Data-driven Fingertip Appearance for Interactive Hand Simulation

Sheldon Andrews[*]
McGill University

Marc Jarvis[†]
McGill University

Paul G. Kry[‡]
McGill University

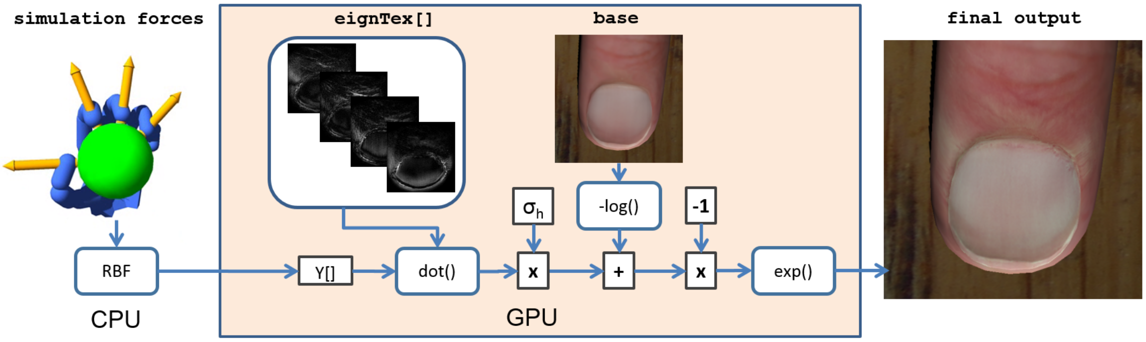

Figure 1: *An illustration of the texture modification process used by our system.*

Abstract

Contact on a finger pad results in deformation that redistributes blood within the fingertip tissue in a manner correlated to the pressure. We build a data-driven model that relates contact information to the visible changes of the finger nail and surrounding tissue on the back of the finger tip. Our data analysis and model construction makes use of the space of hemoglobin concentrations, as opposed to an RGB color space, which permits the model to be transferred across different fingers and different people. We use principal component analysis to build a compact model which maps well to graphics hardware with an efficient fragment program implementation. We provide a validation of our model, and a demonstration of a grasping controller running in a physically based simulation, where grip strength is visible in both hand posture and the appearance of color changes at the fingertips.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: hands, grasping, texture, hemoglobin, light scattering, physics-based animation, interactive simulation

1 Introduction

Interactive character animation is a key part of video games and can be important in many other applications, such as ergonomic design, training simulations, and movie production. Digital characters have become very realistic over the years. Many skinning techniques have been developed to better model the deformation of skin; physically based simulation of garments have improved realism by providing valuable secondary motion to a character's movement; and control strategies have allowed for natural simulated motion that reacts to environmental changes in a plausible manner.

While motion is central to the problem of character animation, and deforming geometry is a primary factor explaining overall appearance, color variations across the character's skin are essential for rendering life-like images. Texture and bump mapping are now standard, while spatially varying surface scattering distribution functions and light transport approximations are becoming more common for highly realistic image production. In some cases, reflectance or scattering models can be parameterized. For instance, varying amounts of perspiration can drive glossiness to permit simple appearance changes, while it is also possible to model subsurface blood variations due to pose, pressure, alcohol consumption, or exercise. This is where our work fits in, as we are modeling color changes due to the contact forces available in an interactive simulation. Our approach to modeling these color changes is with a simple scattering model.

Control, deformation, and appearance can make use of procedural techniques, physics simulation, or data-driven models. Physically based deformation and rendering, in combination with anatomically detailed models, have become a popular approach for improving the quality of images and animation. We follow the alternative, which is to use captured data to create empirical models that replicate the behavior of real world example measurements. In contrast, our grasping motion controller uses a multi-body physics simulation and standard control techniques, similar to others in graphics and robotics.

Figure 1 shows an overview of our appearance modification process. Our contributions can be summarized as follows: we present a simple capture process for collecting fingertip photos and fingerpad pressure images, an analysis method that uses a custom simplified hemoglobin concentration model, and an efficient appearance model implementation within a fragment program, which integrates easily with a real-time interactive physics-based simulation of grasping control.

[*] e-mail:sheldon.andrews@mail.mcgill.ca
[†] e-mail:marc.jarvis@mail.mcgill.ca
[‡] e-mail:kry@cs.mcgill.ca

2 Related Work

Over the years, many improvements have been proposed to improve the appearance of animated characters beyond the standard linear blend skinning model [Magnenat-Thalmann et al. 1988]. For instance, pose space deformation interpolates geometry corrections using radial basis functions [Lewis et al. 2000], shape by example interpolates corrections mapped back into the rest pose with the inverse skinning transform [Sloan et al. 2001]. In an anatomically accurate approach, geometric features on the back of hand can be driven by tendon simulations [Sueda et al. 2008]. Alternatively, Huang et al. [2011] capture and model pose dependent wrinkles on hands, while another popular approach is to build wrinkle texture maps into the character rig to provide these details without the cost of geometric modeling [Oat 2007; Dutreve et al. 2011]. Similar to pose space deformation [Lewis et al. 2000] and shape by example [Sloan et al. 2001], the EigenSkin technique interpolates shapes within a memory efficient basis computed with principal component analysis [Kry et al. 2002]. Our approach is similar to this, except that instead of interpolating geometric shape, we interpolate a blood concentration texture in a memory efficient representation and use this to compute a correction to the skin color. We use contact information within a physical simulation to drive the color changes. This is similar to playable universal capture [Borshukov et al. 2007], except that the texture appearance is driven by simulated attributes (contacts in the physics simulation) rather than a previously recorded appearance trajectory.

There are important changes in the appearance of the fingertips during posture changes and contact. In fact, it is possible to estimate posture and touch force using sensors that measure color changes under the fingernails [Mascaro and Asada 2004]. Inspired by this difficult inverse problem, our focus is the creation of a data-driven model suitable for synthesizing these appearance changes. While it may be convenient to model appearance variations in RGB color space, it is preferable to work in the space of changing hemoglobin concentrations. Displacement of blood due to contact is the primary explanation for the appearance change, and building a model that works directly in the space of hemoglobin concentrations allows for easier reuse of captured data across different fingers and different people. Thus, we estimate hemoglobin variations across the example images that we collect. A variety of techniques have been devised for estimating these quantities for medical purposes. For instance, camera based sensors can be constructed for measuring melanin, hemoglobin, and oxygenation of tissue, through the use of a known spectral illumination [Jakovels et al. 2011]. More recently it has been shown that Wiener estimation methods can produce estimates from an appropriately white balanced RGB camera [Nishidate et al. 2013]. In computer graphics, Tsumura et al. [2003] use independent component analysis to estimate melanin and hemoglobin components using a single image, such as the image of a face.

Our hemoglobin estimation method is inspired by others [Jimenez et al. 2010; Tsumura et al. 2003; Tsumura et al. 1999]. It closely relates to that of Tsumura et al. [1999], in that we use a negative log color space analysis, with the assumption that a simple Lambert-Beer scattering law explains skin color. Otherwise, the difference with our work is that we do not need to perform independent component analysis to extract melanin concentrations because we can focus entirely on the appearance variation due to hemoglobin concentration variations across several images. At run time, we adjust the color of skin in the log-color space using the absorption properties of hemoglobin (estimated from our example images). In contrast, Jimenez et al. [2010] produce skin color variations with a lookup table indexed by hemoglobin and melanin concentrations. Their objective is to model the appearance of dynamic faces using a skin appearance rig, and they use hemoglobin maps to control skin color, permitting variation of appearance under the deformation of different blend shapes, or due to other conditions, such as exercise or alcohol consumption. Kider et al. [2011] have a data driven appearance model driven by fatigue which includes models of flushing and perspiration appearance. In contrast, Boukhalfi [2012] focuses on face geometry during strenuous exercises. Also closely related to our work is that of Donner et al. [2008]. They specifically focus on hands, and modify parameter maps to show the possibility of appearance changes due to blood being squeezed out of areas deformed by contact. In our work, in contrast, we use an interactive physics based simulation to drive the realistic color changes. Likewise, our data-driven model, which interpolates blood distribution examples for different pressure magnitudes, employs data reduction to permit an efficient implementation.

There are other examples where force capture data has been used in computer animation. The *Footsee* system estimates full body pose from pressure images of the feet on the ground [Yin and Pai 2003], while Aladdin and Kry [2012] estimate static climbing poses from force data measured with an instrumented climbing wall. Finally, Kry and Pai [2006] describe a method of capturing and resynthesizing grasps where force measurements are used in conjunction with motion capture to estimate the stiffness of joints. Force measurements have also been used to measure both linear and non-linear deformation of soft tissue, as well as frictional properties [Pai et al. 2001; Bickel et al. 2009].

3 Data Capture

Our capture process involves collecting photographs and force measurements of fingertips that are applying different amounts of pressure. We use a TekScan Grip system for measuring contact pressure at a fingertip. This system has a resolution of 6.2 *taxels* per square centimeter, where a taxel is a "tactile pixel" or "tactile element" within a pressure image. At the fingertip, the pressure image resolution is 4×4, giving a total of 16 taxels. We multiply the pressure value at each taxel by the area of an individual taxel, and sum up the force contributions of all taxels under the fingertip to produce a total force estimate. We also compute a center of pressure, but we currently only investigate the appearance change associated with forces centered on the finger pad. We have collected data with varying center of pressure to be used in future work.

We capture images with a consumer camera. Because we use a makeshift diffuse light box (consisting of David laser scanner calibration panels), we assume that the images effectively capture the albedo variation of the skin and fingernail. The camera white balance is set so that we can compute reasonable hemoglobin concentration estimates (see more details in Section 4).

Figure 2 shows our capture setup along with a sample of the captured pressure data. We have captured data for index and ring fingers of the right hand across three subjects. The capture process starts with the capture subject placing the distal finger pad of the selected finger at the center of the fingertip taxel grid. The capture subject interactively adjusts the positioning of their finger by applying a small amount of perpendicular force to the sensor surface. The fingertip position is adjusted until the center of pressure is aligned with the center of the sensor grid. Once capture begins, the finger remains in contact with the sensor, and if finger slippage occurs, then we restart the process. The back of the finger remains visible to the camera as the subject explores different pressure magnitudes (as well as varying the center of pressure in our extended future work data set). Pressure and image data are captured simultaneously. The interactive nature of the capture software allows the subject to avoid collecting multiple similar examples, which could

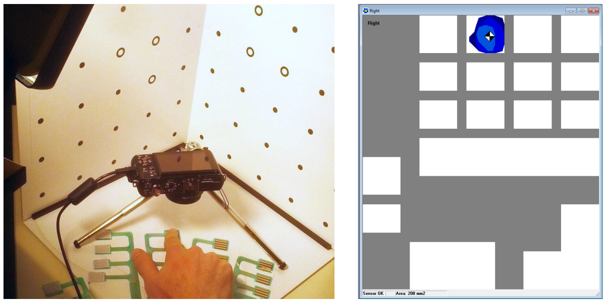

Figure 2: *Left, makeshift diffuse light box and TekScan capture setup. Right, example pressure data showing center of pressure.*

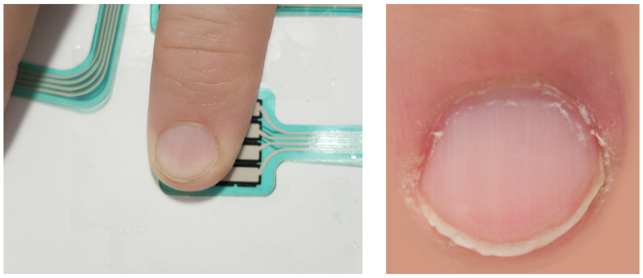

Figure 3: *Left, an example raw captured image, and right, the cropped and masked pre-processed image ready for analysis.*

be problematic for over-fitting during model construction.

Because the fingertip may translate and rotate slightly between different photographs when applying different amounts of pressure, the example images need to be warped into a consistently parameterized texture image (note that rotation is more pronounced in examples that involve varying centers of pressure). In particular it is important to match the shape of the fingernail. While it would be possible to develop a custom automated process to warp the images, we choose to do this manually for the examples that we collect (i.e., 10 samples used for our data-driven model). This pre-processing task is not onerous, and ultimately we observe that the estimated model for one fingertip is sufficiently general to apply to all fingertips and across individuals (validation of this can be found in Section 6). We use the Puppet Warp tool in Photoshop to deform the images, and apply a mask to avoid including any of the fingertip silhouettes and background in the analysis. Figure 3 shows an example photograph, along with a pre-processed example image which is ready for analysis.

4 Model Fitting

While Mascaro and Asada [2004] note that finger posture also plays an important part in fingertip appearance, we focus solely on contact forces. Thus our appearance model can be seen as a function mapping a force on a finger pad to color variations in the texture. Given that these color changes are largely driven by the distribution of blood in the fingertip, we choose to do our regression in the space of hemoglobin concentrations.

In this section, we first explain our simple model for appearance due to varying concentrations of hemoglobin and the effect that it has on light scattering. We then describe how we estimate the absorption properties of hemoglobin in our images, and from there, work in the space of hemoglobin concentration changes across the fingertip images. Finally, we describe an interpolation function that allows us to reproduce the collected examples.

4.1 Material Concentration and Light Absorption

We assume a simple model for light scattering within human skin following previous work [Tsumura et al. 1999; Jensen et al. 2001; Tsumura et al. 2003]. The transmittance T of light through a material at a given wavelength λ, is defined as the fraction of the exiting spectral radiance $L(\lambda)$ over the initial incident radiance $L_0(\lambda)$. The Lambert-Beer law models transmittance as a function of the distance d that the light travels within the material, and the molar absorptivity and concentration of a material $\alpha(\lambda)$ at the given wavelength, specifically,

$$T = \frac{L(\lambda)}{L_0(\lambda)} = 10^{-\alpha(\lambda)d} \in [0, 1]. \tag{1}$$

Since we are using RGB images, we will take the convenience of defining $\alpha \in \mathbb{R}^3$ as a three-tuple of values corresponding to red, green, and blue wavelengths. When there exists a mixture of materials, we can write α as a sum of contributions from each of the materials. For instance, in our case it is useful to write the absorption as the sum of two parts, the first due to hemoglobin, α_h, and the second due to other constituents, α_o, such as melanin and keratin.

In the context of a diffuse reflection, we assume that our RGB images do not contain any lighting variation due to geometry, and in one sense, we can view the pixel values as varying surface albedo, which can be used as the diffuse material parameter in a lighting program. Here, we assume this per pixel *reflected* light percentage can be treated as transmission along a sub-surface light path of unknown fixed length within a homogeneous material. We further assume the light path d to be constant across all pixels. Thus, the negative logarithm of the pixel components gives us a proportional absorption property of the material mixture recorded at that pixel. That is, given a pixel's color as $(R, G, B) \in [0, 1]^3$, we compute

$$\alpha = (-\log(R), -\log(G), -\log(B)) \in [0, \infty]^3. \tag{2}$$

Because the molar absorptivity of a material is constant, we factor α into a normalized *pigment* vector and a scalar *quantity*. Thus, we model color changes in log space as changes in material quantity, and at any given pixel we can write

$$\alpha = q\sigma_h + \alpha_o, \tag{3}$$

where $q \geq 0$ is the hemoglobin quantity, $\sigma_h \in \mathbb{R}^3$ is the hemoglobin pigment, and α_o models the absorption of all other materials. We expect q at a given pixel to change depending on the distribution of blood in the fingertip, while the concentration of other materials, and in turn the absorption, remains constant.

4.2 Hemoglobin Pigment Estimation

With the white balance of the camera set correctly, our images will have colors appropriate for use in a texture map. However, because the illumination spectrum and camera response curves are unknown, the observed pigmentation of hemoglobin may not be fixed and must be estimated from the capture data. This is relatively straightforward because we have numerous sample images in which the only changes are due to the redistribution of blood in the fingertip tissues as different pressures are applied to the finger pad. Thus, the negative log color values of different images at a given pixel should only vary in the direction σ_h.

We estimate σ_h as the first principal component of the negative log color of all pixels and all images. Each pixel's negative log color is centered independently to account for the varying amounts of

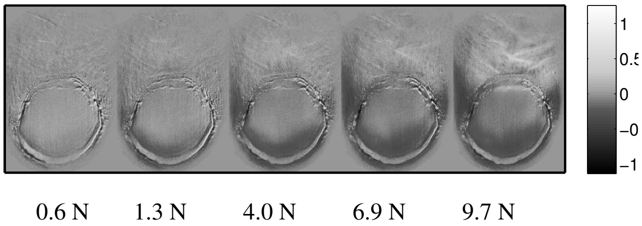

Figure 4: *Example quantity-change images showing considerable correlation across examples at different nonzero finger pad forces.*

other materials (i.e., α_o) across the fingertip. In our case, we use the negative log color of the pixel in the zero pressure image as the center, though the mean negative log value would also be suitable (or likewise, that of any example image).

Note that the vector σ_h will have all positive components because absorption at different wavelengths is positively correlated (i.e., a material can only absorb more of any given light wavelength when its concentration is increased). Averaging across three data sets for three individuals, we estimate $\sigma_h = (0.19, 0.77, 0.61)$, which corresponds to higher absorption of green and blue wavelengths in comparison to red. Section 6 discusses estimates and validation further.

Letting σ_h be normal length, we can compute the hemoglobin quantity for any pixel in any image with a simple dot product,

$$q = \alpha \cdot \sigma_h, \tag{4}$$

while the residual provides absorption due to other materials,

$$\alpha_o = \alpha - q\sigma_h. \tag{5}$$

We compute hemoglobin quantity-change images for all of the examples to permit analysis and modeling (see Figure 4). While the pigment estimation is not sensitive to the center, we choose the zero pressure image as the center for the example hemoglobin quantity-change images. Therefore, the quantity has zero change when there is zero force on the finger pad. We do not store the α or residual α_o vectors as they can easily be computed on the fly in the shader program as described in Section 5.

4.3 PCA Model of Hemoglobin Quantity Changes

Because there is considerable correlation among the hemoglobin quantity-change images that we observe with the different finger pad forces, it is appropriate to use principal component analysis (PCA) to compute an efficient basis to represent these changes in our example images.

There is camera noise in our example images, and because blood redistribution does not happen at the spatial frequencies captured by our full resolution example images, we down-sample the images to 256×256. Furthermore, we have considered applying a Gaussian blur to reduce camera noise, but this is not essential as we have not observed noise as having a significant effect on the first few principal components.

Figure 5 shows the principal component basis, along with the variation explained by the different basis vectors. We call these basis images *eigen textures*. Note that with just the first two we can capture over 94% of the variation observed in our examples.

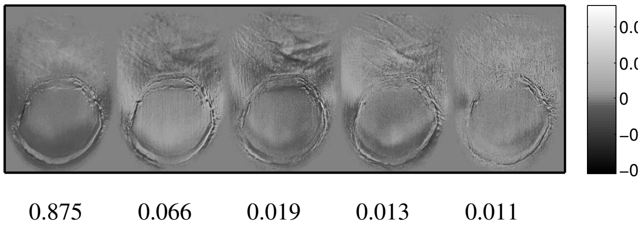

Figure 5: *Visualization of the principal component basis vectors, i.e., eigen textures, explaining quantity-change images. Below each is a value corresponding to the variation explained by each vector.*

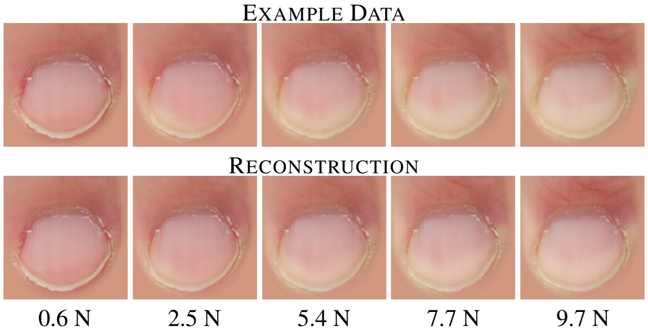

Figure 6: *Top row shows processed example images, while the second row shows reconstructions using 4 principal components. Note the whitening at the sides of the fingertip and under the nail at the tip as the pressure increases due to blood being squeezed out into surrounding tissues. Also notice the similarity between examples and Reconstruction.*

Figure 6 shows reconstructions of the example RGB images using a reduced basis with only 4 components. The negative log color image is modified by adding σ_h scaled by the per pixel quantity change computed with the truncated eigen texture basis. The result is negated, exponentiated, and finally clamped to $[0, 1]$ to produce the reconstruction RGB image. Note that clamping is only necessary for extreme modifications and is typically unnecessary.

We are effectively modeling a process of appearance change due to blood moving about in the fingertip. As such, it would be ideal to use a data interpolation technique that appropriately takes into account mass transport [Bonneel et al. 2011]. In the interest of having an inexpensive interpolation, we instead choose to model blood distribution variation in a principal component basis. The consequence is that we cannot do any better than blending examples, and if we do not have enough examples, then our results will exhibit less prominent differences from either the average or zero force appearance.

4.4 Interpolation

At this point, we have a collection of N examples consisting of a measured total finger pad force in newtons, $x \in \mathbb{R}$, and a hemoglobin concentration change that is represented in a truncated PCA basis, $y \in \mathbb{R}^M$. We use $M = 4$ because it is both sufficient and maps easily to graphics hardware.

We use a quadratic polyharmonic spline to interpolate the PCA co-

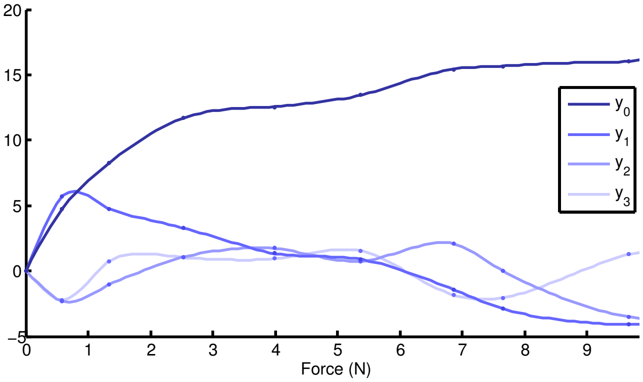

Figure 7: *Interpolation of the examples, shown with dots, in the coefficients of the PCA basis.*

ordinates at the sampled forces,

$$y(x) = \sum_{j=1}^{N} w_j \phi(\| x - x_j \|), \tag{6}$$

where $\phi(r) = r^2 ln(r)$ and x_j is force data for the jth sample. The weights w are computed easily with a linear solve. We use this radial basis function interpolation approach because it will extend easily to interpolate other important features, such as the center of pressure on the finger pad, and the posture of the distal joint. In our preliminary work in this direction, we note that care is necessary in defining the distance metric for x such that it is meaningful. Ranges of different components must be taken into account to allow for meaningful comparison (position, force, angle).

Figure 7 shows plots of how well and how smooth the interpolations fit the data. The coefficients of the higher index principal components are noticeably smaller. This is because the basis vectors are all unit length, and the higher index vectors only capture a small portion of the variation.

5 Reconstruction Implementation

In this section, we describe the reconstruction of the fingertip appearance in the context of rendering our full hand model. Thus, we will address how we pack information into textures and implement the reconstruction process in a fragment shader.

Our model is suitable for any skinned hand rig. We started with a hand model exported from Daz Studio, and in our current work we are using a hand model purchased from TurboSquid, which has higher quality textures. Both models include linear transform blending weights. We use FBX format for the convenience of being able to load and display the models easily within OpenSceneGraph.

5.1 Textures

In order to modify the colors of the diffuse texture map, we create a second texture map shown in Figure 8. We name this texture `stiaTex` to denote the contents of its four components: `s` and `t` texture coordinates, a finger index `i`, and an alpha mask `a`. The red and green channels contain the texture coordinates for the fragment's corresponding location in the pre-processed example image (to access the eigen textures for computing the fragment's hemoglobin quantity change). The blue channel contains an index i to identify the finger tip to which the fragment belongs (this allows us to look up the appropriate finger's PCA coordinates which were computed from the contact force). Finally, the alpha channel contains an alpha mask to allow a smooth blend between the modified colors and the base texture at the edges of the fingertip.

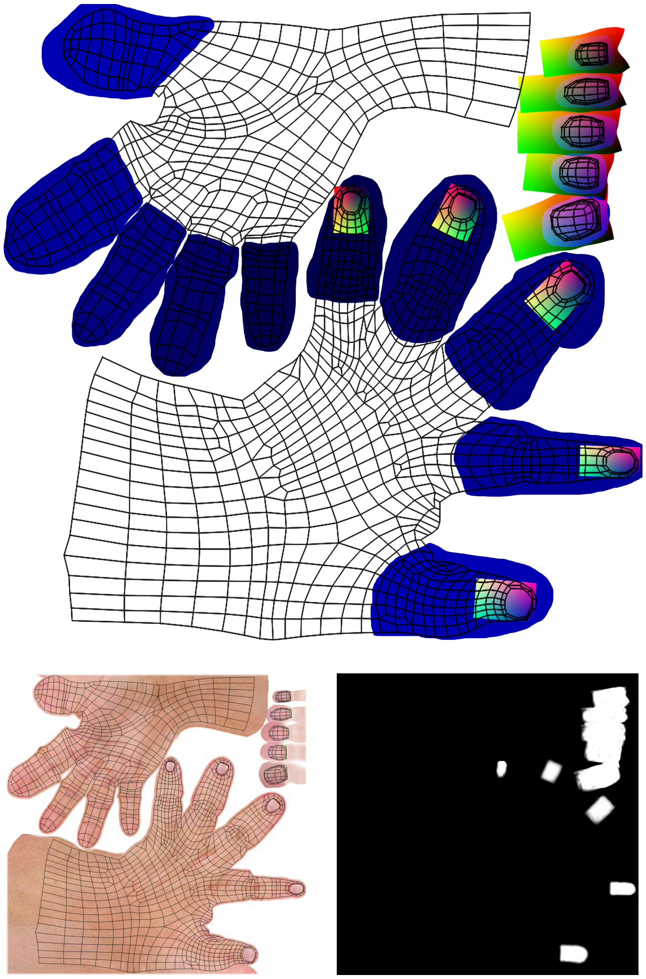

Figure 8: *Textures used in computing varying fragment colors at the fingertips, with the base texture shown at bottom left. Top shows the* `stiaTex`*, with texture coordinates in red and green for looking up hemoglobin concentration changes, and the finger index encoded with different shades in the blue channel. Bottom right shows the alpha mask with feathered edges used to blend the base texture with color values modified due to finger pad forces.*

Note that the alpha component is shown separately at the bottom right of Figure 8 to make it easier to see. Likewise, we overlay black lines to show the mesh coordinates within the texture to display how the texture contents relates to the mesh geometry. Obviously, our run time textures do not contain these black lines. Furthermore, note that these mesh lines reflect the coarse level resolution of the mesh, while we use a subdivided mesh for better quality at run time.

The index information is easy to paint into the `stiaTex` texture image. However, a bit more work is involved to set the red and green channels to the appropriate texture coordinates. The process we use is to first prepare a red-green texture gradient identity map, and overlay this with our pre-processed fingertip example image. We then paste the red-green texture coordinate image along with the zero pressure example image in a coupled layer into the `stiaTex` image. The pasted layers must first be scaled, translated, and rotated to roughly match with the back of a fingertip in the base texture. We then use the Photoshop Puppet Warp tool to carefully deform the fingertip example image to match features in the base texture and mesh geometry. Once we have a good match, the warped red-green gradient can be *flattened* into the red and green channels of the `stiaTex` image, and we repeat the process for all fingers.

The alpha mask is not difficult to produce. We partially automate the process by using the eigen textures to identify where there are important changes in hemoglobin quantities. At each pixel, we sum up the absolute values of the first 4 eigen textures, and normalize to produce the mask shown in Figure 9. This is helpful because the pre-processed example images include areas which were masked and filled with a neutral skin texture to avoid the silhouette edges and background, and using this automatic alpha mask as a guide ensures that we exclude these regions in our blend. Ultimately, we edit this to produce the final alpha mask (e.g., we apply threshold, erosion, blur, and custom per finger modifications). The alpha mask is warped and flattened into the `stiaTex` image in the same process used for the red-green texture coordinate gradient.

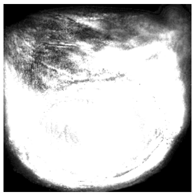

Figure 9: *Automatic alpha mask computed from PCA vectors.*

While we could let our model produce color changes on the base texture, we choose to replace the base texture with the warped zero pressure example image. This typically involves a small HSV space color correction to the base texture to have it match our white balanced example images. We use the alpha mask generated above to smoothly blend the pasted example into the base texture.

Note computing hemoglobin changes involves a chain of two texture lookups. This is a useful alternative to defining a new set of texture coordinates over the surface of the mesh. It allows for distortions in the mapping at the resolution of the texture, as opposed to the resolution of the mesh. This can be useful for closely matching small features, for instance, the length of fingernail may be longer or shorter in the geometry in comparison to our pre-processed images.

It is with the eigen textures that we compute hemoglobin changes, and we use the first four seen in Figure 5. These textures can be assigned to different texture units, but when only four are needed, it is efficient to pack them into the red, green, blue, and alpha components of a single floating point texture image. Note that in the previous section we use Matlab to compute the principal component analysis, and export the eigen textures as floating point tiff images, which are easily loaded into texture memory using OpenSceneGraph.

5.2 GLSL Fragment Program

Almost all the work of modifying the appearance of fingertips is done in a fragment program. Using the total force on each finger pad, we evaluate the interpolation function in Equation 6 to compute the PCA basis coefficients for each finger's hemoglobin quantity change. We have 4 values for each fingertip. Thus, we compute a total of 20 values and store the results in a `vec4` GLSL uniform array. The GLSL fragment program code necessary to compute the appearance modification is quite small and is provided in Figure 10.

```
// Fragment program
sampler2D baseTex;
sampler2D stiaTex;
sampler2D eignTex;
varying vec2 tex0;
uniform vec4 y[5];
uniform vec3 sigma_h;

vec3 base = sampler2D( baseTex, tex0.st );
vec4 stia = sampler2D( stiaTex, tex0.st );
vec4 U03  = sampler2D( eignTex, stia.st );
vec4 y03  = y[int(stia.b)];
float dq = dot( y03, U03 );
vec3 alpha = -log( base );
alpha += dq * sigma_h;
vec3 base2 = exp( -alpha );
base2 = clamp( base2, 0, 1 );
float mask = stia.a;
diffuseColor = mix( base, base2, mask );
// per pixel lighting follows...
```

Figure 10: *GLSL fragment shader code snippet summarizing the necessary additions to a standard per fragment lighting program.*

To summarize the process, it begins with a base color and `stia` value texture lookup, followed by a third texture lookup of the PCA basis at texture coordinates `st`. The negative log color of the base texture is computed, and then modified in the direction of σ_h by the hemoglobin concentration change. Note that σ_h is stored in a uniform and only needs to be set once. The hemoglobin concentration change is computed with a simple dot product of the PCA coordinates and the 4 PCA basis vector components associated with the fragment. Note that some care is necessary when building the `stia` texture to ensure that the blue component can be correctly cast to an integer. The final diffuse color is computed by exponentiating the modified negative log color. The color is clamped to $[0, 1]$, and blended with the base color using the alpha mask.

5.3 Grasping Control and Simulation

The hand grasping animation is simulated with CMLabs' Vortex real-time physics simulation software. We use a variety of simple proportional derivative (PD) feed-forward control trajectories to produce animation, as well as a grip strength controller that works from a single grasping pose based on the method of Liu [2009].

Our grip strength controller permits the computation of forces and torques at the fingertips that will allow for a simple modulation of grasp strength. From the forces and torque at contacts, we compute control torques at the joints by multiplying with the Jacobian transpose [Murray et al. 1994]. This lets us set a reference pose for a proportional derivative controller. We use reasonable joint stiffness and damping values, which are set using the object-in-hand drop test suggested by Pollard and Zordan [2005].

When there is one or more contacts at a finger pad, we sum up the force contributions, and use the magnitude in the PCA coordinate interpolation functions. We note that even when the simulation's collision restitution is set to zero (i.e., no bounce), there can be a spike in the contact force which results in a brief white flash on the fingertips. In reality, the appearance changes should be gradual due to the fact that blood must physically move within the fingertip. Therefore, we add viscosity in the form of an exponential decay, which we apply to the force magnitude (conceptually it would make more sense to do this on the PCA coordinates, but there is no

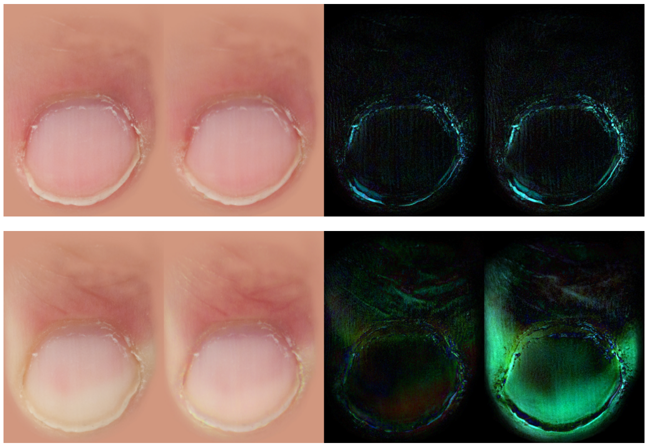

Figure 11: *Leave-one-out cross validation of our synthesis approach with testing examples at 0.18 N (top) and 9.25 N (bottom). From left to right are the example images, synthesized images, the RGB error magnitude of the synthesized image, and the RGB error of the naive solution (errors scaled* $4\times$ *to reveal detail).*

difference in our model). Our simulation runs at 100 Hz, and we note pleasing results by blending 0.95 of the previous time step's force magnitude with 0.05 of the current.

6 Validation and Results

In this section we discuss three types of validation that we have performed on our fingertip appearance modeling efforts: leave one out cross validation of interpolated reconstructions, model transferability between fingers and subjects, and a verification of hemoglobin pigment consistency. We also show results from interactive simulations of grasping.

Figure 11 shows a comparison of images synthesized with interpolation compared to pre-processed example images that were left out of the data analysis and fitting. The two examples show that our method successfully interpolates examples involving both small and large contact forces with little error. We compare our method to the naive solution, which simply uses the base fingernail image for all interaction forces. The absolute error is computed as the mean squared difference in RGB values of the validation and synthesized images. The relative error, computed by a ratio of absolute errors, is 0.6559 at 0.18 N, giving just a moderate improvement over the naive method. However, at larger interaction forces, the benefit of our method becomes clear, as the relative error drops significantly, to 0.1183 at 9.25 N.

To evaluate the transferability of our model across fingers and subjects, we performed the model fitting steps on three separate sets of 10 sample images gathered from three subjects. Figure 12 shows the first principal component from each set. These first eigen textures share qualitative similarities, with most changes occurring toward the distal part of the fingernail, and in the skin to the left and right. The Pearson correlation coefficient of the first eigen texture between the first two subjects is 0.6033, indicating a moderate correlation, but note that these two eigen textures do not share a common parameterization (i.e., we would expect a stronger correlation if image warping was applied).

We have also evaluated the constancy of our hemoglobin pigment estimation across different subjects and example sets. The hemoglobin pigment vectors for the three subjects in Figure 12, number one on the left, two in the middle, and three on the left, were estimated to be

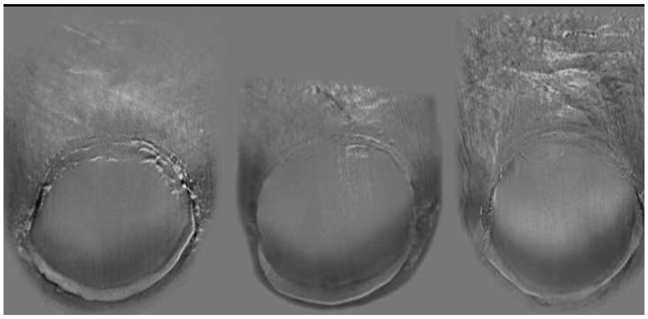

Figure 12: *The first principal component computed using different sets of example data from three different subjects. Shown are the index finger of subject one (left), the ring finger of subject two (middle) and the index finger of subject three (right).*

$$\begin{aligned} {}^1\sigma_h &= (0.1584, 0.7786, 0.6072), \\ {}^2\sigma_h &= (0.1905, 0.7810, 0.5948), \\ {}^3\sigma_h &= (0.2386, 0.7513, 0.6153). \end{aligned}$$

We note that the hemoglobin pigment vectors are remarkably similar, with ${}^1\sigma_h \cdot {}^3\sigma_h = 0.9964$, which is an angle of less than 5 degrees. The variance across the three pigment estimates is $(0.0279, 0.0184, 0.024)$.

Figure 13 shows an example simulation where a virtual hand simulation is grasping an object. When we increase the desired grip strength parameter of the controller, the hand posture changes to produce larger torques at the joints and in turn larger contact forces. We see a change in finger posture for the stronger grip, but also color changes at the fingertip due to blood distribution changes estimated by our data driven model. The effect can be seen more prominently in the supplementary video.

Figure 14 demonstrates both fingertip color changes and the interactivity of our physics based grasping simulation. A mouse-spring interface is used to apply forces on the wooden block, causing it to rotate. The compliant finger joints bend to accommodate the motion, while the fingertips show color changes due to the varying contact forces. Again, this example can be seen in the supplementary video, along with other examples.

7 Conclusion

Our work describes a method for changing the appearance of fingertips due to contact forces. This provides an important visual cue to show the force used to grasp an object, much like shadows give important information about the existence of contact. The PD controller of our hand simulation exhibits posture changes that provide an additional contact force cue. While deformations and wrinkles are also important cues that give an indication of contact force, we have left these aspects out of our work because many techniques already exist to deal with these issues. Ultimately, any of the contact deformation techniques proposed in previous work could be combined with our work to improve the overall quality of our results.

Construction of our data-driven model is straightforward. While there is a human effort in warping images to permit the construction of the PCA basis, it only needs to be done once because the model is transferable across fingers and subjects. Color changes are easily computed in a fragment program using a dot product, an

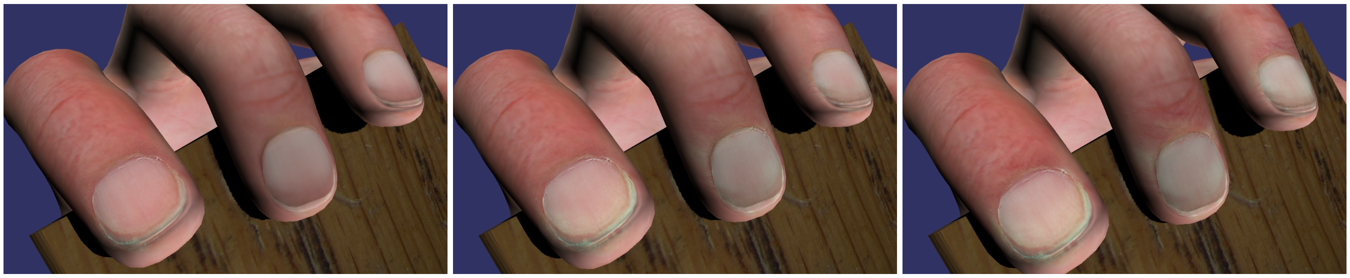

Figure 13: *A simulated grasping sequence where grip strength starts low and increases. Note the whitening of the nail while tissue near the cuticle takes on a more prominent red color.*

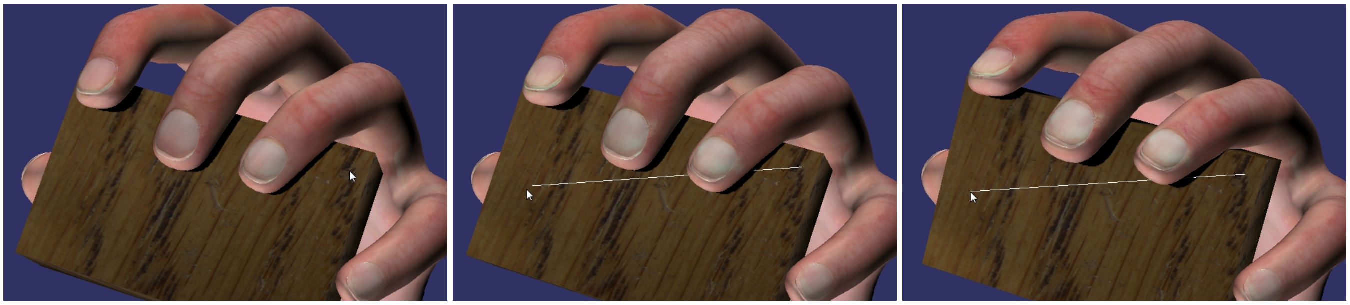

Figure 14: *A simulated interaction where a box can be pulled away from a grasp using a spring attached to the cursor location. Note the color changes at the middle and ring fingers that subtly indicate increased contact forces.*

addition (or subtraction) of hemoglobin pigment, and a logarithm and exponentiation to transform between colors and absorptions.

The inclusion of these appearance changes within a physics-based simulation of grasping helps demonstrate how these subtle cues can improve the overall interactive experience.

7.1 Future work

While we have mentioned previously that it would be natural to include the center of pressure and distal joint angle in our model of hemoglobin changes, we note that we could also measure the variance of the pressure distribution in our data collection process, and in turn create a model that can capture the difference between a sharp and flat contacts (for instance, the corner of a cube, in contrast to a face). Because our physics simulation uses rigid collision proxies, we would likewise need to rework the simulation to include soft elastic contact, or estimate the size of contact patches based on interpenetration volume.

Acknowledgements

We thank CFI, GRAND-NCE, NSERC, MITACS, CM Labs, and Autodesk for research equipment, operational support, and software donations.

References

ALADDIN, R., AND KRY, P. 2012. Static pose reconstruction with an instrumented bouldering wall. In *Proceedings of the 18th ACM symposium on Virtual reality software and technology*, ACM, VRST '12, 177–184.

BICKEL, B., BÄCHER, M., OTADUY, M. A., MATUSIK, W., PFISTER, H., AND GROSS, M. 2009. Capture and modeling of non-linear heterogeneous soft tissue. *ACM Trans. Graph. 28*, 3 (July), 89:1–89:9.

BONNEEL, N., VAN DE PANNE, M., PARIS, S., AND HEIDRICH, W. 2011. Displacement interpolation using lagrangian mass transport. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, SA '11, 158:1–158:12.

BORSHUKOV, G., MONTGOMERY, J., AND HABLE, J. 2007. *GPU Gems 3*. Addison-Wesley Professional, ch. 15. Playable Universal Capture.

BOUKHALFI, T. 2012. *Automatisation des expressions faciales liÃl'es Ãă lâĂŹactivitÃl' physique*. Master's thesis, ÃL'cole de technologie supÃl'rieure.

DONNER, C., WEYRICH, T., D'EON, E., RAMAMOORTHI, R., AND RUSINKIEWICZ, S. 2008. A layered, heterogeneous reflectance model for acquiring and rendering human skin. *ACM Trans. Graph. 27*, 5 (Dec.), 140:1–140:12.

DUTREVE, L., MEYER, A., AND BOUAKAZ, S. 2011. Easy acquisition and real-time animation of facial wrinkles. *Comp. Anim. Virtual Worlds 22*, 2-3, 169–176.

HUANG, H., ZHAO, L., YIN, K., QI, Y., YU, Y., AND TONG, X. 2011. Controllable hand deformation from sparse examples with rich details. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '11, 73–82.

JAKOVELS, D., SPIGULIS, J., AND ROGULE, L. 2011. RGB mapping of hemoglobin distribution in skin. In *Clinical and Biomedical Spectroscopy and Imaging II*, Optical Society of America, 80872B.

JENSEN, H. W., MARSCHNER, S. R., LEVOY, M., AND HANRAHAN, P. 2001. A practical model for subsurface light transport. In *Proceedings of the 28th annual conference on Com-*

puter graphics and interactive techniques, ACM, New York, NY, USA, SIGGRAPH '01, 511–518.

Jimenez, J., Scully, T., Barbosa, N., Donner, C., Alvarez, X., Vieira, T., Matts, P., Orvalho, V., Gutierrez, D., and Weyrich, T. 2010. A practical appearance model for dynamic facial color. In *ACM SIGGRAPH Asia 2010 papers*, SIGGRAPH ASIA '10, 141:1–141:10.

Kider, Jr., J. T., Pollock, K., and Safonova, A. 2011. A data-driven appearance model for human fatigue. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '11, 119–128.

Kry, P. G., and Pai, D. K. 2006. Interaction capture and synthesis. *ACM Trans. Graph. 25*, 3 (July), 872–880.

Kry, P. G., James, D. L., and Pai, D. K. 2002. Eigenskin: real time large deformation character skinning in hardware. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '02, 153–159.

Lewis, J. P., Cordner, M., and Fong, N. 2000. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, 165–172.

Liu, C. K. 2009. Dextrous manipulation from a grasping pose. In *ACM SIGGRAPH 2009 papers*, SIGGRAPH '09, 59:1–59:6.

Magnenat-Thalmann, N., Laperri‘ere, R., and Thalmann, D. 1988. Joint-dependent local deformations for hand animation and object grasping. In *Graphics Interface '88*, 26–33.

Mascaro, S., and Asada, H. 2004. Measurement of finger posture and three-axis fingertip touch force using fingernail sensors. *Robotics and Automation, IEEE Transactions on 20*, 1, 26–35.

Murray, R. M., Sastry, S. S., and Li, Z. 1994. *A Mathematical Introduction to Robotic Manipulation.* CRC Press.

Nishidate, I., Maeda, T., Niizeki, K., and Aizu, Y. 2013. Estimation of melanin and hemoglobin using spectral reflectance images reconstructed from a digital rgb image by the wiener estimation method. *Sensors 13*, 6, 7902–7915.

Oat, C. 2007. Animated wrinkle maps. In *ACM SIGGRAPH 2007 courses*, SIGGRAPH '07, 33–37.

Pai, D. K., Doel, K. v. d., James, D. L., Lang, J., Lloyd, J. E., Richmond, J. L., and Yau, S. H. 2001. Scanning physical interaction behavior of 3d objects. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, 87–96.

Pollard, N. S., and Zordan, V. B. 2005. Physically based grasping control from example. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 311–318.

Sloan, P.-P. J., Rose, III, C. F., and Cohen, M. F. 2001. Shape by example. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, I3D '01, 135–143.

Sueda, S., Kaufman, A., and Pai, D. K. 2008. Musculotendon simulation for hand animation. *ACM Transactions on Graphics 27*, 3 (Aug), 83:1–83:8.

Tsumura, N., Haneishi, H., and Miyake, Y. 1999. Independent component analysis of skin color image. *Journal of Optical Society of America A 16*, 9, 2169–2176.

Tsumura, N., Ojima, N., Sato, K., Shiraishi, M., Shimizu, H., Nabeshima, H., Akazaki, S., Hori, K., and Miyake, Y. 2003. Image-based skin color and texture analysis/synthesis by extracting hemoglobin and melanin information in the skin. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, 770–779.

Yin, K., and Pai, D. K. 2003. Footsee: an interactive animation system. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '03, 329–338.