hat{N}_f$, we resort

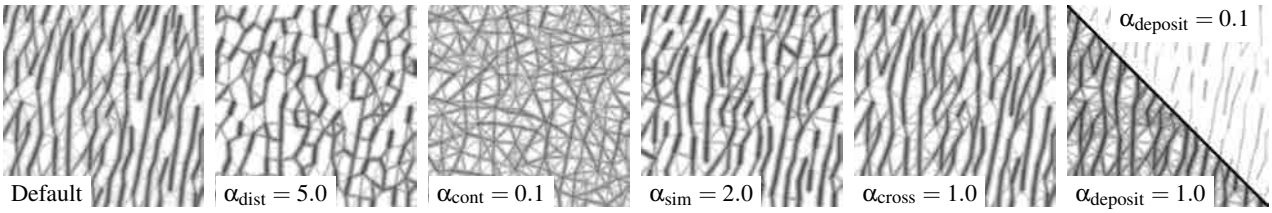

Figure 6: *Visualizing the main user controllable parameters of the simulation. The orientation sampler is kept constant. In "Default", $\alpha_{deposit} = 0.25$ and all four other parameters are set to zero. Then, from left to right: prefer shorter edges, prefer edges that continue incoming edges, penalize edges with similar outgoing directions, penalize edge crossings. Last, the deposit parameter is set to the min and max value.*

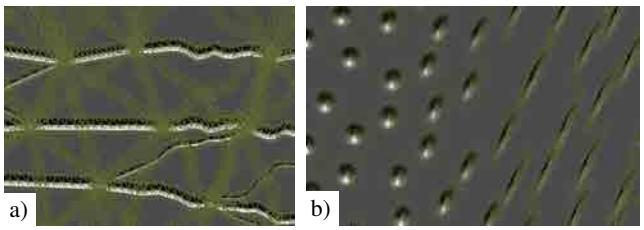

Figure 7: *Geometry that is instantiated for the wrinkles (a) and pores (b). Furthermore, (a) shows the influence of the wrinkle perturbation $\alpha_{sperturb}$ and (b) the node skew α_{skew}, see Section 3.2.2.*

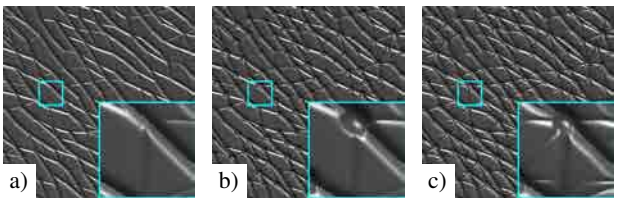

Figure 8: *Drawing only the wrinkles leads to unrealistic displacements (a). To highlight the pores, we render with additional geometry (b). To add more details, a "cushioning" effect is introduced (c).*

to standard MellowMax-function by setting $\hat{N}_f = N_f$. The final blending is given as

$$\begin{aligned} s_{\text{valley}} &= \text{mellowmax}(\{\max(0,-s_i)\}_i), \\ s_{\text{bulge}} &= \text{mellowmax}(\{\max(0,s_i)\}_i), \\ s_{\text{final}} &= s_{\text{bulge}} - s_{\text{valley}}, \end{aligned} \tag{11}$$

where s_i is the displacement of the ith fragment. More verbosely, the blending step is realized in two steps with a three-channel framebuffer to store the intermediate values. First, the wrinkle and pore geometry is rendered. The fragment shader takes the displacement strength s_i and writes $\exp(\beta\max(0,-s_i)), \exp(\beta\max(0,s_i))$ and 1 (for the count N_f) into the offscreen framebuffer with additive blending. In a post-processing step, the three channels are read and the remaining part of Eq. 10 is evaluated to produce s_{valley} and s_{bulge}. For an ablation and comparison to other blending functions, we refer to the supplementary material.

3.2.2. Pores and Perturbation

So far, only the wrinkles themselves are rendered into the displacement map. Depending on the skin region or the artistic design, a deeper indentation at the pores where the wrinkles meet is desired. To model this effect, we draw additional geometry for the pores similar to how wrinkles are drawn. Recall from Section 3 that for a given node/pore v, the incident (half-)edges are given as H_v and the per-edge strength from the simulation as $W_h, h \in H_v$. We then compute the strength of a pore w_v as

$$w_{\max} = max_{h \in H_v} W_h \ , \quad w_{\text{sum}} = \sum_{h \in H_v} W_h, \tag{12}$$

$$w_v := \max\left(w_{\max} + \alpha_{\text{dmin}}, w_{\max} + \alpha_{\text{blend}}(w_{\text{sum}} - w_{\max})\right), \tag{13}$$

where α_{blend} blends between the maximal strength of an incident edge and the sum of all incident edges and α_{dmin} requires the pores to be at least this value deeper than the surrounding edges. An example with $\alpha_{\text{dmin}} = 0.6, \alpha_{\text{blend}} = 1$ can be seen in Fig. 8b. The strength of a pore is then multiplied by the user-defined parameter $\alpha_{\text{pore-width}}$ and used as width to instantiate the per-pore geometry.

Next, we allow the pores to stretch or skew in the direction of the primary orientation with the strength α_{skew}. An example of this effect can be seen in Fig. 7b. This effect can be used to carve out oriented valleys, independent of the wrinkles. For example, the reconstruction optima shown in Fig. 13 use a large value of α_{skew}.

As a further enhancement, we introduce a "cushioning" effect where a fraction α_{cushion} of the node strength is applied back onto the wrinkles, with this effect smoothly falling off to zero towards the center of the wrinkle. This slightly breaks up the regularity of the displacements as seen in Fig. 8c with $\alpha_{\text{cushion}} = 0.3$.

For the final piece of randomness, we displace each vertex x_i of the wrinkle geometry by a small offset given a multi-scale Perlin Noise implemented using hash functions [VTP19], queried at the point along the wrinkle center line closest to x_i. We refer to Fig. 7a how this affects the generated wrinkle geometry. By choosing the frequency α_{fperturb} and strength α_{sperturb} of the Perlin Noise, different effects can be achieved as shown in Fig. 9. Similarly, we support the option to add a small amount of Perlin noise to the overall displacement map with frequency α_{fnoise} and strength α_{snoise}.

4. Editing and Fitting the Simulation Parameters

We present three different options to edit and control the simulation and rendering parameters defined above. For quick reference, all user-controllable parameters are summarized in Table 1. First,

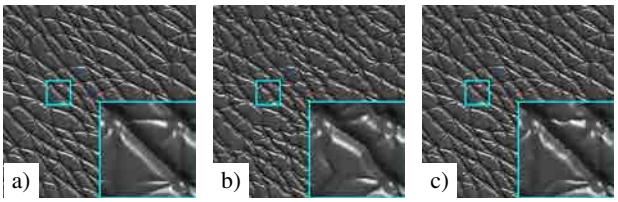

Figure 9: *Drawing wrinkles as straight lines (a) leads to a very sterile look. Adding perturbations to the wrinkle geometry breaks up the regularity. (b) shows a strong but low-frequency perturbation and (c) a weaker but high-frequency perturbation.*

if a coarse scan of the skin is available (e.g. Fig. 1b), we show that the primary wrinkle orientations and an initial set of pores can be computed from those coarse displacements (Section 4.1). Second, if sparse patches of skin displacement maps are available with micro wrinkle detail (e.g. obtained with a specialized patch scanner [GTB*13]), then optimization schemes can be used to find simulation and rendering parameters that approximate these patches for simulating over the entire surface (Section 4.2). Third, we present a collection of real-time editing tools for artists to fine-tune or design the simulation and rendering parameters from scratch (Section 4.3).

Table 1: *All user-controllable parameters per level, for the simulation (top) and drawing (bottom). The concrete values that are used in Fig. 1 and Fig. 16 are listed in the supplementary material.*

Parameter	Range	Description
α_{d}	$[0.001, 0.05]$	Average pore distance
$\alpha_{\hat{\theta}}$	$[0, 2\pi]$	Primary orientation
α_{s}	$[0, 1]$	Orientation uniformity
α_{dist}	$[0.0, 5.0]$	Distance exponent
α_{cont}	$[0.0, 0.01]$	Continuation reward
α_{sim}	$[0.0, 2.0]$	Similarity penalty
α_{cross}	$[0.0, 1.0]$	Crossing penalty
$\alpha_{deposit}$	$[0.3, 0.9]$	Deposit strength
$\alpha_{wrinkle\text{-}width}$	$[0.1, 0.6]$	Wrinkle width
$\alpha_{pore\text{-}width}$	$[0.01, 0.5]$	Pore Width
α_{dmin}	$[0.001, 1.0]$	Min. pore strength
α_{blend}	$[0.0, 1.0]$	Blending from edges to pores
α_{skew}	$[0.0, 1.0]$	Pore skew strength
$\alpha_{cushion}$	$[0.0, 1.0]$	Cushioning effect
$\alpha_{fperturb}$	$[100, 2000]$	Wrinkle pertubation frequency
$\alpha_{sperturb}$	$[0.001, 3.0]$	Wrinkle pertubation strength
α_{fnoise}	$[0.001, 0.2]$	Additive noise frequency
α_{snoise}	$[100, 3000]$	Additive noise strength
α_{scale}	$[0.1, 0.9]$	Global displacement scaling

4.1. Orientation from Coarse Observations

To extract the primary flow orientation from a given coarse displacement map, we apply the method proposed by Golovinskiy *et al.* [GMP*06]. Assuming a 4k displacement map is given from a baseline 3D reconstruction setup [RGB*20], we first extract random crops of size 128×128px. For each patch (Fig. 10a), we first compute the low frequencies using an empirically-defined Gaussian smoothing kernel of $\sigma = 1$px (Fig. 10b), and subtract these frequencies to obtain an image with only the high-frequency

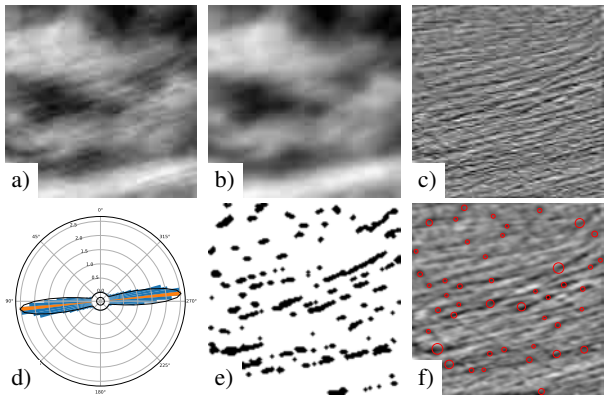

Figure 10: *Given a* 128×128 *patch of a coarse 4k displacement map (a), we first extract the low frequencies (b) and the high frequencies (c) as the difference. We can then find the primary wrinkle orientation (d) and candidate pore locations (e), which are shown superimposed on the high frequency displacement map (f).*

wrinkles (Fig. 10c). Then, we convolve this image with a collection of oriented Gabor filters and record the response per angle as a histogram (Fig. 10d). Smoothing this histogram and extracting the angle with maximal response provides the primary orientation angle $\alpha_{\hat{\theta}}$, as digested by the orientation sampler in Section 3.1.2.

Apart from the flow line orientation, extracting pores from the coarse displacement map and using them as initialization to the pore sampling (see Section 3.1.1) helps to further connect the generated micro wrinkles with the existing macro-scale features. While sophisticated pore extraction algorithms exist, especially in the context of fingerprint matching [JCD07, dAAM13], we found that a simpler, conservative algorithm suffices. First, the displacement map is high-pass filtered with $\sigma = 2$ and thresholded. Then noise is reduced by applying one morphological opening operation. The result can be seen in Fig. 10e in inverted colors. Each connected component with at least 15 pixels forms a pore with the location given by the centroid (Fig. 10f). The result of extracting spatially-varying primary orientations and pore locations from a coarse displacement map of a full face is shown in Fig. 11. Using these parameters for simulating micro wrinkles yields the final result shown in Fig. 1.

4.2. Optimization from Detailed Patches

Next, let us assume that high-resolution measured patches of the skin are available (either in addition or instead of a coarse scan as shown above). Such patches may be recovered from a specialized micro geometry scanner [GTB*13]. In this case, the simulation and rendering parameters can be optimized so that the resulting displacement map approximates the measured patches.

Optimizing for the parameters, however, poses two challenges. First, the simulation itself is not differentiable. It consists of iterative, discrete decisions on which edge to carve out to form a wrinkle. Therefore, a non-gradient-based optimization scheme is needed. Second, when creating the simulation graph for a given (optimized) pore distance, the pore and wrinkle locations will not necessarily

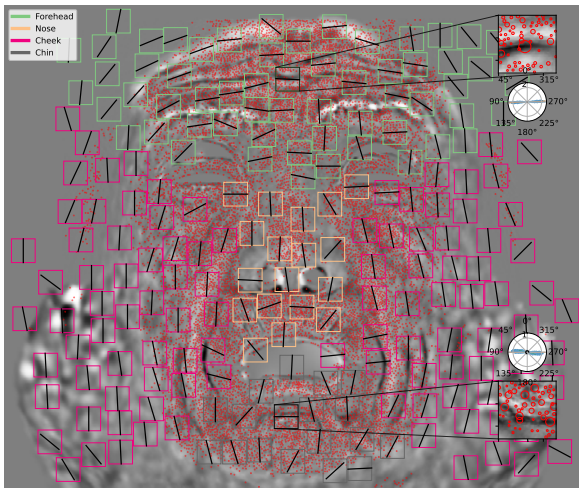

Figure 11: *Extracted pores and per-patch orientations for a coarse displacement map of a full face.*

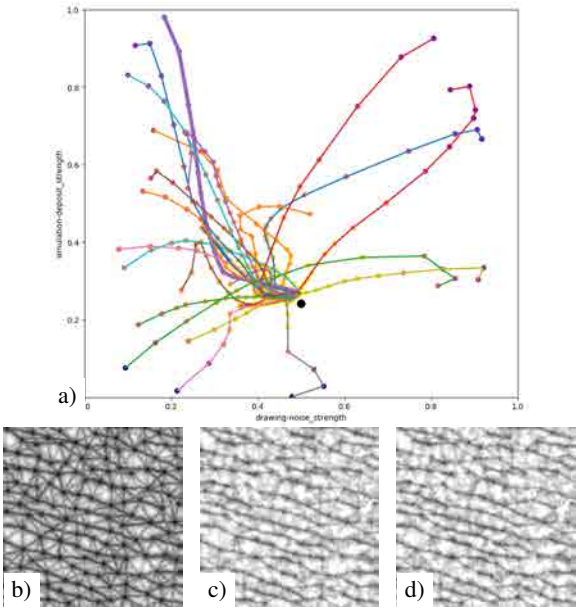

Figure 12: *Particle swarm optimization with two parameters, deposit strength $\alpha_{deposit}$ and additive Perlin noise strength. The worst initial agent (b) is highlighted in purple in (a). After 50 epochs, all agents converge to the optimum (c) that closely matches the target (d). The location of the optimum is marked with a black dot in (a).*

align with the pores and wrinkles in the observation. Therefore, a loss function that does not depend on exact pixel-wise content matches between the predicted and observed displacement map patch is required.

4.2.1. Particle Swarm Optimization

We tackle the above challenges by employing a Particle Swarm Optimization (PSO) [KE95, VSS03] together with a style-based loss function $\mathcal{L}$. Let P be the number of parameters to optimize. The basic form of a PSO follows the following principle: First, M agents are sampled. Each agent i at epoch t stores the current values of the parameters $\mathbf{x}_t^i \in \mathbb{R}^P$ and parameter velocity $\mathbf{v}_t^i \in \mathbb{R}^P$ (both randomly initialized), as well as the best parameter vector $\mathbf{p}^i \in \mathbb{R}^P$ with loss score s^i found so far. Furthermore, the global best position and score of the swarm at time t is tracked in $\mathbf{p}_t^g \in \mathbb{R}^P, s_t^g \in \mathbb{R}$. Then in each epoch, for every agent i, the current score is evaluated $s_t^i = \mathcal{L}(\mathbf{x}_t^i)$, and the per-agent best parameter $\mathbf{p}^i$ and global best parameter $\mathbf{p}_t^g$ are updated. Next, the new velocities are obtained as

$$\mathbf{v}_{t+1}^i = w\mathbf{v}_t^i + c_1 r_1(\mathbf{p}^i - \mathbf{x}_t^i) + c_2 r_2(\mathbf{p}_t^g - \mathbf{x}_t^i), \tag{14}$$

where r_1, r_2 are random numbers sampled independently and uniformly from $[0,1]$, w, c_1, c_2 are hyperparameters describing the momentum, confidence of the agent into itself, and confidence into the swarm. Given a timestep $\Delta t = 0.2$, the positions are then updated as

$$\mathbf{x}_{t+1}^i = \mathbf{x}_t^i + \Delta t \mathbf{v}_{t+1}^i. \tag{15}$$

Regarding the hyperparameters, we follow Kennedy and Eberhart [KE95] and set $c_1 = c_2 = 2.0$. For the momentum, we found that using an exponential decay [Kam09] of $w_t = 0.4 + 0.6 * 0.98^t$ provides a good trade-off between a strong initial exploration with high momentum and a better exploitation at later epochs with a low momentum. In case of cyclic parameters like the wrinkle orientation, we use the shortest distance modulo 2π for the velocity update in Eq. 14. Prior to the optimization, all parameters are normalized into the range [0,1]. To restrict the parameters to the valid range during the PSO, Helwig *et al.* [HBM13] propose various boundary handling methods. We empirically found that the best exploration and exploitation is obtained with "Random+Absorb", i.e. if an agent leaves the feasible region, its position is reset to a random value and its velocity is set to zero. Finally, instead of using a global best position "gbest" $\mathbf{p}_t^g$ for the "swarm-intelligence" term in Eq. 14, Kennedy and Mendes [KM02] propose different topologies where an agent is connected to a subset of other agents during initialization and sees only those agents for the computation of $\mathbf{p}_t^g$ throughout the optimization. Among other topologies, Kennedy and Mendes propose the "lbest" topology where the agents are connected in a ring and only see exactly two neighbors. We empirically found this topology to help the optimization as multiple local minima can be explored, compared to "gbest" or the other proposed topologies.

Because we are interested in matching the texture or "feeling" of the micro-details from the given observations, we employ a style loss on the displacement map. We utilize in our work the style loss proposed by Huang and Belongie [HB17], that is, an MSE-loss on the mean and variance of VGG-features. Furthermore, we normalize the prediction and target displacements to the same global mean and variance prior to the loss function. For an ablation of the choice of loss function, e.g. a comparison to the style loss proposed by Gatys *et al.* [GEB16], we refer to the supplementary material.

4.2.2. Optimization Results

A first demonstration of PSO can be found in Fig. 12. Here, $M = 20$ agents are optimized for 50 epochs to match the deposit strength α_{deposit} of the simulation and the strength of additive Perlin noise

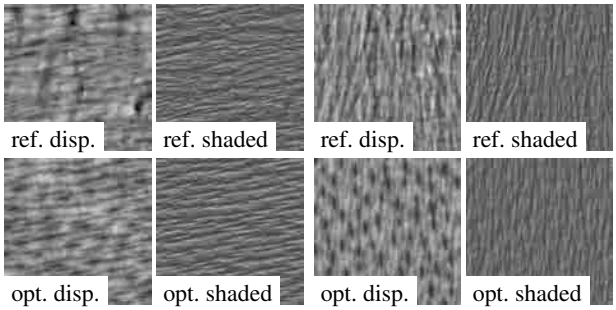

Figure 13: *Particle swarm optimizations using two scanned displacement maps provided by Graham* et al. *[GTB*13] as targets, Subject 1 chin (left) and Subject 1 forehead (right). For each example, the reference displacement map, shaded reference, optimized displacement map and shaded optimized result is shown.*

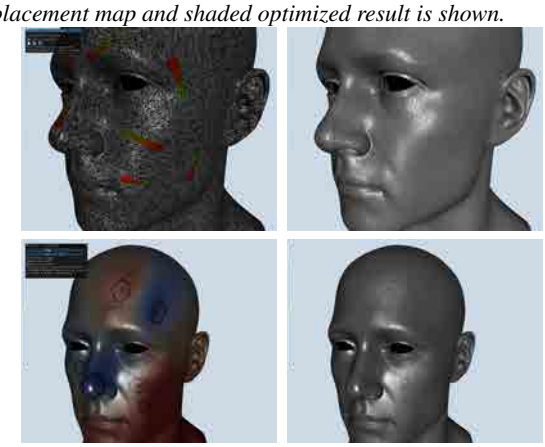

Figure 14: *Interactive artistic control for spatial-varying parameters. The left column shows the control vectors/points edited by the artist overlayed over the simulation, the right column shows only the simulation. The top row shows how the flow line vector field is designed, the bottom row how all other spatial parameters are controlled on the example of noise strength.*

α_{snoise} against a synthetic example generated with the same simulation. Even though some agents (Fig. 12b) start with a state that is far away from the optimum, all agents converge to a solution that is close to the reference.

Next, we show how the particle swarm optimization can be used to match the parameters of the simulation based on real-world measurements. For that purpose, we take the 15 displacement map patches provided by Graham *et al.* [GTB*13], obtained from 5 different regions on the face (forehead, temple, cheek, nose, chin) for three subjects, and use them as targets for our PSO optimization. Each run uses two levels of the simulation where the average pore distance α_{d} of the first level is optimized and the second level uses half the distance. Additionally, the following parameters are optimized: per level $\alpha_{\hat{\theta}}, \alpha_s, \alpha_{\text{cont}}, \alpha_{\text{deposit}}, \alpha_{\text{dist}}, \alpha_{\text{cross}}, \alpha_{\text{wrinkle-width}}, \alpha_{\text{pore-width}}, \alpha_{\text{blend}}, \alpha_{\text{skew}}, \alpha_{\text{cushion}}, \alpha_{\text{sperturb}}$, as well as one additional Perlin noise level $\alpha_{\text{fnoise}}, \alpha_{\text{snoise}}$ (refer to Table 1). Per run, 50 agents are instantiated, 100 epochs of the PSO are performed where each epoch and agent evaluates each level of the simulation for 50 steps, for a total runtime of around 35 minutes on an NVIDIA RTX A6000. To compensate for issues with random initialization, we run three trials with different seeds and choose the best trial based on the loss function score.

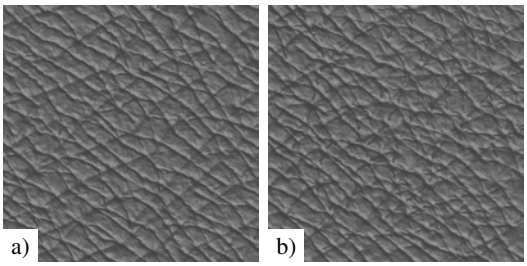

Figure 15: *Keeping all simulation parameters constant over a patch can lead to a repetitive appearance (a). Adding a small Perlin noise to the simulation parameters breaks up the regularity (b).*

The result on two selected displacement maps are shown in Fig. 13. As one can see, despite the displacement maps not matching perfectly, the texture after shading is captured very well. We attribute this to low-frequency features in the displacement map that cannot be reconstructed in the simulation, but have little influence on the shaded result and are also ignored by the style loss function. Since we are mostly interested in the look after shading, the natural next question would be apply the loss function on the shaded textures instead of the raw displacement map. However, we found that this does not increase the reconstruction quality and only adds additional variance due to the needed choice of lighting. Further results can be found in the supplementary material.

While, the optimization manages to capture the rough feel of the target textures, it struggles with the irregularities in natural skin. This is a limitation of the graph-based simulation with per-patch constant parameters and is addressed in Section 4.3. Furthermore, the PSO is still prone to getting stuck in local minima and unable to estimate a good set of simulation parameters for some tested examples. In the future, we will investigate ways to help guide the optimization and make it more robust.

4.3. Artistic Controls

As the final control application, we present interactive tools for artists to edit the parameters. In the simulation, all continuous-valued parameters (Table 1) can be made spatial-varying parameters by specifying them not as a scalar value, but rather as a 2D texture. As the simulation runs in realtime, we expose an interactive UI to the artist where they can sketch the spatial-varying parameters using control points directly on the mesh and inspect the result on the render. Two examples are shown in Fig. 14. First, the vector field of the facial flow lines is designed to create a "swirl" in the wrinkles of the cheek. Second, the noise strength is adapted to add more noise in the cheek and parts of the forehead and less on the nose. While the current sketching tools are simplistic, more advanced interaction methods will be explored in future [ZHT07, VCD*16]. For parameters that are not spatially-varying, conventional slider controls allow artists to tune the parameters globally over the whole

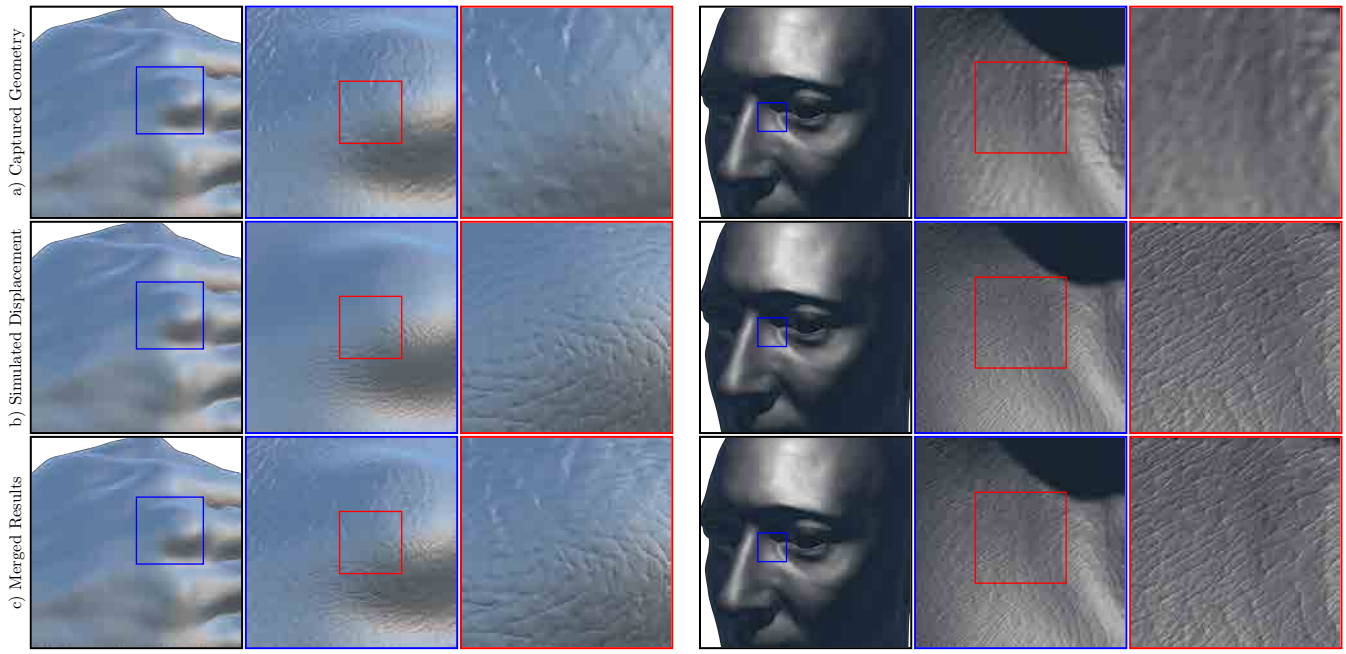

Figure 16: *Examples of enhancing the captured geometry of a hand or face using our simulated wrinkles. First row: the geometry obtained from an existing reconstruction setup [RGB*20]. Second row: the displacements from the proposed simulation. Third row: the combined displacement map. To generate the images, wrinkle geometry was written to a 16k displacement map and rendered with Blender.*

mesh. Please refer to the supplemental video for a real-time demo of parameter tuning by an artist.

Finally, one limitation of the simulation is that it can look repetitive, especially for small-scale wrinkles, if the parameters of the simulation are kept constant (Fig. 15a). To combat this “sterile" look, we allow artists to dial in a spatially-varying Perlin noise function to user-selected parameters, in order to break up the repetitive pattern. An example with added variation in three parameters, α_d (average node distance), α_{scale}, and α_{cont}, is shown in Fig. 15b.

5. Results

We have demonstrated various results of our graph-based simulation and parameter control methods in Section 3 and Section 4, respectively. We now show the full pipeline on two additional examples: a human hand and another human face (Fig. 16). First, the base geometry and displacement maps at 4k resolution are obtained with a multi-view reconstruction method [RGB*20]. Then, wrinkle simulation is performed with three scales of decreasing pore distance. The first level uses the pores and directions extracted from the base displacement map (see Section 4.1). For the other levels, the flow direction was manually edited (see Section 4.3). We used the particle swarm optimization (see Section 4.2) to inform default values of the simulation and rendering parameters, which are then further modified artistically to achieve the desired look. The micro wrinkles generate by the simulation are shown on the base geometry in Fig. 16b. For the example of the human face, a total of approximately 2 million pores and 10 million wrinkles were simulated. Note that the proposed simulation is targeted only for the micro wrinkles, and not macro-scale details. We therefore blend the base displacements containing the macro-scale features and the generated micro-scale features together. First, the fine noise in the base displacements are filtered out using a Gaussian kernel of half-size 5px and then the simulated wrinkles are added. The result can be seen in Fig. 16c. This allows us to preserve the macro-scale features (blue crops) while introducing new fine-scale features (red crops). We refer to the supplemental document and accompanying video for additional results and ablation studies.

6. Conclusion

We have proposed a novel procedural method to generate skin micro wrinkles using a graph-based simulation. First, skin pores are sampled, defining the nodes in the graph, and they are connected to their nearest neighbors. The resulting edges are then iteratively visited to carve out the wrinkles, inspired by pedestrian trail formation. Finally, geometry is instantiated for every pore and wrinkle and rendered into a 16k displacement map. The whole simulation pipeline allows for interactive editing and visualization of all involved parameters and can handle millions of pores and wrinkles. We further show, how to control the simulation parameters and integrate our work into existing 3D scanning applications. For that purpose, we presented a pipeline to estimate the initial pores and orientations from baseline reconstruction setups. If high-resolution skin patches are available, we use a particle swarm optimization to estimate simulation parameters that lead to a similar skin texture. Last, we introduced intuitive tools for artists to edit the parameters interactively. At the moment, we deal with a static facial expression and a static displacement map. A possible direction for future work would be to model the dynamic behavior of wrinkles under facial animation directly in the simulation.

References

[AL17] ASADI K., LITTMAN M. L.: An alternative softmax operator for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning* (2017), vol. 70 of *Proceedings of Machine Learning Research*, pp. 243–252. 5

[AUM76] AZUMI Y., UENO T., MOTEGI K.: Cleavage lines in the facial skin of japanese cadavers. *Bull. Tokyo Med. Dent. Univ. 23*, 2 (1976), 93–100. 4

[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 Transactions on Graphics 26*, 3 (2007). 2

[BBB*10] BEELER T., BICKEL B., BEARDSLEY P., SUMNER B., GROSS M.: High-quality single-shot capture of facial geometry. *ACM Transactions on Graphics 29*, 4 (2010). 2

[BKN02] BANDO Y., KURATATE T., NISHITA T.: A simple method for modeling wrinkles on human skin. In *10th Pacific Conference on Computer Graphics and Applications* (2002), pp. 166–175. 3, 5

[BL11] BAZIN R., LÉVÊQUE J. L.: Longitudinal study of skin aging: from microrelief to wrinkles. *Skin Research & Technology 17*, 2 (2011), 135–140. 2, 3

[BMH*16] BALU M., MIKAMI H., HOU J., POTMA E. O., TROMBERG B. J.: Rapid mesoscale multiphoton microscopy of human skin. *Biomed. Opt. Express 7*, 11 (2016), 4375–4387. 3

[BWWM10] BOWERS J., WANG R., WEI L.-Y., MALETZ D.: Parallel poisson disk sampling with spectrum analysis on surfaces. *ACM Transactions on Graphics 29*, 6 (2010), 1–10. 4

[CBZB15] CAO C., BRADLEY D., ZHOU K., BEELER T.: Real-time high-fidelity facial performance capture. *ACM Transactions on Graphics 34*, 4 (2015), 1–9. 2

[CCL84] CORCUFF P., CHATENAY F., LEVEQUE J. L.: A fully automated system to study skin surface patterns. *International Journal of Cosmetic Science 6*, 4 (1984), 167–176. 2, 3

[CWS*20] CAO X., WAECHTER M., SHI B., GAO Y., ZHENG B., MATSUSHITA Y.: Stereoscopic flash and No-Flash photography for shape and albedo recovery. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 3427–3436. 2

[dAAM13] DE ASSIS ANGELONI M., MARANA A. N.: Improving the ridge based fingerprint recognition method using sweat pores. In *Proceedings of the Seventh International Conference on Digital Society* (2013). 7

[DTA*21] DIB A., THEBAULT C., AHN J., GOSSELIN P., THEOBALT C., CHEVALLIER L.: Towards high fidelity monocular face reconstruction with rich reflectance using self-supervised learning and ray tracing. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2021). 2

[DYX*19] DENG Y., YANG J., XU S., CHEN D., JIA Y., TONG X.: Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *IEEE Computer Vision and Pattern Recognition Workshops* (2019). 2

[EF01] EFROS A. A., FREEMAN W. T.: Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), SIGGRAPH '01, Association for Computing Machinery, pp. 341–346. 3

[FFBB21] FENG Y., FENG H., BLACK M. J., BOLKART T.: Learning an animatable detailed 3D face model from in-the-wild images. *ACM Transactions on Graphics, (Proc. SIGGRAPH) 40*, 8 (2021). 2

[FM10] FLYNN C., MCCORMACK B. A. O.: Simulating the wrinkling and aging of skin with a multi-layer finite element model. *Journal of Biomechanics 43*, 3 (2010), 442–448. 3

[GEB16] GATYS L. A., ECKER A. S., BETHGE M.: Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016). 8

[GFT*11] GHOSH A., FYFFE G., TUNWATTANAPONG B., BUSCH J., YU X., DEBEVEC P.: Multiview face capture using polarized spherical gradient illumination. In *Proceedings of the 2011 SIGGRAPH Asia Conference* (2011), no. 129, Association for Computing Machinery, pp. 1–10. 2

[GMP*06] GOLOVINSKIY A., MATUSIK W., PFISTER H., RUSINKIEWICZ S., FUNKHOUSER T.: A statistical model for synthesis of detailed facial geometry. *ACM Transactions on Graphics 25*, 3 (2006), 1025–1034. 7

[GPKo19] GECER, PLOUMPIS, KOTSIA, OTHERS: Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019). 2

[GRB*18] GOTARDO P., RIVIERE J., BRADLEY D., GHOSH A., BEELER T.: Practical dynamic facial appearance modeling and acquisition. *ACM Transactions on Graphics 37*, 6 (2018), 1–13. 2

[GTB*13] GRAHAM P., TUNWATTANAPONG B., BUSCH J., YU X., JONES A., DEBEVEC P., GHOSH A.: Measurement-based synthesis of facial microgeometry. In *Computer Graphics Forum* (2013), vol. 32, pp. 335–344. 2, 3, 7, 9

[GZY*20] GUO J., ZHU X., YANG Y., YANG F., LEI Z., LI S. Z.: Towards fast, accurate and stable 3d dense face alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2020). 2

[HB17] HUANG X., BELONGIE S.: Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2017). 8

[HBM13] HELWIG S., BRANKE J., MOSTAGHIM S.: Experimental analysis of bound handling techniques in particle swarm optimization. *IEEE Trans. Evol. Comput. 17*, 2 (2013), 259–271. 8

[HCS*18] HUYNH L., CHEN W., SAITO S., XING J., NAGANO K., JONES A., DEBEVEC P., LI H.: Mesoscopic facial geometry inference using deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018). 3

[HGE01] HARO A., GUENTER B., ESSA I.: Real-time, Photo-Realistic, physically based rendering of fine scale human skin structure. In *Rendering Techniques 2001* (2001), pp. 53–62. 2

[HJO*01] HERTZMANN A., JACOBS C. E., OLIVER N., CURLESS B., SALESIN D. H.: Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), SIGGRAPH '01, pp. 327–340. 3

[HKM97] HELBING D., KELTSCH J., MOLNÁR P.: Modelling the evolution of human trail systems. *Nature 388*, 6637 (1997), 47–50. 5

[HMFB01] HELBING D., MOLNÁR P., FARKAS I. J., BOLAY K.: Self-Organizing pedestrian movement. *Environment and Planning B: Planning and Design 28*, 3 (2001), 361–383. 5

[HMS98] HELBING D., MOLNAR P., SCHWEITZER F.: Computer simulations of pedestrian dynamics and trail formation. *arXiv preprint* (1998). arXiv:cond-mat/9805074. 5

[IYYT93] ISHII T., YASUDA T., YOKOI S., TORIWAKI J.-I.: A generation model for human skin texture. In *Communicating with Virtual Worlds* (1993), pp. 139–150. 3

[JCD07] JAIN A. K., CHEN Y., DEMIRKUS M.: Pores and ridges: high-resolution fingerprint matching using level 3 features. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 1 (2007), 15–27. 7

[Kam09] KAMEYAMA K.: Particle swarm optimization – a survey. *IEICE Transactions on Information and Systems 92*, 7 (2009), 1354–1361. 8

[KE95] KENNEDY J., EBERHART R.: Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks* (1995), vol. 4, pp. 1942–1948 vol.4. 8

[KM02] KENNEDY J., MENDES R.: Population structure and particle swarm performance. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC)* (2002), vol. 2, pp. 1671–1676. 8

[Lan61] LANGER K.: Zur Anatomie und Physiologie der Haut. I. Über die Spaltbarkeit der Cutis. *Sitzungsbericht der Mathematisch-naturwissenschaftlichen Classe der Kaiserlichen Academie der Wissenschaften 44* (1861), 19–46. 4

[LC93] LEVEQUE J. L., CORCUFF P.: The surface of the skin — the microrelief. In *Noninvasive Methods for the Quantification of Skin Functions: An Update on Methodology and Clinical Applications*. 1993, pp. 3–24. 2, 3

[LK18] LIMBERT G., KUHL E.: On skin microrelief and the emergence of expression micro-wrinkles. *Soft Matter 14*, 8 (2018), 1292–1300. 3

[LLK*22] LATTAS A., LIN Y., KANNAN J., OZTURK E., FILIPI L., GUARNERA G. C., CHAWLA G., GHOSH A.: Desktop-based High-Quality facial capture for everyone. In *ACM SIGGRAPH 2022 Talks* (2022), no. 15, pp. 1–2. 2

[LLLC11] 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. *Compututers & Graphics 35*, 1 (2011), 175–184. 3

[LWSF10] LI H., WEI L.-Y., SANDER P. V., FU C.-W.: Anisotropic blue noise sampling. In *ACM SIGGRAPH Asia* (2010), no. 167, pp. 1–12. 4

[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 *Fourth International Conference on Image and Graphics (ICIG 2007)* (2007), pp. 933–937. 3

[NFA*15] NAGANO K., FYFFE G., ALEXANDER O., BARBIČ J., LI H., GHOSH A., DEBEVEC P.: Skin microstructure deformation with displacement map convolution. *ACM Transactions on Graphics 34*, 4 (2015), 1–10. 3

[NHRD90] NAHAS M., HUITRIC H., RIOUX M., DOMEY J.: Facial image synthesis using skin texture recording. *The Visual Computer 6*, 6 (1990), 337–343. 2

[NLG22] NOGUE E., LIN Y., GHOSH A.: Polarization-imaging surface reflectometry using near-field display. In *Eurographics Symposium on Rendering* (2022). 2

[PCF05] PATERSON J. A., CLAUS D., FITZGIBBON A. W.: BRDF and geometry capture from extended inhomogeneous samples using flash photography. *Compututer Graphics Forum 24*, 3 (2005), 383–391. 2

[RGB*20] RIVIERE J., GOTARDO P., BRADLEY D., GHOSH A., BEELER T.: Single-shot high-quality facial geometry and skin appearance capture. *ACM Transactions on Graphics 39*, 4 (2020), 81:1–81:12. 1, 2, 7, 10

[RRFG17] RIVIERE J., RESHETOUSKI I., FILIPI L., GHOSH A.: Polarization imaging reflectometry in the wild. *ACM Transactions on Graphics 36*, 6 (2017), 1–14. 2

[SWH*17] SAITO, WEI, HU, NAGANO, OTHERS: Photorealistic facial texture inference using deep neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 3

[TZK*17] TEWARI A., ZOLLÖFER M., KIM H., GARRIDO P., BERNARD F., PEREZ P., CHRISTIAN T.: MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In *The IEEE International Conference on Computer Vision (ICCV)* (2017). 2

[VB18] VANDERFEESTEN R., BIKKER J.: Example-Based skin wrinkle displacement maps. In *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)* (2018), pp. 212–219. 3

[VCD*16] VAXMAN A., CAMPEN M., DIAMANTI O., PANOZZO D., BOMMES D., HILDEBRANDT K., BEN-CHEN M.: Directional field synthesis, design, and processing. *Compututer Graphics Forum 35*, 2 (May 2016), 545–572. 9

[VSS03] VENTER G., SOBIESZCZANSKI-SOBIESKI J.: Particle swarm optimization. *AIAA Journal 41*, 8 (2003), 1583–1589. 8

[VTP19] VALDENEGRO-TORO M., PINCHEIRA H.: Implementing noise with hash functions for graphics processing units. *arXiv preprint* (2019). arXiv:1903.12270. 6

[WBH*22] WOOD E., BALTRUSAITIS T., HEWITT C., JOHNSON M., SHEN J., MILOSAVLJEVIC N., WILDE D., GARBIN S., RAMAN C., SHOTTON J., SHARP T., STOJILJKOVIC I., CASHMAN T., VALENTIN J.: 3d face reconstruction with dense landmarks. *European Conference on Computer Vision* (2022). 2

[WKT96] WU Y., KALRA P., THALMANN N. M.: Simulation of static and dynamic wrinkles of skin. In *Proceedings Computer Animation* (1996), pp. 90–97. 3

[WL00] WEI L.-Y., LEVOY M.: Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques (SIGGRAPH)* (2000), pp. 479–488. 3

[YSN*18] YAMAGUCHI S., SAITO S., NAGANO K., ZHAO Y., CHEN W., OLSZEWSKI K., MORISHIMA S., LI H.: High-fidelity facial reflectance and geometry inference from an unconstrained image. *ACM Transactions on Graphics 37*, 4 (2018), 1–14. 3

[ZHT07] ZHANG E., HAYS J., TURK G.: Interactive tensor field design and visualization on surfaces. *IEEE Transactions on Visualization and Computer Graphics 13*, 1 (2007), 94–107. 9

[ZS05] ZHANG Y., SIM T.: Realistic and efficient wrinkle simulation using an anatomy-based face model with adaptive refinement. In *Proceedings of Computer Graphics International* (2005), pp. 3–10. 5