Appearance-Preserving Tactile Optimization

CHELSEA TYMMS, New York University
SIQI WANG, New York University
DENIS ZORIN, New York University

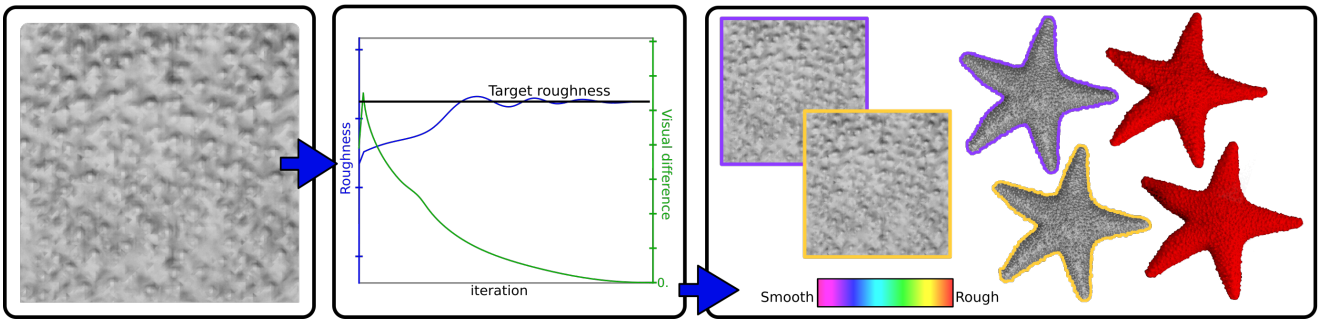


Fig. 1. Our optimization procedure enables the control of a texture's tactile roughness while maintaining its visual appearance. Starting with a target texture (left), the procedure optimizes toward a desired tactile roughness while preserving the visual appearance (center). The resulting textures can be used to fabricate visually similar but tactually different objects, such as these 3D-printed starfish (right, photographed).

Textures are encountered often on various common objects and surfaces. Many textures combine visual and tactile aspects, each serving important purposes; most obviously, a texture alters the object's appearance or tactile feeling as well as serving for visual or tactile identification and improving usability. The tactile feel and visual appearance of objects are often linked, but they may interact in unpredictable ways. Advances in high-resolution 3D printing enable highly flexible control of geometry to permit manipulation of both visual appearance and tactile properties. In this paper, we propose an optimization method to independently control the tactile properties and visual appearance of a texture. Our optimization is enabled by neural network-based models, and allows the creation of textures with a desired tactile feeling while preserving a desired visual appearance at a relatively low computational cost, for use in a variety of applications.

CCS Concepts: • **Human-centered computing** → *User studies*; • **Computing methodologies** → *Perception*.

Additional Key Words and Phrases: Roughness, fabrication, perception

Authors' addresses: Chelsea Tymms, New York University, tymms@cs.nyu.edu; Siqi Wang, New York University, siqi.wang@nyu.edu; Denis Zorin, New York University, dzorin@cs.nyu.edu.

0730-0301/2020/12-ART212 $15.00
https://doi.org/10.1145/3414685.3417857

ACM Reference Format:
Chelsea Tymms, Siqi Wang, and Denis Zorin. 2020. Appearance-Preserving Tactile Optimization. *ACM Trans. Graph.* 39, 6, Article 212 (December 2020), 16 pages. https://doi.org/10.1145/3414685.3417857

1 INTRODUCTION

Tactile textures are ubiquitous in everyday life. We encounter tactile textures on the surfaces of fruits and plants, skin, woven fabrics, and many manufactured surfaces. Tactile texture often serves a specific purpose, practical or aesthetic (an object should feel good, not just look good). Creating a particular tactile feeling is a common task which receives less attention than visual appearance, although is often just as important. Tactile feeling plays a particularly important role for people who are visually impaired, who rely on the sense of touch much more.

The tactile feeling and visual appearance of objects can interact in unpredictable ways; for example, the tactile texture may be a byproduct of creating a particular appearance (e.g., an etched pattern), or vice-versa (e.g., knurled grips have a particular look). The goals of achieving particular visual and tactile appearances may be conflicting: e.g., one may want a particular visual pattern on a tool handle, while achieving specific tactile properties optimal for usability. While in many cases, little can be done about the interaction of visual and tactile properties, advanced fabrication technologies like high-resolution 3D printing enable highly flexible control of both visual and tactile texture.

A characteristic feature of both visual and tactile textures is their statistical nature: that many distinct patterns and geometries may look or feel the same. We refer to distinct (in the sense of per-point

equality) textures that are perceived in a similar way as perceptually equivalent. The large space of perceptually equivalent textures makes it possible to adjust one aspect of a texture (e.g., tactile) without affecting the other (visual). This type of adaptation makes it possible to separate the process of visual and tactile design.

In this paper, we propose an efficient optimization method for independent control of tactile feeling and visual appearance of a surface. More precisely, the problems we solve can be formulated as follows: given input texture geometry, how can we modify it to achieve certain target tactile properties while minimizing changes to its visual appearance? And conversely, how can we achieve specific visual appearance by modifying geometry, while preserving tactile properties? Our method builds on the previous work on quantitative modeling of perceptual roughness, as well as visual appearance perception. One of the main drawbacks of the highly accurate roughness model we use is the relative expense of its evaluation and the lack of differentiability, making it difficult to apply in the optimization context. One of the main contributions of our work is efficient neural network-based differentiable versions of models for tactile roughness, visual appearance and contact area. The roughness model is in close agreement with an accurate but expensive-to-evaluate model while it also does not require expensive 3D meshing and FEM simulation and can be evaluated directly on the input texture geometry. The speedups we obtain are on the order of 10,000 times for roughness evaluation (although the original FEM model we compare to was not fully optimized), making it possible to use this model in the inner optimization loop. In addition, the resulting neural-network model provides gradients, making it trivial to plug it into an efficient optimizer.

Using the same basic approach, we also constructed a similar neural network model for contact area and for a visual similarity measure for geometric textures involving advanced lighting effects, both with multiple-orders-of magnitude speedups.

Using these models, we developed an optimization method that allows for controlling the changes in visual appearance and tactile roughness. With the same approach, it can also control another aspect of tactile perception, temperature sensation. We demonstrate the behavior of our system for a variety of examples in different contexts and validate our approach with several visual and tactile experimental studies on flat and curved surfaces.

2 RELATED WORK

Our work is related to previous work in several domains. Two of the most important works we build on are [Tymms et al. 2018] (we use the roughness model described in that paper as a starting point), and [Isola et al. 2017], which describes an image-to-image CNN that we adapt to our purposes. Our work is connected to a spectrum of work in visual and tactile perception modeling, texture synthesis and applications of CNN to optimization.

Tactile perception. Research on the sense of touch has found that tactile perception consists of 4-5 dimensions ([Tiest 2010]), including large-scale and small-scale roughness; compliance; friction; and temperature. Here we focus on large-scale roughness, elicited by features larger than 0.1 mm in size and detected through strain; we also consider temperature, controlled here by mediating the area of contact between the skin and a surface. Most previous research in roughness perception has used different types of natural or artificial stimuli that are difficult to control, e.g. [Manfredi et al. 2014], [Connor et al. 1990]. We use 3D printing to allow creation of higher-resolution, more precisely controllable surfaces. We also gain insights from [Tiest and Kappers 2009], who performed experiments on temperature perception based on the thermal diffusivity and found a relative threshold of discrimination of 43%.

Tactile fabrication. [Piovarči et al. 2016] developed a quantitative model for tactile *compliance* perception using stimuli fabricated from materials with different perceived tactile compliance, and demonstrated its applications to fabricating shapes with variable properties. Compared to roughness, compliance rarely affects the visual appearance of an object, so combining the two is relatively straightforward. In [Elkharraz et al. 2014] a roughness model was obtained using *tactile* textures fabricated from a set of *visual* textures converted to shallow height maps, implicitly creating a close connection between visual and tactile appearance. In our work, we aim to decouple these.

Other recent work in the fabrication domain has aimed to facilitate the incorporation of tactile properties in 3D printed models. [Torres et al. 2015] provides an interface to fabricate objects with a user-specified weight, compliant infill, and rough displacement map. However, their roughness metric relies on texture feature size, which is not always definable and does not provide a comprehensive model for all textures. [Chen et al. 2013] develops methods to fabricate objects with specified deformation behavior and textured surface displacement, but does not allow direct perceptual control. [Degraen et al. 2019] addresses a more specific question using 3D-printed hair structures to adequately simulate material roughness and softness for use in immersive virtual reality.

Thermal conductivity is of interest in fabrication but is typically controlled by altering the base material or creating a composite; [Wang et al. 2017] reviews several options to vary thermal conductivity and other material properties. We aim to control conductivity for tactile contexts by altering geometry. In a related application, [Zhang et al. 2017] optimizes the tessellation pattern of 3D-printed orthopedic casts for thermal comfort.

Texture synthesis. [Portilla and Simoncelli 2000] created a model for texture synthesis based on a set of image statistics. Their method performs well on some natural and artificial textures, but fails for others; it also requires a significant amount of time and is therefore poorly suited to optimization. [Wallis et al. 2017] is based on CNN feature-based model (VGG-19) but similarly does not provide a close match for many textures. Classical non-parametric texture synthesis work, e.g. [Efros and Leung 1999];[Wei and Levoy 2000], yield high-quality results for many textures, but are not readily adaptable for our optimization purposes. A recent survey of synthesis methods can be found in [Barnes and Zhang 2016]. Works such as [Gatys et al. 2015] and [Ulyanov et al. 2016] present synthesis methods based on CNNs but are not robust enough for our optimization purposes. [Zhou et al. 2018] presents a recent GAN-based texture synthesis method with impressive results, but it requires several hours of training for each image, and similarly [Yu et al. 2019] provides perceptually-based texture synthesis but requires days of training for a set of similar textures; neither is suitable for optimization in

its current form. In contrast, we seek a method that is robust for all textures, and whose loss computation does not require a large amount of time.

Optimizing fabricated visual appearance. Several works use optimization to accomplish a similar goal of appearance preservation for 3D printing. [Schüller et al. 2014] uses optimization to alter the geometry of 3D objects to maintain visual appearance subject to other geometric constraints, to produce bas-reliefs for fabrication. [Rouiller et al. 2013] designed a pipeline to optimize a 3D printed surface's microgeometry to replicate a desired BRDF. [Elek et al. 2017] employs optimization to correct for light scatter to more accurately reproduce color in 3D printing, and [Shi et al. 2018] uses optimization of the internal layer structure of color multimaterial 3D-printing to replicate the full spectrum of color of 2D art, invariant to illumination, more accurately than traditional 2D printing.

Visual similarity of images and textures. Visual similarity metrics are designed to quantify perceptual similarity, with consistency with perception measured by pairwise or three-way comparisons: if the numerical indicator of similarity for one pair of images is higher than for another, then we expect the first pair to be perceived as more different. Well-established visual metrics include those based on *structural similarity*: SSIM [Wang et al. 2004], FSIM [Zhang et al. 2011], MSSIM [Wang et al. 2003]. A different metric designed primarily for evaluation of image compression quality, and based on a complex visual system model, is found in [Mantiuk et al. 2011]. [Zhang et al. 2018] presents a metric based on deep features learned for, e.g., a classification task and combined with a simple metric in the feature space. These metrics were demonstrated to be closer (on relevant datasets) to human perception compared to SSIM. We use a simple, tighter metric based on surface normals discussed in Section 3. We discuss our experiments with other measures there. This is consistent with some of the work on depth images, e.g., [Haefner et al. 2018] a method for increasing resolution of depth images using an additional color channel, uses a metric including estimation of the normal difference. [Martín et al. 2019] develops a procedure to measure texture similarity by matching a localization task to texture statistics; but the current implementation was not shown successful for diverse textures.

Neural networks in model reduction. Model reduction is a well-established area which was using a variety of machine learning-related techniques to decrease the number of parameters needed to simulate a physical model, with the goal of reducing the cost of the simulation, which is particularly important in optimization context. We share this motivation, although we do not aim to achieve this goal through explicitly reducing the number of parameters of the model. Older methods are relatively well-covered in the survey [Forrester and Keane 2009]. Very recently, and concurrently to this work, neural networks were applied for reduced-order modeling of Poisson and fluids in 2D [Hesthaven and Ubbiali 2018]. Other model examples are considered in [Raissi et al. 2019].

Steganography. Steganography algorithms aim to hide watermarking or other types of information in data, with a few papers focusing on 3D data; see e.g., [Wang et al. 2008] for a survey, and more recently [Yang et al. 2017]. As we do in our work, these methods aim to preserve visual appearance, but the goal is to conceal the hidden information from the naive observer; in our case, we do not want to make the modification of tactile properties apparent.

3 OVERVIEW

The main goal of this work is to develop a process to allow the control of a texture's tactile roughness or tactile temperature while maintaining its visual appearance, which can produce a range of effects.

Summary. Given an input 2D height field and a desired tactile roughness value or contact area, the model uses learned functions – one for appearance based on rendered shading, and one either for tactile roughness, based on variation of strain in simulated skin, or for tactile temperature, based on a simulated skin contact area – to perform an optimization for roughness or contact area while minimizing visual distortion. We use psychophysical experiments to validate the results. A general overview of the process is shown in Figure 1.

The development of our optimization process consists of the following steps:

- We create a set of 6300 height maps comprising a variety of textures and grayscale images. We run simulations estimating the human finger contacting these heightmaps, and find the resulting field of maximum compressive strain.
- We use a convolutional neural network to learn a function taking the input heightmap and outputting the maximum compressive strain field, and we compute tactile roughness on this field.
- We use a similar neural network to learn a function taking the input heightmap and outputting the contact area between the skin and the texture.
- We learn a function for the height field's visual appearance using a CNN to learn the rendering with shadow and lighting.
- We develop an optimization procedure taking the losses from the learned roughness or contact function and the learned rendering function to optimize for a target tactile roughness or temperature while minimizing change in appearance.
- We validate this procedure by testing several textures both as renderings and as 3D-printed textures and running human psychophysical experiments. We compare against the simpler method of altering tactile feeling using linear scaling.

4 OPTIMIZATION

The optimization procedure acts to alter the geometry of the input texture height field, in order to modify the tactile feeling of the input while minimizing its change in visual appearance.

4.1 Optimization Overview

We use three functions in our optimization process to compute tactile and visual difference estimates:

- Roughness: $\phi_r : \mathbb{R}^n \to \mathbb{R}^n$, where n is the number of pixels in the height and stress maps, mapping the height field to stress magnitudes at a plane inside the skin where tactile sensors

are located. The stresses are sampled at the same resolution as the input height field.
- Visual appearance: $\phi_v : \mathbb{R}^n \rightarrow \mathbb{R}^{kn}$, mapping the height field to the pixel values of k rendered images with different lighting.
- Contact area: $\phi_c : \mathbb{R}^{2n} \rightarrow \mathbb{R}^n$, where n is the number of pixels in the height and contact maps, mapping the height field and corresponding strain field to the distance between the skin and the surface at each point.

In addition, we use a function $V : \mathbb{R}^n \rightarrow \mathbb{R}$, to evaluate the perceptual roughness estimate from the stress field $\sigma = \phi_r(H)$ of height field H.

Using these functions, which we define precisely below, our target functional is defined as follows. For a given input texture height field H_0, and target perceptual roughness r_{trg}, target contact area c_{trg} and target height range $[0, H_{trg}]$ we define the following energy terms:

(1) $E_{rough}(H) = |r_{trg} - V(\phi_r(H))|$: the difference between the current roughness and the target roughness, with the strain variation function V defined in Section 4.2.
(2) $E_{contact}(H) = |c_{trg} - A(H))|$: the difference between the current contact and the target contact, where A is the weighted contact distance function defined in section 4.3.
(3) $E_{vis}(H, H_0) = \sum \frac{1}{n} \|\phi_v(H_0) - \phi_{v_k}(H)\|_2$: the visual difference, computed as the L2-norm of the pixel-wise difference between the current rendered image and target rendered image, summed over the three different rendering conditions used.
(4) $E_{reg}(H, H_0) = \sum \frac{1}{n}(\|\Delta_x(\phi_v(H_0) - \phi_{v_k}(H))\|_2 + \|\Delta_y(\phi_v(H_0) - \phi_{v_k}(H))\|_2)$: the sum of difference variation regularization energies for all rendering conditions, where Δ_x and Δ_y are finite difference matrix operators for horizontal and vertical directions; i.e., an approximation of $\int \|\nabla(\phi_v(H_0) - \phi_v(H)\|_2 dA$.
(5) $E_{clamp}(H) = \|H - \text{clamp}_{[0,H_{trg}]}(H)\|_2^2$: the clamping energy to keep the result in the $[0, H_{trg}]$ range.

The total energy we minimize is defined as

$$E(H, H_0) = E_{rough}(H) + w_1 E_{vis}(H, H_0) + w_2 E_{reg}(H, H_0) + w_3 E_{clamp}(H) \quad (1)$$

To make the optimization of this function practical, we need to compute $E(H, H_0)$ as well as $\nabla_H E(H, H_0)$ efficiently. However, computation of E_{rough} involves a 3D finite element simulation, including 3D domain meshing and contact resolution; computation of E_{vis} requires rendering of textures with some global illumination effects.

We address both of these problems by approximating ϕ_r, ϕ_c and ϕ_v using neural networks, as these provide (a) fast evaluation of function values (b) evaluation of derivatives with respect to the input parameters. The details of the approximations are discussed below.

Convergence criteria and weight choices. The main parameter of the optimization is w_1, controlled by the user, which represents the trade-off between visual fidelity and closeness to the target roughness.

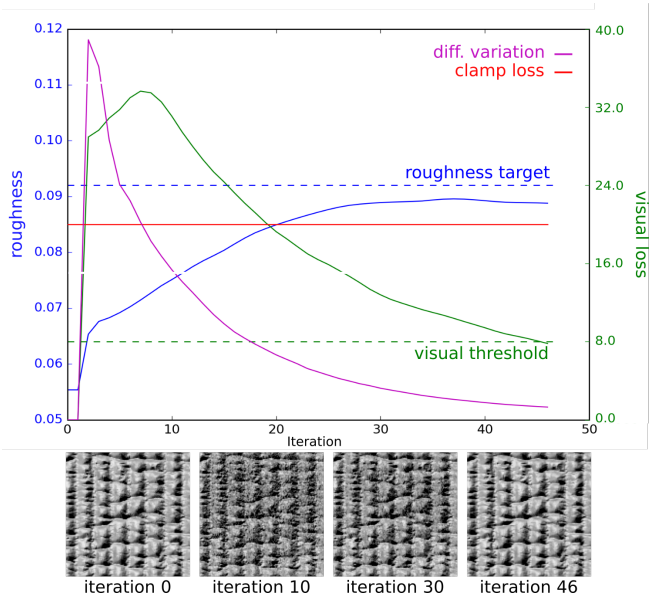


Fig. 2. Parameter convergence during optimization for roughness and visual appearance. The goal is to alter the roughness of the input texture (iteration 0) while preserving its visual appearance, which is done by the final iteration.

The weight w_3 is chosen to be relatively high, 10^5, so that the last term operates as constraint. The weight w_2 is chosen to be lower compared to w_1, as E_{reg} acts as a regularizing term, minimizing small-scale noise by picking smoother solutions among those with low values of the first two terms. We use $w_2 = 0.06$.

For contact area, which has values on the order of 100 mm^2, approximately 1000 times the typical roughness values, these weights were scaled up by 1000.

We use a stopping criteria for optimization that places bounds on three of the energy components: For roughness, $E_{rough} < \varepsilon_r r_{trg}$, with $\varepsilon_r = 0.1$, about half of the 19% threshold for tactile roughness discrimination described in [Tymms et al. 2018]. For visual difference, $E_{vis} < \varepsilon_v \|\phi_v(H_0)\|_2$, with $\varepsilon_v = 8$; this is proportional to image resolution, and was experimentally found as a conservative goal to avoid visible changes, corresponding to a 2% change in pixel values.

The height constraint is expected to be satisfied nearly precisely: $E_{clamp} < \varepsilon_c$, with $\varepsilon_c = 10^{-4}$. We used $H_{trg} = 3$, to ensure the height remain below 3 mm, selected as a reasonable maximum height for a fabricable tactile texture.

An example of the optimization process for a texture is shown in 2. The effect on convergence of using altering the weights is shown in Figure 3.

The Adam optimizer ([Kingma and Ba 2014]) implemented in Pytorch is used for optimization. A learning rate of 0.027 was chosen through trials with single parameters to permit convergence of the parameters but avoid excessive oscillation. In the next sections, we explain how the roughness, contact and visual functions, respectively ϕ_r, ϕ_c and ϕ_v, are defined.

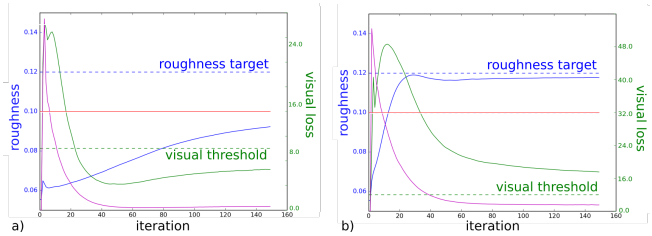


Fig. 3. a) When a significantly (10x) lower weight is used for w_1, convergence of roughness to the target may not occur. b) A significantly higher (10x) weight for w_1 causes the visual energy to converge more slowly, and it may not reach the target threshold.

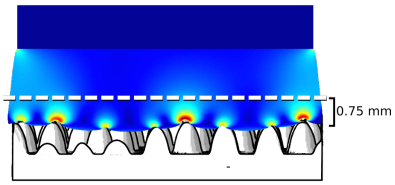


Fig. 4. In the original roughness model, a 3D FEM simulation was used to simulate the skin touching a textured surface, and the maximum compressive strain field was sampled from a depth of 0.75 mm.

4.2 Tactile roughness

We use a modified version of the model developed in [Tymms et al. 2018], which computes the tactile roughness of a surface by simulating the strain variation field resulting from skin contact on the surface. The computation of the model is relatively expensive; we briefly summarize the model here for completeness. The main step of the model is a finite element method simulation of the contact of the skin with the tactile texture defined by $H(x, y)$, to obtain a corresponding displacement field $\mathbf{u}_H(u, v, w)$, where u, v, w are 3D coordinates in the undeformed layer of skin, with $w = 0$ corresponding to the surface, and $w_0 = 0.75$ mm corresponds to the approximate depth of the tactile receptors.

To approximate the skin, we use the same two-layer block model as [Tymms et al. 2018]. The block is $1cm^2$ in surface area and $0.5cm$ in height, with a rigid upper half and soft lower half, and a force of 10 N is used. A model of the simulation is shown in Figure 4.

For a displacement field $\mathbf{u}$, $\epsilon[\mathbf{u}] = \frac{1}{2}(\nabla\mathbf{u}+\nabla^T\mathbf{u})$ is small-deformation strain tensor. If $\lambda_3(\epsilon)$ is the largest-magnitude negative (compressive) eigenvalue of the strain tensor, our perceptual roughness estimate $f(H)$ can be written as

$$f(H) = \mathrm{V}(\lambda_3(\epsilon(\mathbf{u}_H(\cdot, \cdot, w_0))))$$

where V is the strain variation function on the plane $w = w_0$. We replace a stochastic function defined in the original model with a deterministic function described in more detail below.

The expensive step is the computation of displacements $\mathbf{u}_H$ for a given H: it requires sufficiently fine 3D meshing to resolve the detail at the scale of smaller texture features, and solving a nonlinear (due to contact) constrained elastic deformation problem, which in our current implementation has a computation time of 20-40 minutes

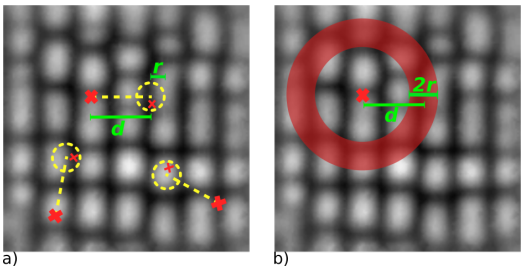


Fig. 5. a) Stochastic sampling; b) Equivalent deterministic sampling

and uses 15GB of memory when using the required highly-refined mesh. In addition to the cost of evaluation, it is difficult to obtain an approximation of the derivative of this function other than by extremely expensive finite differences, so optimizing a functional depending on the $\mathbf{u}_H$ can only be done with gradient-free methods.

This is the step that we replace with a direct map

$$\phi_r(H)(u, v) \approx \lambda_3(\epsilon(\mathbf{u}_H(u, v, w_0))),$$

represented with a neural network.

Strain variation function. In [Tymms et al. 2018], the strain variation function $V(\sigma)$ was computed using a large set S of N randomized pairs of samples (p_1, p_2), $p_i = (u_i, v_i)$, separated, on average, by a distance d, are computed. Denoting $\sigma(u, v) = \lambda_3(\epsilon[\mathbf{u}_H](u, v, w_0))$,

$$V(H) = \frac{1}{N} \sum_{(p_1,p_2)\in S} |\sigma(u_1, v_1) - \sigma(u_2, v_2)|$$

$N = 8000$ sample pairs were used, sampled from disks of radius 0.8 mm placed at the endpoints of randomly selected segments of length 2.2 mm.

Instead of using a random sampling of points, here we use a deterministic evaluation of variation between each point and its neighbors within the desired distance, in order to derive a strain variation field (Figure 5):

$$V(H) = \frac{1}{2rl} \int_{x=0}^{l} \int_{y=0}^{l} \int_{\Delta=d-r}^{d+r} \int_{\theta=0}^{\pi} |\sigma(x, y) - \sigma(x + \Delta\cos\theta, y + \Delta\cos\theta)| dx dy d\Delta d\theta \quad (2)$$

This function is smooth, so the gradient of the complete roughness estimates can be computed.

Learning the strain field. The FEM simulation used to compute $\sigma(u, v)$ in [Tymms et al. 2018] is used solely to find a single 2D strain field; that is, the simulation takes as input a 2D grid (the heightmap defining the boundary conditions for the contact area), and returns as output a 2D grid (the maximum compressive strain at a depth of 0.75 mm). Image-to-image translation problems have been studied extensively in machine learning, and here we adapt a convolutional neural network described in [Isola et al. 2017] to learn a relationship ϕ_r between the input height map and the output maximum compressive strain.

To acquire ground truth simulation data, we ran the FEM simulation for the 3D skin model using a heightmap dataset of with

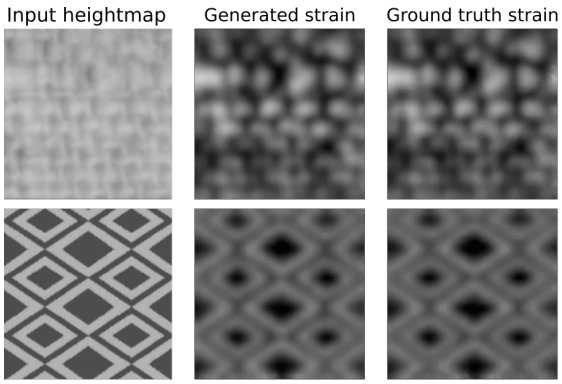


Fig. 6. Two examples from the learned CNN test set show the learned and ground-truth maximum compressive strain fields from the input heightmaps. Strain fields are shown with increased contrast for visual clarity.

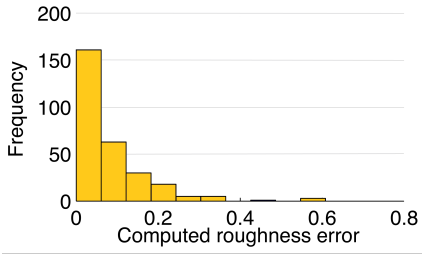


Fig. 7. The difference in computed roughness between the learned strain field and the real strain field is typically very low, with a median of 5.3%.

6307 image pairs, similar to the size of several datasets successfully trained with this neural network structure. We use a set of black and white images and textures (including the Describable Textures Dataset [Cimpoi et al. 2014], VisTeX [MITMediaLab 1995], and Brodatz texture database [Brodatz 1966]) and procedural textures to enrich the dataset. In some cases, images were randomly cropped and/or scaled, and in some cases procedural noise was added. Heightmaps had a maximum vertical height of 3 mm and represented a texture of size 100 mm^2. As suggested in [Tymms et al. 2018], for each simulation we found the maximum compressive strain field at a depth 0.75 mm, and the strain field of a flat texture simulation was subtracted to discount any effect from edges.

Inputs and outputs were scaled to 128×128px images. The set was split randomly into three sets: testing (312 images); training (4918), and validation (1077). We used the convolutional neural network used as the generator in [Isola et al. 2017], with no dropout and using BCE loss, and trained for 200 epochs with batch size 1.

The learned strain field and its resulting tactile roughness value were computed from an unseen testing set, and the learned value was compared against the actual value. The median error in roughness was 5.3%, and the average error was 8.0%, well below the perceptual threshold of 19%. These values are well below the threshold of discrimination of 19% described in [Tymms et al. 2018]. The error distribution is shown in Figure 7.

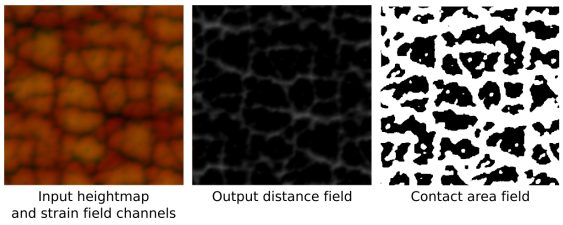


Fig. 8. The contact area function takes as input the input heightmap (left, red channel) and the strain field (left, green channel) and outputs the distance field (center, where black indicates a distance of 0). The distance field can be used to compute the contact area (right, where the contact area is black)

.

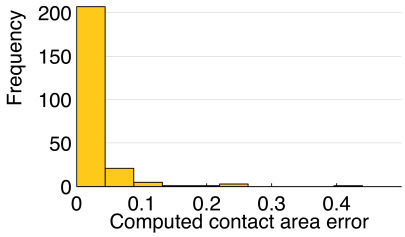


Fig. 9. The learned contact area matches the actual contact area very closely, with an error of 2.7%.

.

The network allows the roughness to be computed in an average of 0.05 seconds, a significant speedup compared to the 20-40 minutes required to run the full FEM simulation.

The learned function and its gradient are used in optimization for a texture to converge toward a desired tactile roughness, as shown in Figure 2.

4.3 Contact area

Computing the contact area requires the same time-intensive FEM simulation as computing the roughness field. To compute the contact area, we use a function taking as input the height field and outputting the field of distances between the surface and the simulated skin at each point. The computation of this distance field is expensive and requires an FEM simulation as described in section 4.2. Therefore, we replace this step with a neural network.

Learning the contact distance field. The FEM simulation takes in the input height field H and outputs a mesh displacement field $\mathbf{u}_H$ describing the displacement of the skin when in contact with height field H. From this displacement field and the height field, we can acquire the field of the distance $\mathbf{d}$ between the skin and the input texture at each point, where a distance of 0 indicates skin contact with the texture surface.

We adapt a similar convolutional neural network to learn the relationship between the input height map H and the output distance field $\mathbf{d} = \phi_c(H)$. We used the same height field training set as used previously in Section 4.2, which had about 6300 pairs. To improve the accuracy of the learned function, we also provided the strain field

as input, so that the input to the function has 2 channels of input: the height field and the strain field. An example of the function's input and output is shown in Figure 8.

To compute the error for the testing set of size 250, the learned distance field was computed for each input heightmap with its learned strain fields, and the contact area was computed and compared to the actual contact area derived from simulation. The errors in computed contact area for this set had a mean of 2.7%, as shown in Figure 9.

Contact optimization. The optimization aims to modify a texture so that its total contact area moves to a particular target. Because contact area itself is a discontinuous function, the optimization process was often unable to converge. Therefore, we use a smooth function weighting the contact area at each point proportionally to the inverse of its distance. That is, for contact distance field $\mathbf{d}$, contact area is approximated by:

$$A(H) = \int_{x=0}^{l} \int_{y=0}^{l} \frac{1}{1 + 80 * \mathbf{d}(x, y)} \tag{3}$$

This function provides a smooth weighted contact distance, so that a distance of 0 has a weight of 1; weights decay rapidly so that a distance of 0.01 mm has a weight of 0.5 and a distance of 0.1 mm has a weight of 0.1.

4.4 Visual appearance

To preserve a texture's visual appearance during optimization, we use a custom function based on visual similarity of the original height field and the optimized one. Ideally, to measure visual similarity, we would consider all possible views of a pair of textures under different lighting conditions, apply a visual difference metric between each pair, and compute an aggregate metric. We follow these steps, but use a restricted set of lighting conditions and use the simplest visual metric to compare the images. In Section 5, we validate the setup we use comparing it with a more expensive multiview optimization.

We found that to ensure realistic results some features of images used to evaluate visual similarity are critical. Specifically, we have observed that *shadows*, *ambient occlusion* and *gloss* affect visual texture perception in a critical way (Figure 10), as a texture comprises many small elements that cast shadows over the surface. For this reason, we must opt for a rendering pipeline supporting these features to generate views of the texture, rather than, e.g., approximating the texture image with the dot products of the normal with the light direction.

As discussed in Section 2, a variety of measures of visual similarity exist and are widely used. Most could be used in our context in a way similar to the function V above used for roughness; e.g., [Zhang et al. 2018] describes a perceptual measure of visual similarity represented with a neural network, that can be easily applied in our context. However, we found that in the optimization context, these measures tend to be too "permissive": while these metrics are good for measuring distance between real images, synthetic images can be far from a given image perceptually, but close in the sense of these metrics. For this reason, we opt for a relatively conservative

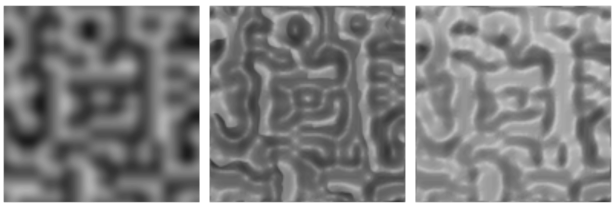


Fig. 10. A texture heightmap rendered with (center) or without (right) shadowing and ambient occlusion. Shadowing in small regions of lowered height is critical to a texture's visual appearance.

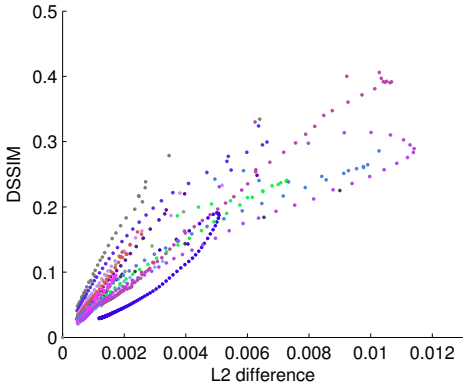


Fig. 11. Plot showing render DSSIM and L2 difference errors for a set of textures in optimization steps.

L2 norm of the difference between images. Figure 11 shows a scatter plot exhibiting that *L2* has a correlation with DSSIM.

Rendering. Heightmaps were rendered in a gray material with low specularity, similar to matte plastic, using a Phong shader. Objects were rendered with three different lighting conditions, with a single constant-direction parallel-ray light sources at an angle of 35°from the x-y plane and rotated on the z-axis 10°, 130°, or 250°. Images were rendered at 128 × 128px using Blender.

While differentiable renders have recently appeared [Li et al. 2018], given the highly restricted nature of the renderings that we need to compute (square texture samples), we opted for a similar approach as we use for the stress maps for the roughness measure. As an additional benefit, this approach also provides a gradient of the rendered image with respect to the heightfield.

For each lighting condition, we trained the generative adversarial network of [Isola et al. 2017] on a set of 4764 images, with a validation set of size 1059. The network was trained for 200 iterations. Results showed high accuracy, as seen in Figures 12 and 13. All three lighting conditions had similarly high accuracy (with mean pixel errors of 2.2%, 2.3%, and 2.4%).

The neural network offers a significant speedup to rendering: the neural network render computation time is only 0.05 seconds after the network is loaded; while traditional rendering time is approximately 15 seconds with ray-tracing shadows using Adaptive

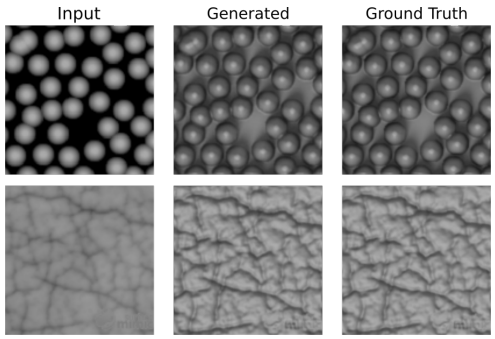


Fig. 12. Two examples from the test set for visual rendering. The learned function for the rendering of heightmaps was learned with high accuracy: in most cases generated and real renderings are visually indistinguishable.

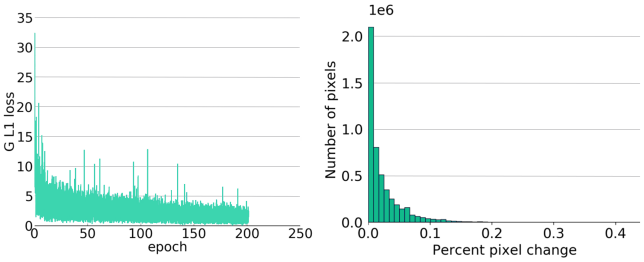


Fig. 13. Left: L1 loss convergence of the generator during training of the GAN on texture rendering for one lighting condition. Right: Histogram of the percent differences of all rendered pixel values across 300 real and generated texture pairs in the test set. Real pixel values are approximated very closely by the network, with most pixels changing by less than 5%. The mean difference is 2.3%.

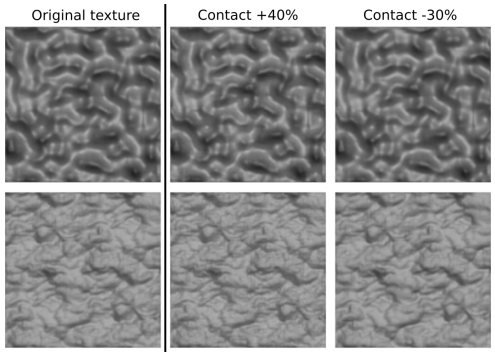


Fig. 14. Renderings of two textures optimized for different contact areas.

QMC and 20 samples for the lighting source, and ambient lighting and occlusion.

5 RESULTS

5.1 Optimization results

Figure 15 shows the results of altering the roughness of a selection of textures using our optimization for a desired tactile roughness maintaining visual appearance. Textures are rendered here using a different lighting setup than the ones used for learning. Textures shown represent $10 \times 10\,\mathrm{mm}^2$ in size.

Choosing a ground truth to compare to in our experiments is somewhat difficult, as we are not aware of any previous work on optimizing tactile properties for complex textures. We have chosen *linear scaling* as one obvious way to change geometry to increase texture roughness, while maintaining similarity to the original texture; this method was used in [Tymms et al. 2018].

On the righthand side of Figure 15, we show the results of using linear scaling of textures to achieve the desired roughness. Textures are first scaled in height up to a limit of 3 mm; then, if necessary, they are scaled in the x-y direction. The optimization-generated textures are nearly indistinguishable from the original textures, while the textures modified with linear scaling are almost always noticeably different, except in cases where the desired roughness is very close to the original roughness. Making the texture flatter or scaling it upwards results in obvious differences. Additionally, for some textures, a sufficient change in roughness is not achievable through linear scaling alone.

5.1.1 *Errors.*

Roughness. To ensure that the learned functions were robust to the types of textures generated with optimization, the errors in computed roughness were also computed for a set of 150 optimized textures, with three different target roughness values. The average error between the simulated and learning-computed roughness for the optimized texture was 8.4% with a median of 6.4%, compared to an average of 8.0% and mean of 5.3% for the overall test set.

Contact area. The same test was performed for a set of 100 textures optimized to have significantly different target contact area. Here the average error in contact area between the simulation and the learned data was 9% with a median of 4.1%. The average error for the non-optimized input set was 7.9%, with a median of 2.4%.

5.2 Evaluation and comparisons

The relationship between a surface's geometry and its tactile properties is intricate, as it depends on the difficult-to-predict way the elastic skin contacts the texture geometry. The tactile roughness is dependent on the uneven distribution of pressure resulting from that contact area. Optimization for these tactile properties while maintaining a similar appearance results in subtle changes to the texture geometry, as shown in Figure 16. It typically is not as simple as, for example, using height modification or frequency filtering: the target may be impossible to achieve, and the visual appearance may not be preserved well, as discussed below.

5.2.1 *Comparison with other methods.*

Height modification. Linear scaling is a simple method of altering a texture's roughness or contact area. If a texture is scaled up vertically, the contact area will decrease and the roughness will increase. The relative geometry is preserved, which suggests the appearance is also preserved to an extent. However, as seen in Figures 17 and 15, the appearance often cannot be preserved. In contrast, our contact

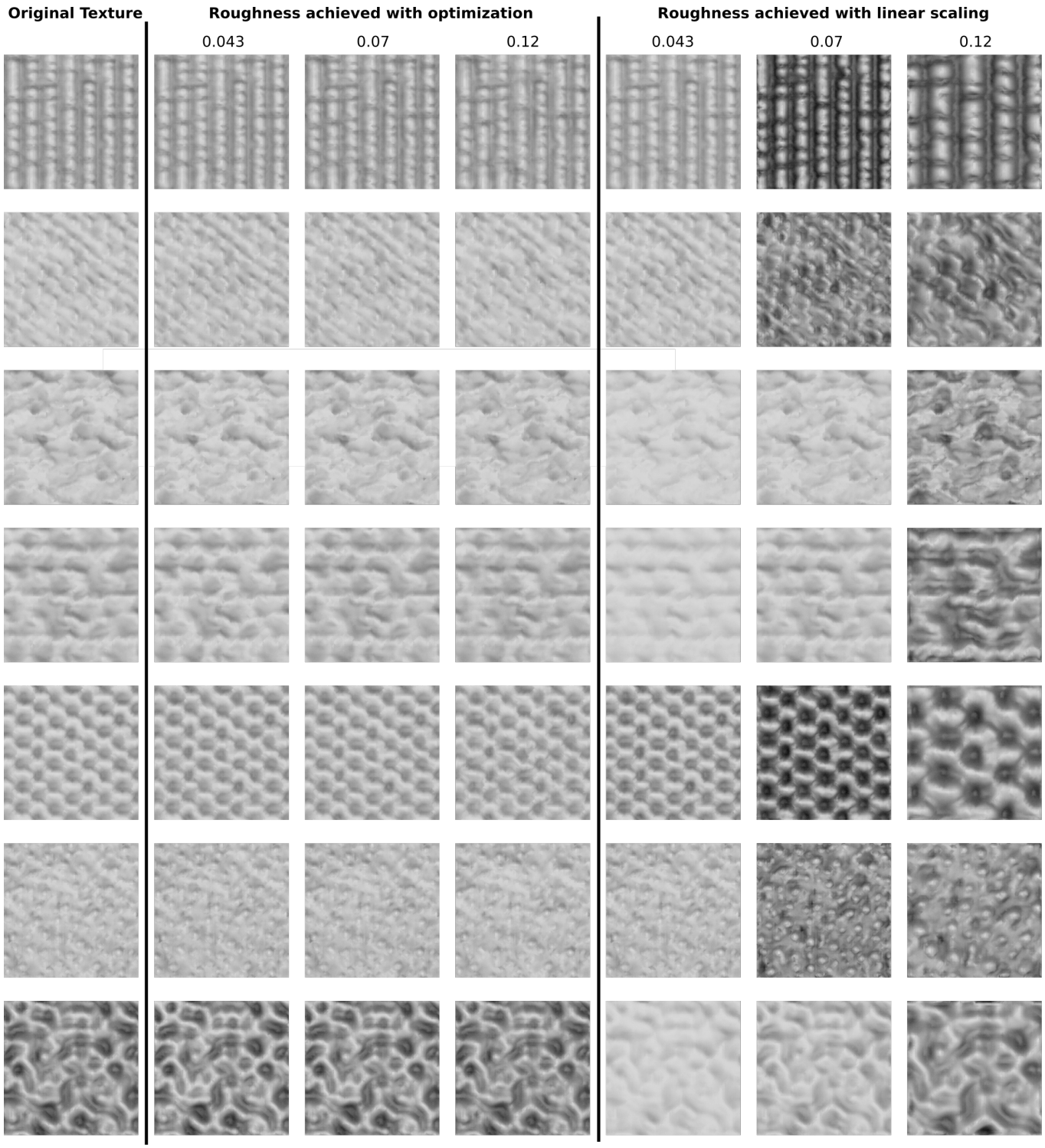


Fig. 15. Seven example textures optimized for a desired roughness. The leftmost column shows the original, target visual texture; the next three columns show the results when the roughness is achieved through our optimization process; the final three columns show the results when the same roughness is achieved through linear scaling in the z and/or xy directions. The optimization process achieves the desired roughness with nearly-imperceptible changes to the visual appearance.

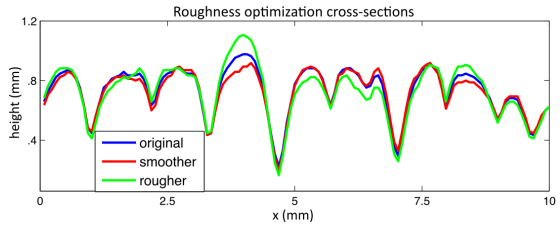


Fig. 16. Example of a texture cross-section for textures optimized for roughness. Changes to peaks and troughs are not easily predictable.

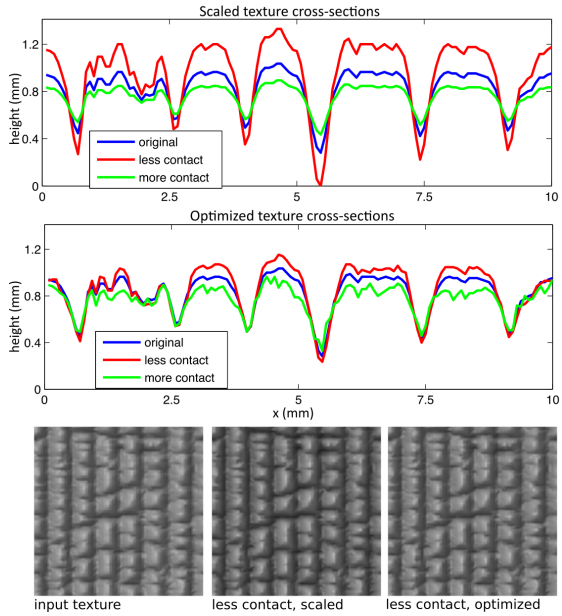


Fig. 17. Comparison of textures optimized or vertically scaled to alter contact area. The optimization results in smaller changes to the geometry and better preservation of visual appearance.

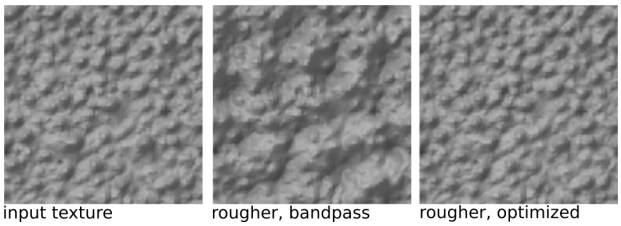
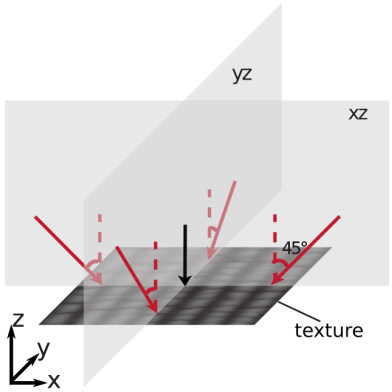


Fig. 18. Comparison of modifying a texture's roughness by modifying the dominant frequency of the texture, and using the optimization process. Modifying the frequency adds large-scale noise to the geometry, which is clearly visible on the texture.

and roughness optimizations alter the geometry in precise and small ways to change the contact while preserving appearance.

Frequency modification. Another intuitive method of altering a texture's roughness is to use bandpass filtering, altering the texture



	top only	all five	top & xz 45°	top & yz 45°
top only				
all five	2.09%			
top & xz 45°	1.88%	1.43%		
top & yz 45°	1.79%	1.29%	1.71%	

Fig. 19. Top: We tested the optimization with the addition of more points of view, 45° from vertical on the yz and xz planes, along with the original single top point of view. Bottom: The table shows the percent pixel difference between the optimization height map results, for the original top point of view, one additional point of view, or all five points of view. Only small changes occur when one or more additional points of view are added.

in the frequency domain to reduce or amplify certain frequencies. Literature and psychophysics studies suggest that roughness perception is highest when features are spaced at a wavelength of 2-3 mm apart [Hollins and Bensmaïa 2007].

However, we have observed that modifying a texture to alter the frequencies in that range does not alter the roughness is a reliable manner for all textures. For example, if the frequency is increased but the amplified areas are not contacted by the skin, the roughness will not be affected. More importantly, modifying the frequencies does not guarantee preservation of visual appearance. Figure 18 shows an example of modifying a texture's roughness by 4 just-noticeable-difference (JND) thresholds by increasing the geometry's frequencies in the 2.5mm wavelength range. The large-scale noise added to the geometry to achieve the target roughness is visible in the texture. Our optimization produces smaller changes in the geometry that are not easily apparent.

Other filtering methods. Contact area and tactile roughness depend on the way the elastic skin conforms around the geometry, which changes in nontrivial ways when the geometry is e.g. smoothed using a filter. For example, smoothing a sharp peak results in increased contact area and decreased height, which decreases roughness; but smoothing a round peak results in decreased contact area and therefore may increase perceived roughness.

5.2.2 Alternate points of view. Our visual optimization used a single, top point of view and multiple lighting conditions. To determine the value of utilizing additional view directions, we ran the optimization with four additional points of view, 45° from the vertical direction at four angles (Figure 19, top) and with the same three lighting

conditions. As with the top view, additional neural networks (one per view) were successfully trained to produce the rendered image from the specified angle and light source. Rendered images were rescaled to 128x128 pixels. The roughness optimization was run with visual weight assigned to the new points of view: the single additional point of view was weighted 40%; and when using all five points of view, 40% was given to the four new points of view. The remainder of the visual weight was given to the top view direction. We choose a higher weight for the top view to reflect higher importance of direct viewing in surface perception. As the sensitivity of the results to the addition of new points of view was shown to be low, we did not explore other options for weight allocation further.

We evaluated the results of the new optimizations against each other and the results from the previous optimization, using pairwise comparisons. As shown in the table in Figure 19, the optimized heightmap did not change significantly when new points of view were incorporated: the image pixel difference between the results differed by less than 2.1%. For this reason, we determined that using a single top viewpoint in the visual difference functional is an adequate choice: this agrees with the intuition that views of a texture from different directions are highly correlated from a broad range of angles.

5.3 Visual Experiments

In a set of visual psychophysics user studies, we tested the accuracy of our visual optimization by comparing the source texture appearance to the optimized texture appearance and a simple baseline method. For the baseline, we used a version of the texture scaled linearly in the direction perpendicular to the surface to achieve the same roughness. We also tested the validity of our single-view formulation by comparing it with a more expensive multiple-view formulation.

5.3.1 Stimuli. Six source textures were tested (shown in Figure 20). Source textures comprised different types of natural and manufactured textures and had different base tactile roughness values.

For each source texture, four target roughness values were selected, and textures were optimized to achieve those four roughness values. Additionally, successive linear scaling was used to create alternate textures with the same four target roughnesses.

For each source textures, a set of eleven 25mm square textured stimuli plates were 3D-printed using a B9Creator DLP stereolithography printer, at 50 µm resolution. Three of the textures were derived from different patches of the source textures; four were the optimized textures; and four were linearly scaled textures.

As we have used B9 Black resin to yield the most accurate geometric results, to improve visibility, textures were spray-painted with matte gray primer (Rust-Oleum Flat Gray Primer) and a coat of clear matte varnish.

5.3.2 Experiments. In each trial, two textures were placed in a case that slides beneath a circular window, through which one of the textures could be seen. Observers viewed the textures overhead at a distance of 40 cm, viewing through a mirror placed at an angle of 45° as seen in Figure 21.

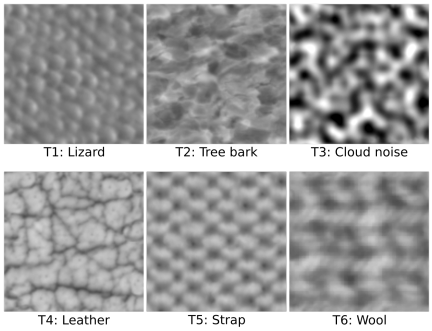


Fig. 20. Heightmaps of the six textures used in visual experiments.

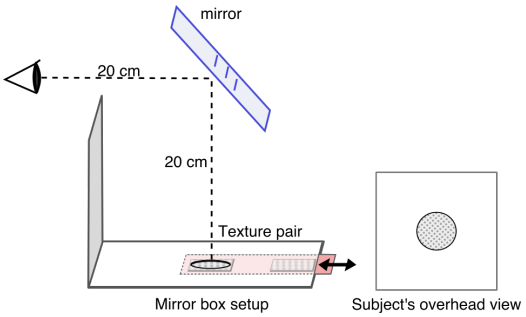


Fig. 21. The experimental setup for visual experiments, which allows the subject to comfortably view the three trial textures from an overhead view.

During each trial, observers were presented with two different textured surfaces sequentially. One of the pair was derived from the original source texture, and the other could be either another patch of the source texture; a version scaled to a different roughness using linear scaling; or a version scaled to a different roughness using our optimization process. Observers were tasked to choose whether the two textures appeared the same (i.e., derived from the same texture source) or different. Locations of the pair of textures were switched with equal probability. Observers were given 4 seconds to view the textures two times each.

Trials were presented in a pseudorandom ordering, with the constraint that trials using the same source texture were separated by at least two trials.

Six subjects took part in the experiments and performed four repetitions per texture pair. Experiments took place in an office setting with ambient fluorescent lighting.

Experiment results. Our experiment results showed that the optimization process performed substantially better than linear scaling. Figure 22 shows the proportions for the 48 test textures. The dotted black line on each graph shows the threshold at which subjects judged the reference textures from the same patch as similar to each other. Of the 24 optimized textures tested, 20 of them were judged the same as the source at least 50% of the time. In contrast, only 4 of the linearly scaled textures were judged the same as the source at least 50% of the time. In fact, half of the linearly scaled textures were judged different from the reference textures over 90% of the

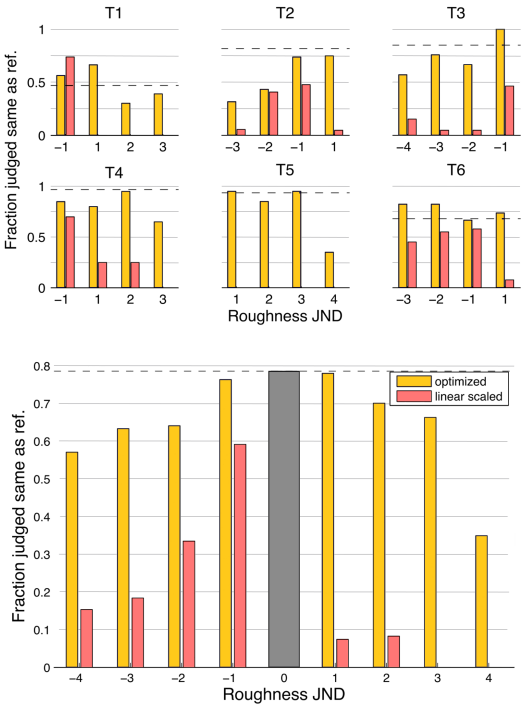


Fig. 22. Top panel: results from the experiments for each of our six textures. Bottom panel: Proportions for all textures accumulated by JND distance from the reference texture. The x-axis for each graph shows the distance in just-noticeable-differences in roughness values between the test texture and the reference, and the y-axis shows the proportion judged the same. The dotted black line shows the reference threshold at which the reference textures were judged the same as each other.

time throughout all trials. 23 of 24 of the optimized textures were judged more similar than the non-optimized version. The other one was derived from T1, a texture which was had high sensitivity to small changes, as shown by the fact that only 50% of textures from the same source were judged the same.

The bottom panel of Figure 22 shows textures accumulated by JND threshold distance from the reference texture, according to the difference threshold of 19% found in [Tymms et al. 2017]. In all cases, the optimized textures match the references better than the linearly scaled textures. In general, linear scaling tends to perform more successfully for small decreases in roughness, but performs poorly for larger decreases or increases in roughness. Optimization creates textures that appear very similar for small differences in roughness; the visual difference is only visible when the target roughness is much larger.

5.3.3 Alternate points of view. To validate the visual similarity for different points of view and complex geometry, a subset of three textures was chosen for a less restricted version of the experiments. In these experiments, three texture heightmaps (T4, T5, T6) were used to fabricate a new set of textured objects, whose curved geometry includes a local maximum and saddle (Figure 23). As in the previous experiment, the reference stimulus was the source texture, and the

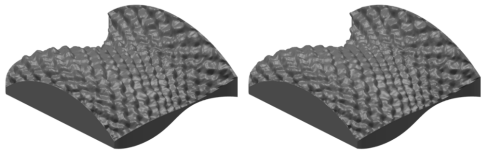


Fig. 23. A shape with curvature, including a local maximum and a saddle, was used for a less-restricted visual experiment. A rendering of the textured shape with the T5 input texture and a two-JND rougher optimized output texture is shown here.

test stimuli included a source texture along with three optimized textures and three linearly-scaled textures of different roughness values. The textures on the test stimuli were shifted by 50% so as to not appear identical to the reference source texture. Protocols were similar to the previous experiment: the reference and test stimulus were displayed sequentially for one second each, and the participant was asked whether the two plates had the same or different textures. The participant sat around 30 cm from the textures on the table, and was free to move their head and rotate the textures; in combination with the shape's curvature, the viewer was able to see the texture geometry from many directions.

Eight people participated in the study, and each performed two evaluations of each texture against the source. As shown in the results in Figure 24, participants judged the optimized textures the same as the reference a majority of the time, but the linearly-scaled textures were almost never judged the same. The linearly-scaled textures were judged the same as the reference at a lower rate than the previous experiment as a result of the new viewing angles, suggesting differences are more apparent when many view directions are allowed; in contrast, the optimized textures were judged the same at a rate similar to the previous experiments, showing that our optimization is robust to different viewing directions.

The bottom panel of Figure 22 shows textures accumulated by just-noticeable-difference (JND) threshold distance from the reference texture, according to the difference threshold of 19% found in [Tymms et al. 2017]. In all cases, the optimized textures match the references better than the linearly scaled textures. In general, linear scaling tends to perform more successfully for small decreases in roughness, but performs poorly for larger decreases or increases in roughness. Optimization creates textures that appear very similar for small differences in roughness; the visual difference is only visible when the target roughness is much larger.

5.4 Tactile roughness experiments

Tactile roughness experiments were used to validate the tactile roughness optimization. Stimuli for this experiment were the same six sets of five 3D-printed texture plates used in the first visual experiments.

In the tactile experiments, ten participants were asked to sort groups of five texture plates by touch from smoothest to roughest. In each trial, the five plates were placed in a random order beneath a translucent panel that obscured the textures' fine-scale appearance. Participants used their dominant index finger to press each plate and

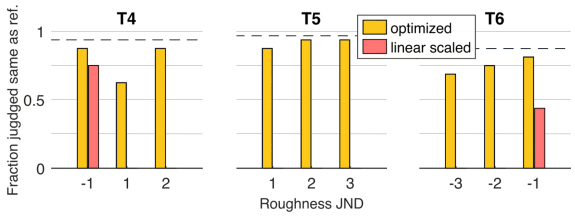


Fig. 24. Texture similarity results from the experiments on a subset of three textured shapes with curved geometry. The x-axis shows the difference in roughness just-noticeable-difference intervals between the test texture and the reference, and the y-axis shows the proportion judged the same. The dotted line shows the reference threshold a which the reference textures were judged the same as each other.

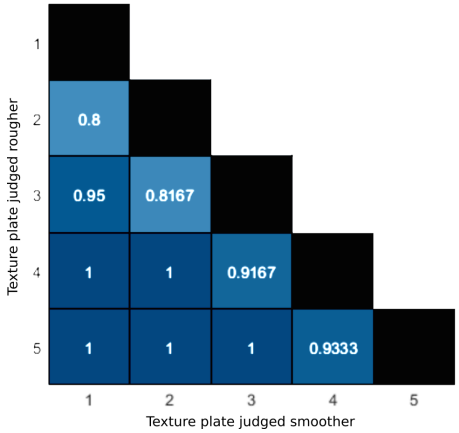


Fig. 25. This map shows the average proportion of trials in which each texture (vertical) was sorted as tactually rougher than each other texture (horizontal). Almost all discrepancies were between textures designed to differ by one threshold, and the error rate is close to the expected 84%.

determine a sorted order. They were free to feel the plates multiple times and to use as much time as needed.

5.4.1 Results. The heat map in Figure 25 shows the mean proportion with which each texture plate was judged rougher than each other of the same texture source. Textures are numbered from 1 to 5 according to the designed JND level from smoothest to roughest. Participants were able to reliably sort the plates, including pairs differing by only one threshold, a majority of the time. Participants sorted these most-similar pairs according to the designed ordering 85% of the time (across-subject standard deviation 4%), which is nearly the expected threshold of 84% with which consecutive plates were designed. Only three of the 60 total comparisons resulted in an ordering discrepancy between a pair of textures differing by more than one threshold step.

5.5 Contact area experiments

Twelve textures were fabricated to experimentally verify the change in contact area. For each texture, three versions were fabricated: the original texture, a texture optimized to have 70% the contact area, and a texture optimized to 140% the contact area. These 36 textures were fabricated as 10mm squares, using B9Creator V1.2 with a resolution of 50 µm in B9 black resin.

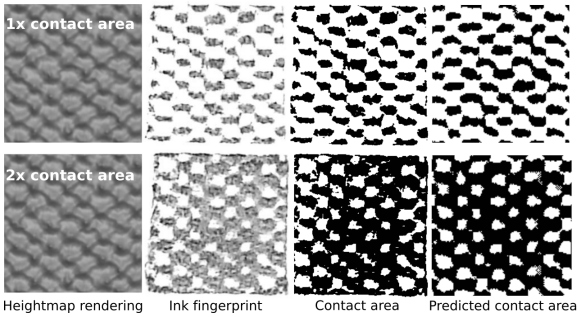


Fig. 26. The results of one texture optimized for a lower (top) or higher (bottom) contact area. From left to right: texture rendering, texture fingerprint, thresholded finger contact, and simulated contact.

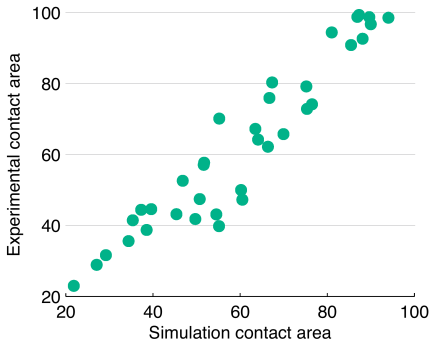


Fig. 27. Comparison of the experimental and simulated contact area of 36 textures.

To compute the contact area, the fabricated texture surfaces were coated in ink using a compliant sponge and an ink-pad. The thumb or second finger of each participant was covered with Tegaderm (3M), a thin layer of transparent plastic with a thickness of 0.1 mm. The tegaderm was used to avoid discrepancies due to the fingerprint ridges, and to provide easier cleaning of the finger surface between trials to avoid ink residue. The finger was pressed against the texture with a weight of 8.8 N placed on the finger to ensure uniform force. Then the finger was pressed to a sheet of paper to derive an inkprint of the contact surface.

Nine participants provided texture finger prints in this manner. The prints were scanned at 600dpi in 8-bit grayscale, and the contact areas were computed and averaged over all subjects. The pipeline is shown in Figure 26. All optimized contact surfaces fell within 20% of their target contact area, with an average difference of 9.2%. Additionally, the contact areas of the 36 printed textures were compared against the simulated contact areas. Figure 27, shows the result of this comparison, with a close linear correlation with an approximate slope of 1.

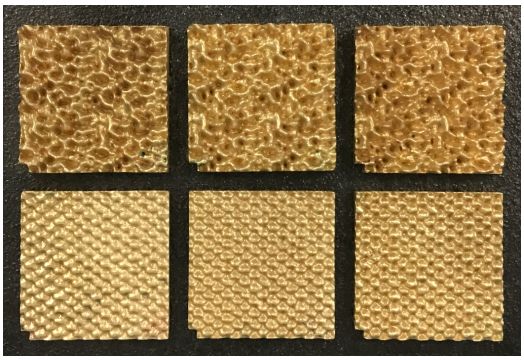


Fig. 28. Photograph of sets of bronze-cast textures used for tactile temperature experiments (T3 and T5). Textures are ordered from less to more contact area. Inconsistencies in appearance may be due to the manufacturing process and polishing.

5.5.1 Temperature Experiments. An additional set of experiments was used to determine whether the printed textures felt different from one another in tactile temperature.

Stimuli. The stimuli for this experiment were twelve textures: four base textures (T2, T3, T4, T5) each optimized with three different contact areas differing by 40%. Each texture surface was applied to the top of a flat plate 1.2mm in height. To enable tactile discriminability at room temperature for the purposes of the experiment, we used metal rather than plastic, due to its higher thermal conductivity: texture models were 3D printed and cast in bronze. A photograph is shown in Figure 28.

Setup and protocols. In the experiments, textures placed on a flat cast-iron plate over ice, which maintained a temperature at the top surface of approximately 16°C as measured by a laser thermometer.

In each trial, the participant was presented with two textures of the same class with different optimized contact area (either 40% smaller or 40% greater, where 40% is approximately the JND threshold for thermal discrimination described by [Tiest and Kappers 2009]). A cover was placed over the experiment area to hide the textures from view.

In experiments, eight participants were asked to feel the two textures using static pressure with the index finger and to answer which texture felt colder. Participants were allowed as much time as needed to feel the textures, and were given time between trials to ensure the finger itself was not too cold.

Results. Throughout the trials, participants responded that the texture with more contact area felt colder 83.1% of the time, which suggests that the threshold of discrimination is indeed approximately 40%, as found by previous research. For pairs that differed by two JND, the texture with more contact area was judged as colder 91% of the time.

6 APPLICATIONS

Applying tactile textures to fabricated objects is useful for both aesthetic and practical purposes. We have formulated several examples and have fabricated a subset as textured 3D models (Figure 29).

Modeling. Often, one might prefer a particular visual texture for an object while preferring a distinct tactile feeling. For example, imitation plastics are often used to match a specific material's appearance, and our model could help match the material's desired feeling. Our model could enable the creation of multiple surfaces that look similar but feel different, either for aesthetic purposes or to serve as a tactile signifier of another characteristic. It could also be used to make surfaces that feel similar but look different, which could be combined in a visual pattern or logo, for example on a mat, that feels uniform when touched.

We manufactured two different animal models as examples. First, a starfish model was textured with a relatively smooth surface texture (roughness 0.05). The texture was altered to feel rougher (roughness 0.092), and was applied to produce another, rougher fabricated starfish with the same appearance (shown in Figure 1). We also fabricated a textured model of a tree frog with a keeled wood pattern. The initial texture had a roughness of 0.045, and was modified to produce a smoother texture of roughness 0.03 and used to fabricate a smoother frog with the same appearance (Figure 29a).

Wearables. Tactile and visual aesthetics are common to clothing, jewelry, and other wearables, which often touch the skin. Tactile properties may also serve as functional. Some wearable devices, such as headphones with buttons, have areas that the user finds and uses by touch rather than sight; these areas could be hidden visually for aesthetic appearance or for more discreet use. Wearables could also use roughness actively to convey haptic signals that are unobtrusive to the user and invisible to others: a watchband with a high-resolution pin array could produce different tactile textures that could be felt by the user to convey different signals; similarly, altering the contact area could allow different rates of thermal transfer between a wearable and the user's skin.

As an example of a wearable with tactile aesthetics, we fabricated a bracelet band with links having the same texture appearance, but alternating smoother and rougher tactile feelings (Figure 29b). Smoother links had a roughness of 0.06, and the rougher links had roughness 0.09.

Accessibility. Tactile items and textures are particularly useful for people with visual impairments. If a designer creates two objects that look different, our model could be used to tune the textures so that they also feel different, while preserving the designed visual appearance. Visual textured objects are commonly used in pieces for board games, puzzles, and household items, where colors or visual labels are often used to distinguish between otherwise similar objects or regions. A variation in tactile feeling can provide similar cues for a person unable to see the differences. As an example, we produced a model for a dimmer light switch slider. The texture gradient looks the same throughout, but the roughness increases such that it will correspond with the light intensity as the slider is moved (Figure 29c).

7 CONCLUSION

We have presented an optimization procedure to preserve texture appearance while altering tactile roughness or temperature. We used neural networks to enable computation of tactile roughness, contact

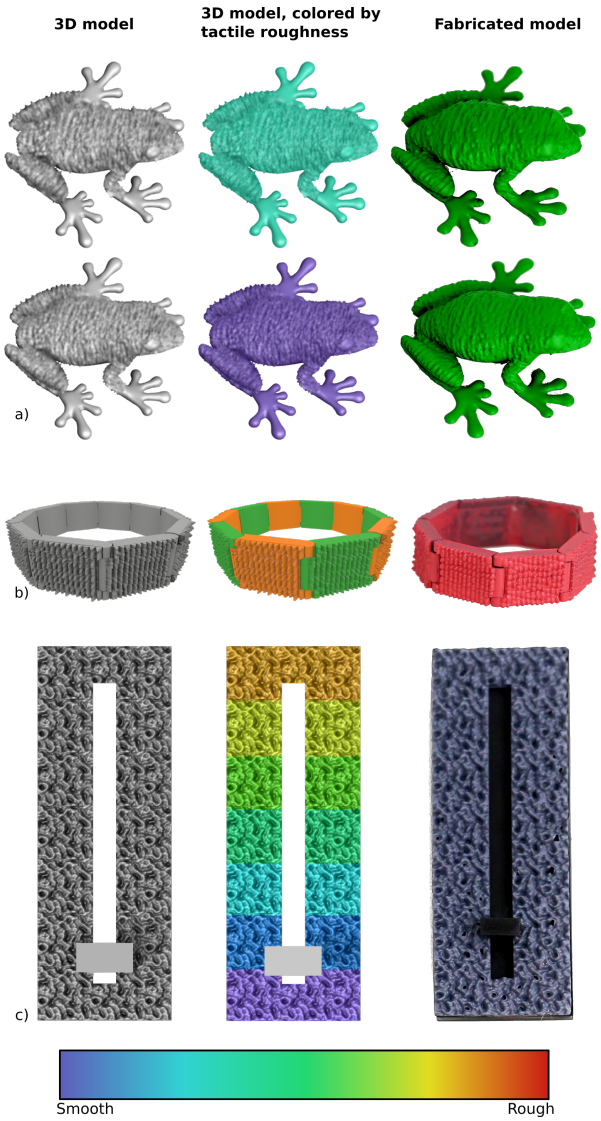


Fig. 29. From left to right: textured models; models colored by roughness; photographs of 3D-printed models. a) Two frogs textured with a tactile wood texture optimized for different roughnesses. (Modified from [YahooJAPAN 2013]). b) A bracelet with textured links that have the same visual texture but have alternating roughnesses. c) A procedural texture slider for a light switch, where the tactile roughness corresponds to light intensity.

area, and visual appearance at speeds several orders of magnitude faster than the standard methods, providing differentiable functions usable in optimization for a target appearance and feeling. We used psychophysical experiments to demonstrate that our method provides a significant improvement over simple linear scaling in controlling tactile roughness, and we provided several examples of how our procedure can be used to produce interesting and useful textured objects.

7.1 Limitations and Future Work

While the tactile model has been tested on objects with moderate curvature ([Tymms et al. 2018]), it may not be usable for high-curvature 3D objects. Furthermore it was tuned for hard materials, and it has a minimum feature resolution; using it for soft or fine materials may require changes. Our static touch simulation is adequate for dynamic touch up to a certain resolution, as static touch receptors dominate perception for features over $100\mu m$ ([Hollins and Risner 2000]); nevertheless it may be improved by dynamic simulation. The model is also based on a simulation of a simplified model of human skin, which, while found to be robust, may be improved by a more complex model. Our procedure could used for a different material or more physically complex skin structure by retraining the neural network on a new set of simulation field outputs.

Similarly, our procedure describing visual appearance was tuned for the shading of the our material (diffuse plastic resin) and may not be directly usable for surfaces that are much more glossy, translucent, or non-smooth. In these cases, our method could be modified to learn the rendered appearance for a particular desired material given a suitable training set. However, as seen in Figure 28, our current visual model can still work to preserve visual appearance fairly well even for non-matte materials.

Our model presents a tradeoff between preserving exact visual appearance and achieving an exact tactile roughness. Very high changes in tactile roughness may not be achievable while fully preserving visual appearance. We found that similarly-appearing textures can typically be produced within a range of 3-4 JND thresholds in each direction. Our metric for visual appearance similarity is likely a lower bound for perceptual similarity, so a fast perceptually-based method for texture similarity could be used instead in the optimization and could improve texture generation. Our model was evaluated to target either a tactile roughness or contact area; optimizing for both or more quantities is future work. Other optimization parameters could also be used: for example, we could enforce printability constraints depending on the printer used to manufacture a model.

Our visual model uses shading from an overhead view with ambient lighting. At severe angles or severe lighting conditions, the differences may be more apparent. Our model could be tuned to a particular lighting condition or viewpoint if it were used in the training set, but any optimized texture likely will not appear exactly the same under all lighting and viewing conditions, as some geometric changes will always be present near the surface. However, as we found in both optimization tests and human user studies with curved objects, our optimization using a single viewpoint is robust, and the results look similar to the target even when viewed at other angles.

Our model limits texture height to 3mm as a manufacturing constraint. We have observed that due to the limited elasticity of the finger, textures deeper than this are not different tactually from those with the lower-depth truncated to 3mm. However, we could easily optimize a taller texture by optimizing the top 3mm of it, and preserving the remainder.

To aid in the fabrication process, our method could be integrated into a 3D modeling tool to provide precise control of tactile feeling

when modeling a textured fabricable object. Using our model and existing models for compliance using compliant microstructures, it would be possible to control three of major dimensions of touch: compliance, temperature, and roughness, and to study the unknown interactions between these properties. Creating objects with differing tactile properties as separate from appearance may also be of interest to the fields of neuroscience and neurophysiology in future studies of psychophysics and multi-modal perception.

REFERENCES

Connelly Barnes and Fang-Lue Zhang. 2016. A survey of the state-of-the-art in patch-based synthesis. *Computational Visual Media* (2016), 1–18. https://doi.org/10.1007/s41095-016-0064-2

Phil Brodatz. 1966. *Textures: A Photographic Album for Artists and Designers.* Dover.

Desai Chen, David IW Levin, Piotr Didyk, Pitchaya Sitthi-Amorn, and Wojciech Matusik. 2013. Spec2Fab: a reducer-tuner model for translating specifications to 3D prints. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 135.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. 2014. Describing Textures in the Wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR).*

Charles E Connor, Steven S Hsiao, John R Phillips, and Kenneth O Johnson. 1990. Tactile roughness: neural codes that account for psychophysical magnitude estimates. *The Journal of neuroscience* 10, 12 (1990), 3823–3836.

Donald Degraen, André Zenner, and Antonio Krüger. 2019. Enhancing Texture Perception in Virtual Reality Using 3D-Printed Hair Structures. (2019).

Alexei A Efros and Thomas K Leung. 1999. Texture synthesis by non-parametric sampling. In *iccv*. IEEE, 1033.

Oskar Elek, Denis Sumin, Ran Zhang, Tim Weyrich, Karol Myszkowski, Bernd Bickel, Alexander Wilkie, and Jaroslav Křivánek. 2017. Scattering-aware texture reproduction for 3D printing. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 241.

Galal Elkharraz, Stefan Thumfart, Diyar Akay, Christian Eitzinger, and Benjamin Henson. 2014. Making tactile textures with predefined affective properties. *Affective Computing, IEEE Transactions on* 5, 1 (2014), 57–70.

Alexander IJ Forrester and Andy J Keane. 2009. Recent advances in surrogate-based optimization. *Progress in aerospace sciences* 45, 1-3 (2009), 50–79.

Leon Gatys, Alexander S Ecker, and Matthias Bethge. 2015. Texture synthesis using convolutional neural networks. In *Advances in neural information processing systems*. 262–270.

Bjoern Haefner, Yvain Quéau, Thomas Möllenhoff, and Daniel Cremers. 2018. Fight ill-posedness with ill-posedness: Single-shot variational depth super-resolution from shading. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 164–174.

Jan S Hesthaven and Stefano Ubbiali. 2018. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *J. Comput. Phys.* 363 (2018), 55–78.

Mark Hollins and Sliman J Bensmaïa. 2007. The coding of roughness. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie experimentale* 61, 3 (2007), 184.

Mark Hollins and S Ryan Risner. 2000. Evidence for the duplex theory of tactile texture perception. *Perception & psychophysics* 62, 4 (2000), 695–705.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. *arXiv preprint* (2017).

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018. Differentiable Monte Carlo Ray Tracing through Edge Sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 37, 6 (2018), 222:1–222:11.

Louise R Manfredi, Hannes P Saal, Kyler J Brown, Mark C Zielinski, John F Dammann, Vicky S Polashock, and Sliman J Bensmaia. 2014. Natural scenes in tactile texture. *Journal of neurophysiology* 111, 9 (2014), 1792–1802.

Rafat Mantiuk, Kil Joong Kim, Allan G Rempel, and Wolfgang Heidrich. 2011. HDR-VDP-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions. In *ACM Transactions on graphics (TOG)*, Vol. 30. ACM, 40.

Rodrigo Martín, Min Xue, Reinhard Klein, Matthias B Hullin, and Michael Weinmann. 2019. Using patch-based image synthesis to measure perceptual texture similarity. *Computers & Graphics* 81 (2019), 104–116.

MITMediaLab. 1995. Vision texture, VisTexdatabase. http://vismod.media.mit.edu/vismod/imagery/VisionTexture/

Michal Piovarči, David I.W. Levin, Jason Rebello, Desai Chen, Roman Ďurikovič, Hanspeter Pfister, Wojciech Matusik, and Piotr Didyk. 2016. An Interaction-Aware, Perceptual Model For Non-Linear Elastic Objects. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 35, 4 (2016).

Javier Portilla and Eero P Simoncelli. 2000. A parametric texture model based on joint statistics of complex wavelet coefficients. *International journal of computer vision* 40, 1 (2000), 49–70.

M Raissi, P Perdikaris, and GE Karniadakis. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378 (2019), 686–707.

Olivier Rouiller, Bernd Bickel, Jan Kautz, Wojciech Matusik, and Marc Alexa. 2013. 3D-printing spatially varying BRDFs. *IEEE computer graphics and applications* 33, 6 (2013), 48–57.

Christian Schüller, Daniele Panozzo, and Olga Sorkine-Hornung. 2014. Appearance-mimicking surfaces. *ACM Transactions on Graphics (TOG)* 33, 6 (2014), 216.

Liang Shi, Vahid Babaei, Changil Kim, Michael Foshey, Yuanming Hu, Pitchaya Sitthi-Amorn, Szymon Rusinkiewicz, and Wojciech Matusik. 2018. Deep multispectral painting reproduction via multi-layer, custom-ink printing. In *SIGGRAPH Asia 2018 Technical Papers*. ACM, 271.

Wouter M Bergmann Tiest. 2010. Tactual perception of material properties. *Vision research* 50, 24 (2010), 2775–2782.

Wouter M Bergmann Tiest and Astrid ML Kappers. 2009. Tactile perception of thermal diffusivity. *Attention, perception, & psychophysics* 71, 3 (2009), 481–489.

Cesar Torres, Tim Campbell, Neil Kumar, and Eric Paulos. 2015. HapticPrint: Designing Feel Aesthetics for Digital Fabrication. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 583–591.

Chelsea Tymms, Esther P Gardner, and Denis Zorin. 2018. A Quantitative Perceptual Model for Tactile Roughness. *ACM Transactions on Graphics (TOG)* 37, 5 (2018), 168.

Chelsea Tymms, Denis Zorin, and Esther P Gardner. 2017. Tactile perception of the roughness of 3D-printed textures. *Journal of Neurophysiology* 119, 3 (2017), 862–876.

Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. 2016. Texture Networks: Feed-forward Synthesis of Textures and Stylized Images.. In *ICML*. 1349–1357.

Thomas SA Wallis, Christina M Funke, Alexander S Ecker, Leon A Gatys, Felix A Wichmann, and Matthias Bethge. 2017. A parametric texture model based on deep convolutional features closely matches texture appearance for humans. *Journal of vision* 17, 12 (2017), 5–5.

Kai Wang, Guillaume Lavoué, Florence Denis, and Atilla Baskurt. 2008. A comprehensive survey on three-dimensional mesh watermarking. *IEEE Transactions on Multimedia* 10, 8 (2008), 1513–1527.

Xin Wang, Man Jiang, Zuowan Zhou, Jihua Gou, and David Hui. 2017. 3D printing of polymer matrix composites: A review and prospective. *Composites Part B: Engineering* 110 (2017), 442–458.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Vol. 2. Ieee, 1398–1402.

Li-Yi Wei and Marc Levoy. 2000. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 479–488.

YahooJAPAN. 2013. Frog. https://www.thingiverse.com/thing:182144 Modified.

Ying Yang, Ruggero Pintus, Holly Rushmeier, and Ioannis Ivrissimtzis. 2017. A 3D steganalytic algorithm and steganalysis-resistant watermarking. *IEEE transactions on visualization and computer graphics* 23, 2 (2017), 1002–1013.

Ning Yu, Connelly Barnes, Eli Shechtman, Sohrab Amirghodsi, and Michal Lukac. 2019. Texture Mixer: A Network for Controllable Synthesis and Interpolation of Texture. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Lin Zhang, Lei Zhang, Xuanqin Mou, David Zhang, et al. 2011. FSIM: a feature similarity index for image quality assessment. *IEEE transactions on Image Processing* 20, 8 (2011), 2378–2386.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Xiaoting Zhang, Guoxin Fang, Chengkai Dai, Jouke Verlinden, Jun Wu, Emily Whiting, and Charlie CL Wang. 2017. Thermal-comfort design of personalized casts. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, 243–254.

Yang Zhou, Zhen Zhu, Xiang Bai, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. 2018. Non-stationary Texture Synthesis by Adversarial Expansion. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 37, 4 (2018), 49:1–49:13.