A Simple Method for Modeling Wrinkles on Human Skin

Yosuke Bando[†] Takaaki Kuratate[‡] Tomoyuki Nishita[†]

[†]The University of Tokyo
7-3-1 Hongo, Bunkyo-ku,
Tokyo 113-0033, Japan
{ybando, nis}@nis-lab.is.s.u-tokyo.ac.jp

[‡]ATR Human Information Science Laboratories
2-2-2 Hikaridai, Seika-cho, Soraku-gun,
Kyoto 619-0288, Japan
kuratate@his.atr.co.jp

Abstract

Realism of rendered human skin can be strongly enhanced by taking into account skin wrinkles. However, modeling wrinkles is a difficult task, and considerable time and effort are necessary to achieve satisfactory results. This paper presents a simple method to easily model wrinkles on human skin, nevertheless taking into account the properties of real wrinkles. Wrinkles are specified using intuitive parameters, and are generated over a triangle mesh representing a body part, such as a hand or a face. Wrinkled skin surfaces are rendered at an interactive frame rate, dynamically modulating wrinkle amplitude according to skin surface deformation while animating the body part. We demonstrate the ability of our method to model realistic wrinkle shapes by comparing them with real wrinkles.

Keywords: *wrinkle simulation, human skin rendering, facial animation, human body simulation, texture mapping.*

1. Introduction

As virtual humans appear more frequently in various fields such as virtual reality, entertainment and surgical operation simulation, it becomes increasingly important to display them realistically. To do so, there are many things to consider: for example, modeling of body shape, animation, and material representations of skin, hair and clothes. Among these features, the skin is arguably the most subtle and complicated. It covers the entire human body. Therefore, realistic display of skin contributes significantly to the overall realism of virtual humans.

Wrinkling, together with skin color, is the main factor which determines the realism of skin rendering. Traditionally, however, rendering skin with wrinkles involves painstaking tasks such as manually drawing textures, and considerable time and effort are necessary to achieve satisfactory results. Our goal is to reduce this burden with a new simple method for modeling wrinkles.

Our method creates wrinkles on a triangle mesh representing a body part such as a hand or a face. Wrinkles can roughly be classified into two types: *fine-scale wrinkles* and *large-scale wrinkles* (see Section 3). Having different properties, each type of wrinkle is modeled in a different manner: fine-scale wrinkles are modeled by carving their furrows along a user-specified direction field over the body-part mesh; large-scale wrinkles are modeled by deforming the mesh based on specified location and shape of each wrinkle.

As this is a simple top-down approach, users can specify wrinkles using intuitive parameters (e.g., direction, depth and width) and easily model wrinkles as they desire. Thanks to their simplicity, the wrinkle generation algorithms are easy to implement and perform relatively fast. Wrinkled skin surfaces are rendered at an interactive frame rate, dynamically modulating wrinkle amplitude according to skin surface deformation while animating the body part. Nevertheless, the created wrinkles satisfactorily enhance the realism of skin rendering since the properties of real wrinkles are taken into account. For large-scale wrinkles, we demonstrate the ability of our method to model realistic wrinkle shapes by comparing them with real wrinkles.

In the following sections we describe the process in detail. Section 2 presents a review of related work. Section 3 explains the properties of wrinkles on human skin that our method takes into account. Section 4 describes how we construct the body-part mesh so that the wrinkle modeling method can be applied to it. Sections 5 and 6 describe modeling of fine-scale wrinkles and large-scale wrinkles respectively. Section 7 describes a method to modulate wrinkle amplitude while animating the body part. Section 8 shows results and Section 9 concludes.

Proceedings of the 10 th Pacific Conference on Computer Graphics and Applications (PG'02)
0-7695-1784-6/02 $17.00 © 2002 IEEE

2. Related work

Only a few methods have been proposed to create fine-scale wrinkles. Ishii et al. [8] divided skin surface into polygons by hierarchical Voronoi division, and rendered each edge of these polygons as a furrow of a wrinkle. Wu et al. [21] employed a hierarchical Delauney triangulation instead. Other studies captured and reconstructed details of the skin surface from actual skin samples by using a laser range scanner [14] and by image analysis [7].

Most of the works for large-scale wrinkles are intended for facial animation. Some studies modeled the skin as several layers, and animated the face by physically-based simulation [15, 20]. As the result of skin deformation, some kinds of wrinkles were formed. Boissieux et al. [2] simulated various kinds of wrinkles with a more sophisticated method, although on an abstract, simplified piece of skin. They also proposed an image-based approach in the same paper. Viaud and Yahia [16] modeled wrinkle bulges as spline segments. Wu et al. [21-23] presented a method to simulate facial animation with both fine- and large-scale wrinkles (one of these also dealt with static wrinkles on the hand [21]). Volino and Thalmann [18] rapidly animated wrinkles on deformable models by modulating the amplitude of a given wrinkle pattern. This work was further extended mainly for cloth wrinkling by Hadap et al. [6].

Our approach differs from these methods principally in that it provides easy control over wrinkle characteristics. In contrast, physically-based simulation does not facilitate direct manipulation of wrinkles. Moreover, it is difficult for users to modify the wrinkle patterns acquired from actual skin samples or images. The works by Volino and Thalmann, and Hadap et al. assumed wrinkle pattern textures to be drawn by the user, although Hadap et al. provided strain patterns in a garment as a guide. For large-scale wrinkles, we demonstrate the ability of our method to model realistic wrinkle shapes by comparing them with real wrinkles, whereas none of the above works for modeling wrinkles are guaranteed to have such abilities. Another difference is that our method dynamically modulates the amplitude of *both* fine- and large-scale wrinkles according to the skin surface deformation during animation.

3. Wrinkle properties

Wrinkles are the main factor determining the undulation of the skin surface. We can roughly distinguish two types of wrinkle and we define them as follows: *fine-scale wrinkles* are small wrinkles that cover the entire skin surface; *large-scale wrinkles* are distinct wrinkles unique to a particular body part (e.g., forehead wrinkles).

Fine-scale wrinkles have a similar structure over the entire body except palms and soles in the sense that their furrows intersect each other over the surface [12]. These furrows run in particular directions depending on the location on the skin. As shown in Figure 1, a close look at the skin surface reveals their properties to be the following. 1) Furrows run locally in two directions. 2) The intersection point of two furrows running in the same direction becomes the endpoint of either. The first property has some exceptions, but holds over most of the surface. This is also supported by the observation that the outline of intersecting furrows is roughly diamond-shaped [13].

Large-scale wrinkles are formed due to shrinking of the skin surface caused by body part movement. Skin material itself is incompressible [4], and therefore, when the skin surface shrinks, the excess skin buckles and forms wrinkles. These become gradually more prominent with repeated wrinkling and age. As shown in Figure 2, we can observe that large-scale wrinkles have the following property: each has a sharp furrow at its center line and round bulges on its both sides.

Both types of wrinkle become more pronounced when the skin surface shrinks perpendicularly to their directions.

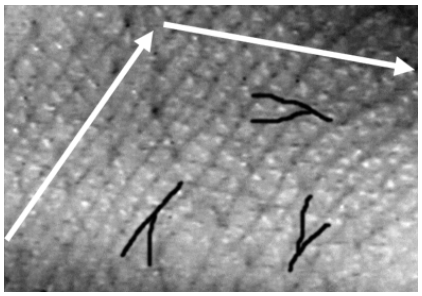
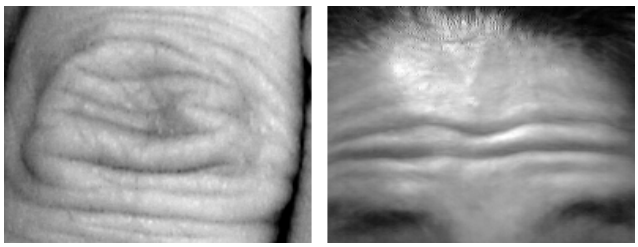


Figure 1: A photograph of fine-scale wrinkles on the back of the hand. White arrows show two wrinkle directions. Black lines show some of the intersection points of wrinkle furrows.



(a) (b)

Figure 2: Photographs of large-scale wrinkles around the finger joint and on the forehead.

4. Construction of body-part mesh

We use a triangle mesh to represent a body part. This

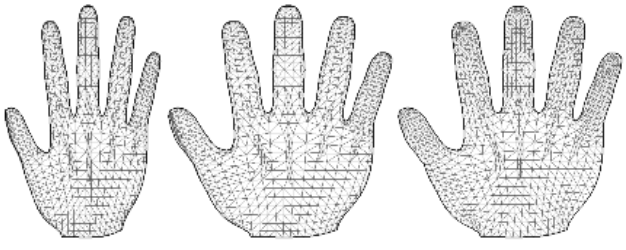
Proceedings of the 10 th Pacific Conference on Computer Graphics and Applications (PG'02)
0-7695-1784-6/02 $17.00 © 2002 IEEE

section describes how to construct a mesh so that the wrinkle modeling methods described in the following sections can be applied to it.

The geometry of the mesh is modeled with CG software, or acquired with a range scanner. The resolution of the mesh can be arbitrary. If the mesh is not fine enough to represent large-scale wrinkles, we apply the adaptive refinement proposed by Volino and Thalmann [18].

The mesh is mapped to the two-dimensional texture space in order to make the wrinkle modeling process simpler and faster (see Sections 5 and 6). The mapping should be such that two distinct points on the mesh do not correspond to a single point in the texture space (one to one mapping), and that the topology of the mapped mesh is the same as that of the original mesh (topology preserving). We use planar or cylindrical projection depending on the shape of the body part. Cylindrical projection may produce a seam in the mapped mesh, but this will not be a problem if we treat the texture space as wrapped around. If distortion of the current projected mesh is not negligible, we fix the boundary vertices of the mesh, and smooth the internal vertices by minimizing the total energy of springs placed along the edges. We set spring constants in the same way as Eck et al. [5] constructed harmonic maps. In our case, however, the boundary is not generally convex as opposed to the requirement for harmonic maps, but this method works fine if we move some vertices so that the boundary becomes less concave. Figure 3 shows this distortion minimization process. The mesh is scaled as shown in Figure 3b so that it better fits the unit square texture space.

In addition, the way of animating the body-part mesh is defined. We employ relatively simple and fast methods in order to display the skin with wrinkles at an interactive frame rate. Specifically, most body parts are animated with articulated skeletons [17], while the face is animated with the facial muscle model proposed by Waters [19].



(a) (b) (c)

Figure 3: (a) The hand mesh mapped with planar projection. (b) The mesh modified by scaling and by moving some boundary vertices. (c) The mesh after distortion minimization.

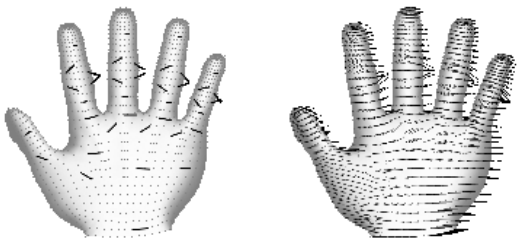
5. Modeling of fine-scale wrinkles

As described in Section 3, fine-scale wrinkles cover the entire skin surface, and are similarly structured in the sense that their furrows run along particular directions depending on their location on the skin. To simulate these characteristics, wrinkle furrows are generated along a direction field over the body-part mesh.

Wrinkle modeling procedure is performed in the texture space (see Section 4). The user first specifies a direction field to indicate the wrinkle directions. Next, wrinkle furrows are carved along the direction field, and the resultant undulation of the skin surface is recorded as a height field. Wrinkle characteristics are controlled by the magnitude of the direction field. Following the wrinkle property that the furrows of fine-scale wrinkles run locally in two directions (see Section 3), this procedure is performed independently for the two different direction fields, and two height fields are recorded as shown in Figure 6. These are superimposed, and the skin with fine-scale wrinkles is displayed by bump-mapping [1].

5.1. Formation of direction field

The direction field is used to indicate the directions of fine-scale wrinkles. It is formed in the texture space, and is represented as a direction vector at each vertex of the mapped mesh. The user first specifies direction vectors at several vertices as shown in Figure 4a. Note that the mesh is mapped to the texture space, but it is shaded based on its geometry to inform the user of its shape. Next, direction vectors at the rest of the vertices are interpolated from these vectors. For interpolation, again we minimize the spring energy similarly to the mapping distortion minimization described in Section 4. In this case, however, the values associated to the vertices are direction vectors instead of texture coordinates, and we fix the user-specified direction vectors instead of the texture coordinates of the boundary vertices. The resultant vectors form a direction field as shown in Figure 4b.



(a) (b)

Figure 4: (a) The direction vectors specified at several vertices of the hand mesh. (b) The direction field formed after interpolation.

Proceedings of the 10 th Pacific Conference on Computer Graphics and Applications (PG'02)
0-7695-1784-6/02 $17.00 © 2002 IEEE

5.2. Carving of wrinkle furrows

Wrinkles are generated by carving their furrows along the direction field. Since we work in the texture space, the undulation of the skin surface due to these furrows is represented as a height field. This height field is defined as *height image*, a grayscale image whose intensity represents the height value. This image is initialized in white, and wrinkle furrows are carved by drawing gray line segments on it. In addition, we use another image called *occupancy image* to control the distance between adjacent furrows. This is a binary image which records the regions occupied by the already generated furrows. We control the distance between furrows by forbidding new furrows to be generated from these regions. The occupancy image is initialized to 0 (not forbidden) where the mesh is mapped, and 1 (forbidden) otherwise, because we do not want to generate wrinkles that have no correspondence to the mesh. As most graphics systems have double buffers for animation, we use one for the height image, and the other for the occupancy image.

A single wrinkle furrow is generated from a given point P in the texture space by the algorithm in the following sequence of steps (also see Figure 5). Note that each vertex has a direction vector which indicates the wrinkle direction. n holds the number of furrow segments to generate, and is initialized to infinity. The way to determine the parameters of wrinkle, N, θ, D, I, W_h and W_o, is described in Section 5.3.

1. Obtain wrinkle direction **d** at point P by interpolating from three vertices of the triangle to which P belongs.
2. Compute N, the number of furrow segments to generate from point P. Set n to $\min(n, N)$.
3. Add perturbation to wrinkle direction **d** by rotating it by angle θ.
4. Let Q be the point at distance D from point P in direction **d**. If segment PQ intersects other wrinkle furrows, reset Q to the intersection point closest to P, and set n to 0.
5. Carve one segment of the wrinkle furrow by drawing line segment PQ on the height image with intensity I and width W_h.
6. Record the region occupied by the furrow by drawing line segment PQ on the occupancy image with value (or 'intensity') 1 and width W_o.
7. Stop if $n = 0$. Otherwise reset P to Q, decrement n by 1, and return to step 1.

We also apply this algorithm to the opposite direction field, setting **d** to **−d**. Therefore, a wrinkle furrow grows to both sides of the starting point. In step 2, the number of furrow segments that can be generated from the current point is computed. If this number (denoted by N) is smaller than n, then n is reset to N. This determines how far the furrow grows depending on its location. In step 3, perturbation is added to the wrinkle direction to generate a natural furrow which otherwise becomes regular. In step 4, one furrow segment is determined. Distance D corresponds to its length. The latter half of step 4 follows one of the wrinkle properties described in Section 3: the intersection point of two furrows becomes the endpoint of either. If the current furrow intersects another furrow that has already been generated, it terminates its growth there as shown in Figure 5b-2. In step 5, one furrow segment is carved. Intensity I and width W_h correspond to its depth and width, respectively. In step 6, the region occupied by the generated furrow segment is recorded so that no other furrow is generated from there (since the occupancy image is binary, the region is recorded as value 1). Thus a larger W_o keeps other furrows further away from the current one. In step 7, if the number of furrow segments to generate is 0, the algorithm terminates. Otherwise P is set to point Q, the current endpoint of the furrow, as a new starting point from which the furrow grows. n is decremented by 1 because one segment was generated, and we return to step 1 to generate more segments.

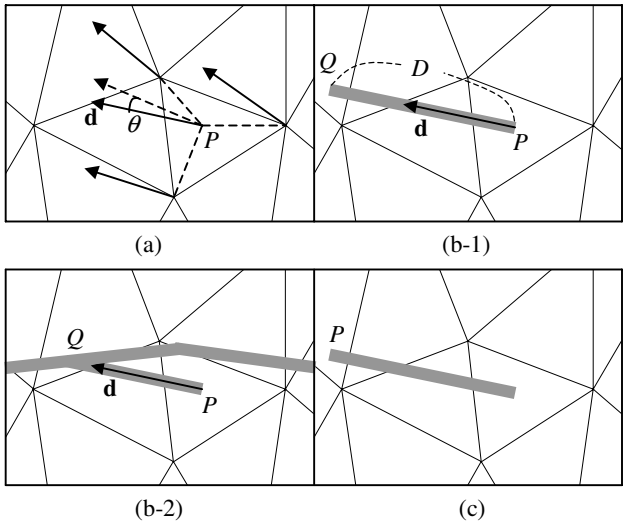


Figure 5: (a) Steps 1-3. The wrinkle direction obtained by interpolation and perturbation. (b) Steps 4-6. Generated one segment of the wrinkle furrow. (b-1) With no intersection with other furrows. (b-2) With intersection. (c) Step 7. A new starting point of the furrow preceded by the case shown in (b-1).

Proceedings of the 10 th Pacific Conference on Computer Graphics and Applications (PG'02)
0-7695-1784-6/02 $17.00 © 2002 IEEE

Using the above algorithm, wrinkle furrows are generated from a sequence of starting points until they cover the entire mesh in the texture space. We randomly select point P on the mapped mesh, and generate a furrow from P if the value of the occupancy image at P is 0. This process is iterated until many furrows fail to be generated. Now that the mesh is roughly covered with wrinkles, we fill in the rest by generating furrows from each point in the texture space whose value is 0 in the occupancy image. The way of selecting starting points does not make much difference in the final result as long as the point distribution is relatively uniform. Figure 6a shows the resultant height field for the direction field shown in Figure 4b.

(a) (b)

Figure 6: Two height fields over the hand. Each represents the undulation of the skin surface due to the wrinkle furrows generated along two different direction fields.

5.3. Controlling Wrinkle Characteristics

We introduced six parameters that control wrinkle characteristics in the generation algorithm described above. N: the number of furrow segments, θ: perturbation angle for the wrinkle direction, D: length of the furrow segment, I: depth of the furrow, W_h: width of the furrow, and W_o: distance between adjacent furrows. All these values are determined according to the magnitude of the direction field. If the user specifies the direction vector with larger magnitude at a particular vertex, wrinkle furrows around the area get more strongly oriented. That is, they become longer (having many long segments), deeper and wider, with smaller irregularity (smaller perturbation) and with larger distance to adjacent furrows. In our empirical model, Equation 1 defines these six parameters as functions of magnitude m of the direction vector at the point from which the furrow grows. m is normalized to be in the range [0, 1].

$$\begin{cases} N = \lfloor 30 - 20m \rfloor \\ \theta = (0.5 - 0.4m) \cdot r \\ D = 4 + 2m \end{cases} \qquad \begin{cases} I = 1 - m^2 \\ W_h = 2m \\ W_o = 1 + 12m \end{cases}, \qquad (1)$$

where r is a random number in the range (-1, 1). θ is expressed in radians. The unit length for D, W_h and W_o is usually set to the size of a pixel in the height image and the occupancy image. However, this depends on the size of the images and the body part, and only the relative values are important. The average magnitude of the direction vectors shown in Figure 4a is about 0.5.

5.4. Wrinkles over multiple body parts

It is not practical to generate wrinkles over a whole body mesh at once because this requires enormous texture size and also makes it difficult to map the mesh to the texture space with small distortion. Therefore, the mesh is divided into several body parts, and fine-scale wrinkles are generated over each of them. In order to generate seamless height fields between two neighboring body parts, the divided body-part meshes share their boundary portion. After wrinkles are generated over one body part, the direction field and the height field over the shared portion of the mesh are copied to the other body-part mesh. Starting with this initial state, wrinkles are generated over the other body part.

6. Modeling of large-scale wrinkles

As described in Section 3, large-scale wrinkles are distinct, local wrinkles formed due to shrinking of the skin surface. Therefore, we have the user specify wrinkles one by one, by their locations and shapes. The body-part mesh is then deformed based on this information to represent the wrinkles. The reason for not displaying them by bump-mapping is that the undulation due to these wrinkles is relatively large in comparison to the size of the body part.

Location of a wrinkle is specified in the texture space by drawing a cubic Bezier curve as a furrow of the wrinkle as shown in Figure 7. The cross sectional shape of the wrinkle is given as *wrinkle shape function* $S(l)$, a height function of distance l from the specified furrow. l is measured in perpendicular direction to the furrow in the texture space as shown in Figure 8a. We adopt Equation 2 as the wrinkle shape function for the following three reasons.

$$S(l) = d(l/w - 1)\exp(-l/w). \qquad (2)$$

First, as shown in Figure 8b, it simulates the wrinkle property that the furrow of a wrinkle is sharp and the bulges round (see Section 3). Second, it satisfies the following equation.

Proceedings of the 10 th Pacific Conference on Computer Graphics and Applications (PG'02)
0-7695-1784-6/02 $17.00 © 2002 IEEE

$$\int_0^\infty S(l)\,dl = 0\,. \tag{3}$$

This means that the volume depressed at the furrow goes to the side to make up the bulge, which simulates the incompressibility of skin. Third, it is easily controlled by two intuitive parameters, depth d and width w.

To represent wrinkles, the body-part mesh is deformed by displacing its vertices along their normal vectors according to the wrinkle shape function. If the mesh is not fine enough to represent wrinkles, we apply the adaptive refinement [18] around the specified wrinkle locations. Displacing only vertices near the wrinkle locations improves computational efficiency, and wrinkles are generated on the fly. Therefore, users can see what happens immediately after they modify the parameters assigned to each wrinkle.

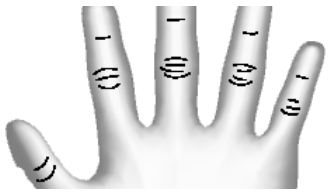


Figure 7: The wrinkle locations specified around the finger joints of the hand.

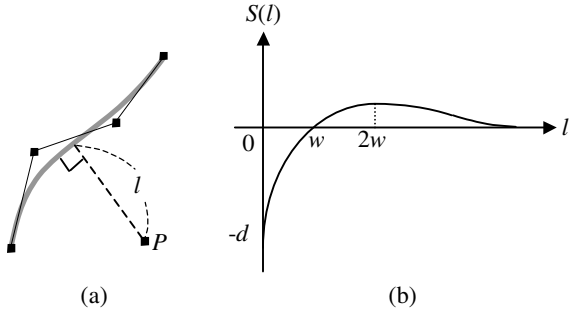


Figure 8: (a) A cubic Bezier curve representing the location of a wrinkle furrow in the texture space. Distance from the furrow is measured in its perpendicular direction. (b) The wrinkle shape function.

7. Modulation of wrinkle amplitude

While animating the body part, both fine- and large-scale wrinkles are also animated by dynamically modulating their amplitude according to skin surface deformation. To measure the deformation, we compute shrinkage s of the skin surface at each vertex of the body-part mesh. Shrinkage inside the triangle of the mesh is interpolated from its three vertices. Then we modulate wrinkle amplitude by *the amplitude modulation function* $M(s)$.

The following explains how to compute skin shrinkage of the animated mesh. As described in Section 3, wrinkles become more pronounced when the skin surface shrinks perpendicularly to their directions. The direction of a fine-scale wrinkle is given as the direction vector assigned at the vertex (see Section 5), and the direction of a large-scale wrinkle is given as the tangential vector of the cubic Bezier curve indicating the wrinkle location (see Section 6). Shrinkage is measured in its perpendicular direction, $\mathbf{d}_p$, in the texture space. Figure 9a shows the vertex of the mesh (denoted by v_1) in the texture space for which we want to compute shrinkage, and we consider triangle $v_1v_2v_3$ of the mesh located in direction $\mathbf{d}_p$ from v_1. The original length L of the triangle is defined as the distance between vertex v_1 and edge v_2v_3 in direction $\mathbf{d}_p$ as shown in Figure 9a. Suppose that the length of edge v_1v_2 and that of edge v_1v_3 are scaled by t_2 and t_3, respectively in the three-dimensional object space as the result of animating the body-part mesh. Assuming that the lengths of these edges are scaled by the same values in the texture space, the current length L' of the triangle is defined similarly as shown in Figure 9b (however, we do not actually change the texture coordinates of vertices v_2 and v_3). From these values, we compute shrinkage s as follows.

$$s = \frac{L - L'}{L} = 1 - \frac{L'}{L}\,. \tag{4}$$

s is the amount of shortening relative to the original length, which equals 0 in the rest state (i.e., the state in which the mesh is not deformed). s becomes negative when the skin expands, and increases up to the maximum value of 1 as the skin shrinks. Generally, there are two triangles located in the direction perpendicular to the wrinkle furrow, in direction $\mathbf{d}_p$ and $-\mathbf{d}_p$ from v_1. In this case, we average two shrinkage values. Note that shrinkage s is dependent on wrinkle directions, and varies for different wrinkles even if it is measured at the same location. This realizes anisotropic wrinkle response to skin surface deformation.

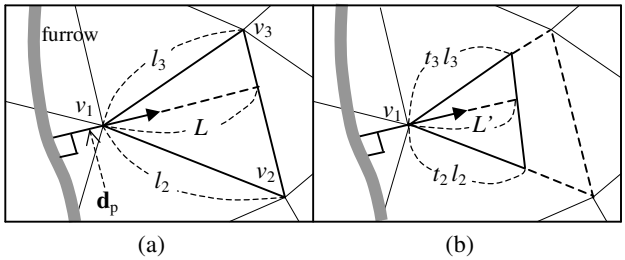


Figure 9: (a) The original length of a triangle measured in the texture space perpendicularly to the wrinkle direction. (b) The current length of the triangle. The lengths of two edges in the texture space are assumed to be scaled by the same values as in the three-dimensional object space.

Proceedings of the 10 th Pacific Conference on Computer Graphics and Applications (PG'02)
0-7695-1784-6/02 $17.00 © 2002 IEEE

According to the shrinkage s, wrinkle amplitude (or height) is scaled by the amplitude modulation function $M(s)$ defined as Equation 5. Figure 10 shows its graph. It is C^1 continuous at $s = 0$. Parameter r, in the range [0, 1], controls the initial height of a wrinkle relative to its maximum height. Wrinkles become more pronounced when the skin surface shrinks, and less when it expands. If $r = 0$, $M(s) = 0$ for $s < 0$, and wrinkles do not appear unless the skin surface shrinks.

$$M(s) = \begin{cases} (1-r)\cdot s + r & s \geq 0 \\ r\exp\{(1/r-1)\cdot s\} & s < 0 \end{cases} . \quad (5)$$

For large-scale wrinkles, parameter r is assigned to each wrinkle. The vertices relevant to a particular wrinkle are displaced according to the wrinkle shape function (see Section 6) and the amplitude modulation function with the parameters assigned to this wrinkle. For fine-scale wrinkles, parameter r is set to non-zero value (usually 0.5) everywhere in the texture space since the wrinkles are already visible in the rest state. As described in Section 5, we have two height field textures for the two different direction fields. Amplitude modulation is performed for each texture with its corresponding direction field, and the two modulated textures are superimposed and bump-mapped.

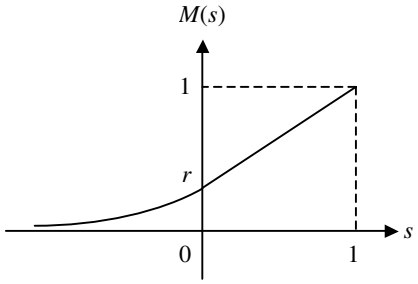


Figure 10: The amplitude modulation function.

8. Results

Figures 15a and 15b show examples of the hand with and without wrinkles. Note that the created wrinkles satisfactorily enhance the realism of skin rendering. The original mesh consists of about 4,000 triangles, and we applied the adaptive refinement in order to represent the large-scale wrinkles specified around the finger joints. As a result, about 34,000 triangles were rendered. The values of the shape parameters d and w for these wrinkles range from 0.5 mm to 1 mm. We set parameter r to 1 for all of them because the fingers of the hand mesh are stretched at the rest state and the wrinkles should be at their maximum height. Figures 15c and 15d show the simulated skin surface of the animated hand. Bump-mapping is enhanced excessively so that fine-scale wrinkles are clearly visible. The shapes of wrinkles running in the direction of skin deformation remain almost the same, whereas those of wrinkles running perpendicularly become less pronounced as the skin surface expands.

Figure 16 shows examples of the face. Large-scale wrinkles serve as expressive wrinkles on the face, making the face more expressive. The face mesh was acquired by scanning a subject's face with neutral expression using a color 3D scanner 3030RGB/PS (Cyberware). It consists of about 20,000 triangles, and is fine enough to represent large-scale wrinkles. The parameter values for d and w range from 2 mm to 12 mm, and r is set to 0 so that the wrinkles do not appear at neutral expression. In addition, these figures show examples of various skin colors. Skin colors in Figures 16a and 16b were rendered taking into account the amount of skin pigment. Possible skin colors lie on a two-dimensional surface patch within RGB color space, with two axes corresponding to the amount of melanin and hemoglobin [3]. We simplified the surface patch to be planar, and determined skin color $\mathbf{c}$ by the amount of melanin m and that of hemoglobin h as follows.

$$\mathbf{c} = \mathbf{b} - m\mathbf{a}_m - h\mathbf{a}_h, \quad (6)$$

where $\mathbf{b}$ is the base color of skin, and $\mathbf{a}_m$ and $\mathbf{a}_h$ are the light absorption ratios of melanin and hemoglobin, respectively. Their values were determined empirically. We used these colors as ambient and diffuse reflectivity for rendering. The color in Figure 16b is due to large amount of melanin. Colors in Figures 16c and 16d were rendered by the Lafortune model [10] with the BRDF data of skin measured by Marschner et al. [11].

Figure 17 shows other examples including the foot and the hand with the arm. These were rendered with other objects and shadows at the cost of rendering time. Figures 13 and 14 show the aged skin on the hand and the face with prominent wrinkles in the rest state.

For large-scale wrinkles, we demonstrate the ability of our method to model realistic wrinkle shapes by comparing them with real wrinkles. By using a 3D digitizer VIVID700 (Minolta Co.), we digitized real skin surfaces of a forehead when it was wrinkled (Figure 12a) and when it was not (Figure 11). We specified large-scale wrinkles on the mesh shown in Figure 11, and wrinkles were generated after animating the forehead mesh to shrink the skin surface. Figure 12b shows the resultant mesh. We compared the two meshes shown in Figure 12 by measuring the distance from each vertex of one mesh to the nearest triangle of the other mesh. The average distance was 0.5 mm, which is relatively small in comparison to the depth of forehead wrinkles (about 5 mm).

Proceedings of the 10 th Pacific Conference on Computer Graphics and Applications (PG'02)
0-7695-1784-6/02 $17.00 © 2002 IEEE

Our method generates and renders wrinkles relatively fast. Fine-scale wrinkles are generated in about 10 seconds. And a body-part mesh with fine- and large-scale wrinkles is animated and rendered at an interactive frame rate. Large-scale wrinkles are generated on the fly. Therefore, users can see what happens immediately after they modify the wrinkle parameters. The hand is displayed at 6.4 fps, and the face at 5.0 fps on a XEON 2GHz with a Wildcat II 5110 graphics card. Image size and texture size are 512 x 512.

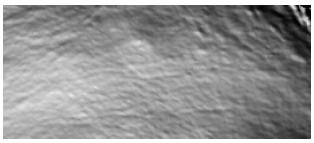
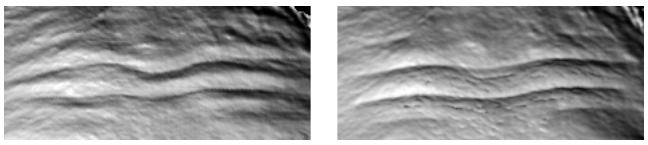


Figure 11: A digitized forehead with no wrinkles.



(a) (b)

Figure 12: (a) A digitized forehead with wrinkles. (b) Generated wrinkles over the mesh shown in Figure 11.

9. Conclusion

This paper has presented a simple method to easily model wrinkles on human skin. The advantages of our method are:

- It provides easy control over wrinkle characteristics. Users can specify wrinkles using intuitive parameters, and easily model wrinkles as they desire.
- It is easy to implement thanks to its simplicity.
- It performs relatively fast. Wrinkled skin surfaces are rendered at an interactive frame rate.
- It dynamically modulates the amplitude of *both* fine- and large-scale wrinkles according to the skin surface deformation while animating the body part.
- The created wrinkles satisfactorily enhance the realism of skin rendering since our method takes into account the properties of real wrinkles.

For large-scale wrinkles, we have demonstrated the ability of our method to model realistic wrinkle shapes by comparing them with real wrinkles. This lends support to both the first and the fifth advantages listed above. On the other hand, the limitations of our method are the following:

- It cannot be applied to fine-scale wrinkles on palms and soles because they have different properties from those on the rest of the skin surface.
- It ignores self-contact of large-scale wrinkles that might occur on the inside of joints. Though this may visually be acceptable because wrinkles are hidden in the folds, it is better to take it into account.
- It does not model the underlying tissues of skin. However, we intend our method to be used for any animated body part regardless of its animation method, thus this problem is beyond our scope.
- It provides too much freedom for modeling wrinkles. It is better to let users model wrinkles with some constraints by inferring wrinkle locations and directions from the skin surface deformation.

In the future, in addition to addressing the above limitations, we would like to add more details to further enhance realism by displaying not only wrinkles, but also other elements on the skin (e.g., pores, hairs, blood vessels), and also by simulating subsurface scattering of skin as was done by Jensen et al. [9].

Acknowledgement

We would like to thank the members of ATR Human Information Science Laboratories Department 2 for providing materials and a work environment, and for correcting our English. We would like to acknowledge the BRDF data of human skin from the Cornell University Program of Computer Graphics.

References

[1] J. Blinn. Simulation of Wrinkled Surface. *Proc. SIGGRAPH 78*, 286-292, 1978.

[2] L. Boissieux, G. Kiss, N. M. Thalmann and P. Kalra. Simulation of Skin Aging and Wrinkles with Cosmetics Insight. *Proc. Eurographics Workshop on Computer Animation and Simulation 2000*, 15-27, 2000.

[3] S. Cotton, E. Claridge and P. Hall. A Skin Imaging Method Based on a Color Formation Model and its Application to the Diagnosis of Pigmented Skin Lesions. *Proc. Medical Image Understanding and Analysis '99*, 49-52, 1999.

[4] J. Doyle and J. Philips. *Manual on Experimental Stress Analysis (fifth edition)*. Society for Experimental Mechanics, 1989.

[5] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery and W. Stuetzle. Multiresolution Analysis of Arbitrary Meshes. *Proc. SIGGRAPH 95*, 173-182, 1995.

Proceedings of the 10 th Pacific Conference on Computer Graphics and Applications (PG'02)
0-7695-1784-6/02 $17.00 © 2002 **IEEE**

[6] S. Hadap, E. Bangerter, P. Volino and N. M. Thalmann. Animating Wrinkles on Clothes. *Proc. IEEE Visualization '99*, 175-182, 1999.

[7] A. Haro, B. Guenter and I. Essa. Real-time, Photo-realistic, Physically Based Rendering of Fine Scale Human Skin Structure. *Proc. Eurographics Workshop on Rendering Techniques 2001*, 53-62, 2001.

[8] T. Ishii, T. Yasuda, S. Yokoi and J. Toriwaki. A Generation Model for Human Skin Texture. *Proc. Computer Graphics International '93*, 139-150, 1993.

[9] H. W. Jensen, S. R. Marschner, M. Levoy and P. Hanrahan. A Practical Model for Subsurface Light Transport. *Proc. SIGGRAPH 2001*, 511-518, 2001.

[10] E. P. F. Lafortune, S. C. Foo, K. E. Torrance and D. P. Greenberg. Non-Linear Approximation of Reflectance Functions. *Proc. SIGGRAPH 97*, 117-126, 1997.

[11] S. R. Marschner, S. H. Westin, E. P. F. Lafortune, K. E. Torrance and D. P. Greenberg. Image-Based BRDF Measurement Including Human Skin. *Proc. Eurographics Workshop on Rendering '99*, 139-152, 1999.

[12] W. Montagna, A. M. Kligman and K. S. Carlisle. *Atlas of Normal Human Skin.* Springer, 1992.

[13] S. L. Moschella and H. J. Hurley. *Dermatology (third edition) volume 1.* W. B. Saunders Company, 1992.

[14] M. Nahas, H. Huitric, M. Rioux and J. Domey. Facial Image Synthesis Using Skin Texture Recording. *The Visual Computer 6*, 6, 337-343, 1990.

[15] D. Terzopoulos and K. Waters. Physically-Based Facial Modeling, Analysis, and Animation. *Journal of Visualization and Computer Animation 1*, 73-80, 1990.

[16] M. L. Viaud and H. Yahia. Facial Animation with Wrinkles. *Proc. Eurographics Workshop on Animation and Simulation '92*, 1-13, 1992.

[17] J. Vince. *Essential Computer Animation Fast.* Springer, 2000.

[18] P. Volino and N. M. Thalmann. Fast Geometrical Wrinkles on Animated Surfaces. *Proc. WSCG '99, 1999.*

[19] K. Waters. A Muscle Model for Animating Three-Dimensional Facial Expression. *Proc. SIGGRAPH 87*, 17-24, 1987.

[20] Y. Wu, N. M. Thalmann and D. Thalmann. A Plastic-visco-elastic Model for Wrinkles in Facial Animation and Skin Aging. *Proc. Pacific Graphics '94*, 201-213, 1994.

[21] Y. Wu, P. Kalra and N. M. Thalmann. Simulation of Static and Dynamic Wrinkles of Skin. *Proc. Computer Animation '96*, 90-97, 1996.

[22] Y. Wu, P. Kalra and N. M. Thalmann. Physically-Based Wrinkle Simulation & Skin Rendering. *Proc. Eurographics Workshop on Computer Animation and Simulation '97*, 69-79, 1997.

[23] Y. Wu, P. Kalra, L. Moccozet and N. M. Thalmann. Simulating Wrinkles and Skin Aging. *The Visual Computer 15*, 4, 183-198, 1999.

Figure 13: The aged skin on the hand.

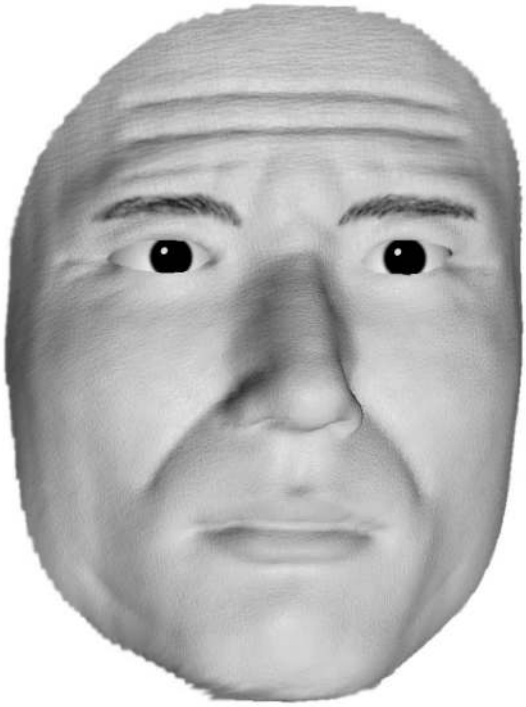
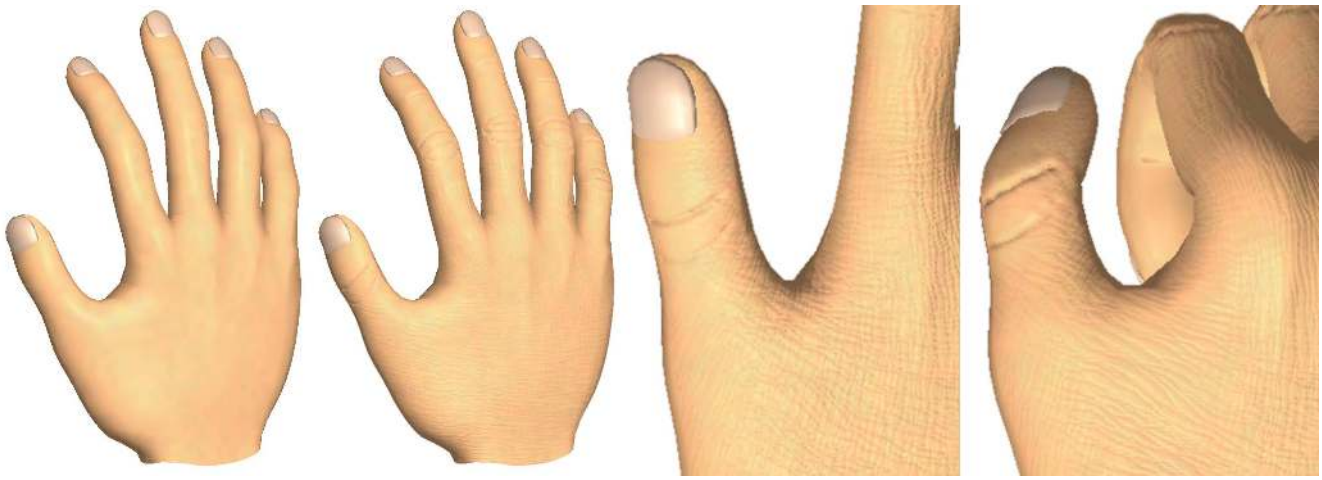


Figure 14: The aged skin on the face.

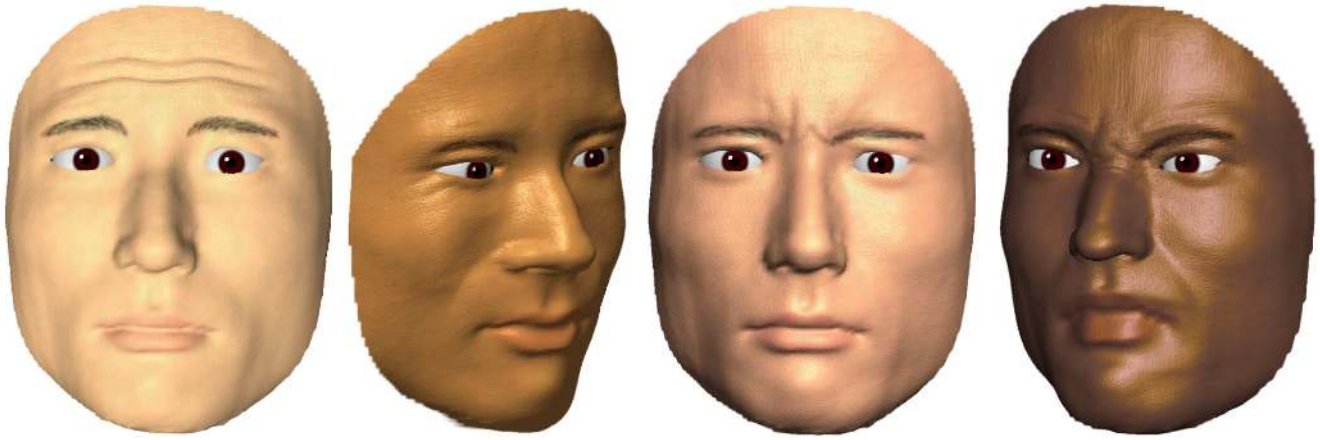
Proceedings of the 10 th Pacific Conference on Computer Graphics and Applications (PG'02)
0-7695-1784-6/02 $17.00 © 2002 **IEEE**

This is a color page.



(a) (b) (c) (d)

Figure 15: (a) The hand without wrinkles. (b) With wrinkles. (c)(d) Animated fine-scale wrinkles.



(a) (b) (c) (d)

Figure 16: Various facial expressions with various skin colors.

(a) (b)

Figure 17: (a) The foot. (b) The hand with the arm.

Proceedings of the 10 th Pacific Conference on Computer Graphics and Applications (PG'02)
0-7695-1784-6/02 $17.00 © 2002 **IEEE**