

## Visual Simulation of Hand Touching Introducing Elastic Deformation and Color Change

Youhei ISHIGURO<sup>†</sup>, Tsukasa KIKUCHI<sup>†\*</sup>

Takushoku University

E-mail: {yishigur<sup>†</sup>,tkikuchi<sup>†\*</sup>}@id.takushoku-u.ac.jp

**Abstract:** The hands are the most characteristic part of the human body from the standpoint that they are related to other elements through actions like grasping and touching. Several elaborate animation techniques for the hand with 3DCG have recently been proposed, but they are not good enough for simulations of hand touching, which requires simulation of complicated actions and position-dependent elastic forces in the hand. In this report, an algorithm is proposed which the user can operate intuitively for the deformation and the color change of the hand induced by touching. An outline of the algorithm is as follows; 1) elastic deformation, depending on the position of the hands, is controlled by channel images, and 2) color change is expressed by synthesizing three kinds of texture images.

**Key words:** *Human hand, Elastic deformation, Color change, 3DCG simulation*

### 1. Introduction

With the recent progress of 3-Dimensional Computer Graphics(3DCG) techniques, images using computer animation have been employed in various fields. In particular, because of its high information density, 3DCG has been utilized multi-directionally. The production of computer animations requires enormous time and labor, as it requires composition of a scene at each moment in time and its continuous display of that scene. Thus, various types of simulation techniques to automatically produce object movement have long been desired. 3DCG expression techniques for the human hand are needed in many fields[1], for example in medicine, as simulations for surgery, in education, as tools to learn techniques using the hands, such as sign language or playing musical instruments, and in the field of equipment design, as tools for the virtual evaluation of digital mockup-human compatibility. In entertainment, in particular, it is necessary to build up models with a high degree of freedom together with accurate dynamic simulations to reflect the intent of the makers.

In the hand-touching simulation proposed in this work, the extent of the deformed region induced when objects expressed by a spring model overlap can be visually controlled, and positional variation of the elasticity in the hand can be introduced by channel images. Based on the resulting simulation data, a color change effect can also be introduced by synthesizing three kinds of textures, leading to a hand-touching simulation technique which

enables intuitive control with a high degree of freedom.

### 2. Previous studies

As computer processing speeds continue to increase, the expression of the human form in virtual spaces is attracting interest from many fields. In the entertainment field, for instance, human hands are typically expressed with intricate detail in movies, games and commercial films[2]. However, because these representations do not express skin elasticity by means of touching (as the technique described in this paper attempts to do), they look rather like the hard hands of mannequins. One of the most widely used methods for expressing the elasticity of matter is the spring model[3]. The method proposed in this paper also uses a computation technique that applies the spring model. Recently, some elasticity models for use in real-time simulations of deformation have been proposed[4], [5], [6], [7]. Among the techniques that have been proposed to simulate hand-joint motions are the finite element method[8], the anatomical skeleton and virtual muscle model[9], the example-based deformation technique using medical images[10] and a model that simulates the action of grasping[11]. However, since these techniques adopt bottom-up type algorithms, they cannot be applied directly to actual graphic representations, thus limiting the freedom of the animator. Moreover, when the purpose is to produce high-quality images, it is not

always necessary that the elasticity model run in real time. Therefore, the method we are proposing puts the emphasis on expanding the range of expressions available to animator rather than on enabling real-time generation of animations.

Since animators usually employ the key frame method — which generates animations by moving objects visually on the screen, the method proposed here functions like a touch-up effect. It is added to the user-produced animation through a process similar to interpolation, rather than being specified with parameters.

A number of methods have also been proposed for changing the surface shapes. These include a technique that generates wrinkles along the vector field[12], one that generates wrinkles from a cubic Bezier curve using a geometric model[13], and one that deforming the skin model by means of muscles and expresses the shapes of wrinkles as functions according to the deformation of the skin model[14]. However, while these techniques express fine wrinkles or wrinkles on the muscular structure below a simple skin surface such as the face, they do not consider the mechanism of formation of wrinkles specific to the palm.

Several techniques have also been developed to express the colors and textures of the skin, including a technique that expresses texture by referring to the complicated interplay of multiplex reflections of light on the microscopic structure of the skin surface[15]. Other techniques include a technique for expressing colors based on the pigments contained in skin[16], a technique for expressing both colors and textures using a volume model[17] and an analysis/synthesis technique that extracts hemoglobin and melanin information based on actual measurement data[18]. However, there have been no attempts to simulate color changes caused by pressure on the skin such as touching, as we are proposing. On the other hand, there are reports on the change in pigment component distribution due to hemostasis and the results of measurement of reactions against load on the hand by measuring the variation in the oxygen saturation under anaerobic motions[19]. The method proposed in this paper makes it possible to express color change effects according to the intention of the user, and is also capable of simulation using measurement results.

### 3. Algorithm

#### 3.1. Simulation of deformation

Fig.1 shows the outline of the algorithm. First, animation data, where mesh objects (partially intersected), are input

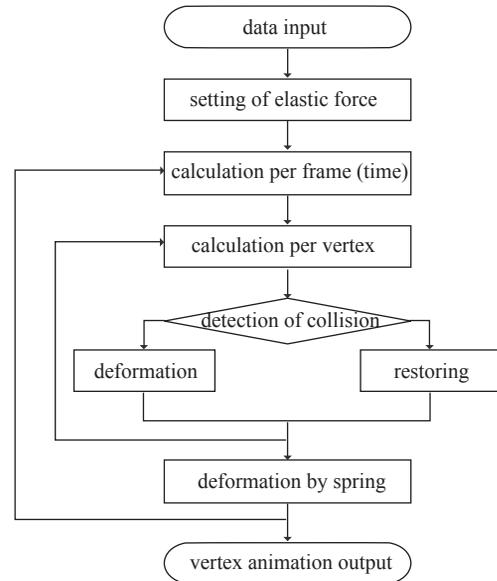


Fig.1: Outline of deformation simulation

(Fig.2). The extent of the intersecting region, indicating the magnitude of the collision of the objects, can be controlled by the user. Also, the extent of the deformation of the spring and the elastic force at each vertex(section 3.3) can be arbitrarily controlled. Next, if the objects are colliding with each other as described in Fig.3, the point of intersection of the inverse vector of the normal line vector at the vertex  $i$  with the other object is defined as collision point  $j$ . Using the movement vectors in the user-input animation,  $M_i$  for vertex  $i$  and  $M_j$  for the collision point  $j$ , length to the surface of target object  $L$ , a force vector applied on point  $i$  will be obtained by

$$f_i = L \left( \frac{M_j - M_i}{|M_j - M_i|} \right). \quad \dots (1)$$

By obtaining the point of intersection of the inverse vector of normal line vector at vertex  $i$  with the other object using the vector  $f_i$  obtained from eq.1, the position of the vertex  $i$  is determined. Also, as shown in Fig.4(a), when moving approximately perpendicular to the normal line of the object at the time of data input, the user can choose whether to move by pushing the collision peak as in Fig.4(b), or by restoring the elastic body as in Fig.4(c). In the case of Fig.4(c), the force vector,  $f_i$  can be defined by

$$f_i = L \left( \frac{M_j - M_i}{|M_j - M_i|} - \frac{N_i}{|N_i|} \right) / \left| \frac{M_j - M_i}{|M_j - M_i|} - \frac{N_i}{|N_i|} \right|, \quad \dots (2)$$

where  $N_i$  represents a normal vector at the vertex  $i$ . If both objects are elastic, the displacement is derived from the comparison between the elastic forces at the vertex  $i$  and

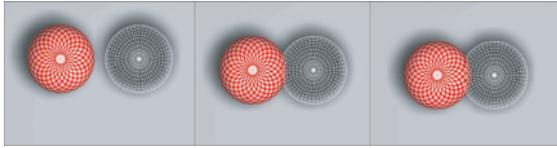


Fig.2: A simple example of two bodies making contact without collision detection and deformation

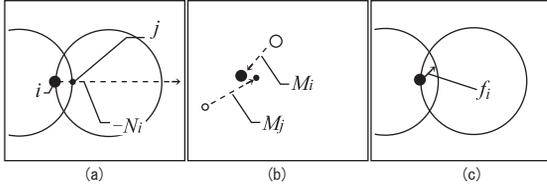


Fig.3: The two big circles represent the objects that are colliding, and calculations are performed on vertex  $i$ . Assuming that the point of intersection between the negative vector from vertex  $i$  to normal line  $N_i$  and the object is point  $j$  (a), the movement vector  $M_i, M_j$  caused by the user-input animation between vertex  $i$  and intersecting point  $j$  is obtained (b). Then, deformation vector  $f_i$  due to collision at vertex  $i$  is determined based on formula (1) (c).

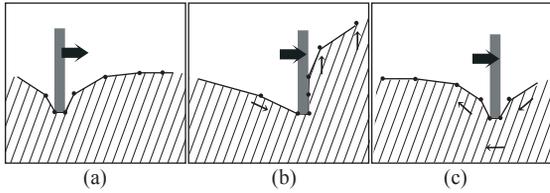


Fig.4: The hashed area represents the elastic object, the polygonal lines represent the object edges, the dot represents the vertex, the rectangle represents the collision target object, the thick arrow represents the movement of the target object and the thin arrow represents the movement of the vertex. When movement is in the direction perpendicular to the normal line of the elastic object (a), the user selects whether the movement pushes the colliding region onto the elastic object (b) or moves along it as if tracing its surface (c). When (c) is selected, formula (2) is used in the deformation calculation.

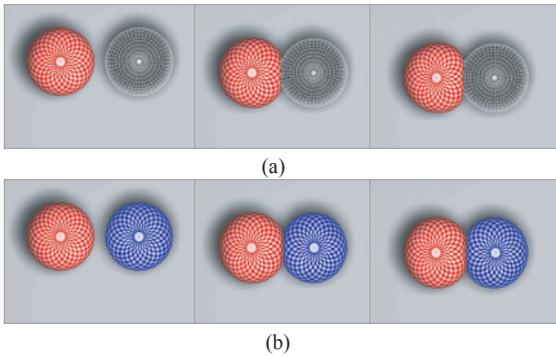


Fig.5: Result of deformation simulation using the animation shown in Fig.3(a) shows the deformation of an elastic object (the red sphere) when it collides with a rigid body (the wire frame) (b) shows the resulting deformation when both objects colliding are elastic.

the collision point  $j$ , leading to the determination of the position. When there is no collision and the objects are deformed, force is applied to restore them to their original shape. The amount of restorative force applied can be set arbitrarily by the user during data input. After finishing

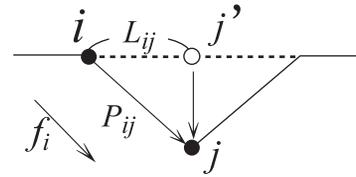


Fig.6: Spring model calculation technique used by the proposed method. Which was initially in position  $j'$ , comes to position  $j$ , the force imposed on vertex  $i$  can be calculated as  $f_i$  using formula (3).

the calculations for all the vertexes, forces generated by the springs are added to the vertexes around the deformed vertex (section 3.2). By outputting these calculations as a vertex animation, an animation of the deformation of elastic matter can be produced (Fig.5).

### 3.2. Spring Model

Spring models are widely used for motional simulations of elastic matter [3]. The models consist of mass vertexes and springs linking the vertexes. As shown in Fig.6, when the vertex that has initially been in position  $j'$  moves to position  $j$  as a result of collision, a force  $f_i$ , acting on vertex  $i$  is calculated from the following equation (3), using the relative position vector  $P_{ij}$  between the vertex  $i$  and the collision vertex  $j$ , the natural length between the two vertexes  $L_{ij}$  (length with input animation), the user-set elastic coefficient  $K_i$  and the attenuation coefficient  $D_i$  at vertex  $i$ .

$$f_i = \sum_j \{ (|P_{ij}| - L_{ij}) \times \frac{P_{ij}}{|P_{ij}|} \times K_i \times D_i \} \quad \dots (3)$$

In the algorithm proposed in this paper, the user can arbitrarily set the area influenced by the spring force at the time of data input. For example, if a load is applied to the fingertip, the simulation can be bound to the joint, and if a load is applied to a region of the palm, the simulation can be bound to the whole palm. Such a setting can greatly save computation time, and can also suppress deformations not expected by the user. However, limiting the region affected by the spring force leads to some inconvenience since application of the force will be discontinuous at the region's boundary. Thus, to avoid a gap between the two regions, a weight function of the distance from the vertex  $j$  to each vertexes, which is 1.0 at the vertex  $j$  and 0.0 at the longest distance within the area influenced by the spring force, is introduced. This is defined as the attenuation coefficient,  $D$ .

### 3.3. Elastic Displacement

Channel images are used to interactively control the variation of the elastic force depending on the position in the hand. A 2D channel image is prepared for a hand model developed as shown in Fig.7 or a parallel-projected

mesh object (Fig.7(a)). In the simulation of touching, the brightness (0.0-1.0) at the corresponding points from each vertex is determined. This is defined as the elastic coefficient  $K$ . Currently, the most common technique for expressing the hand in image production is animator to draw a texture map on a 2D image that is usually expanded by means of parallel projection or cylindrical projection. This fact suggests that drawing the elastic channel is also a relatively easy and intuitive task for animators. Using this, it is possible to intuitively control partial hardening, e.g., for the nail, and the formation of wrinkles when external forces are applied[23]. In Fig.8, the nail area is regarded as an image with a brightness of 0, indicating that the nail is a non-deformed element. Fig.9 show that the wrinkles induced by external forces (Fig.9(d)) can be expressed by drawing lines on the hand

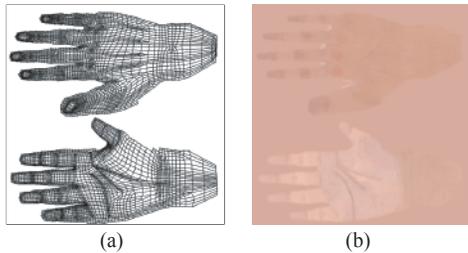


Fig.7: (a) Parallel projection of the mesh model of hand. (b) Color map image of the hand model used in the figures.

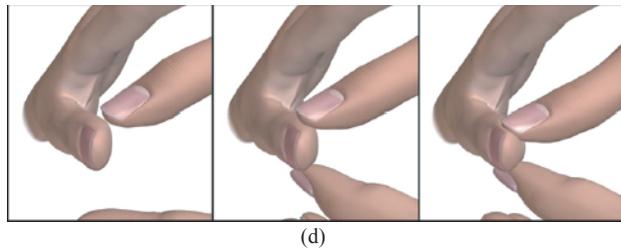
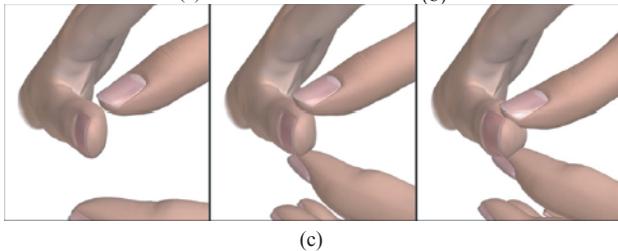
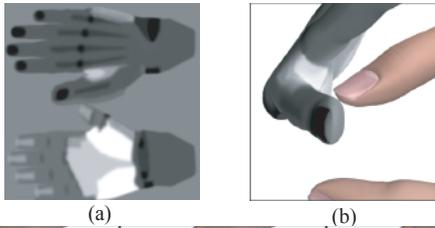


Fig.8: Examples of how the elastic channel is used. (a) shows the elastic channel image, (b) shows an example of mapping on an object, (c) shows the animation without the elastic displacement, and (d) shows an animation to which elastic displacement has been applied.

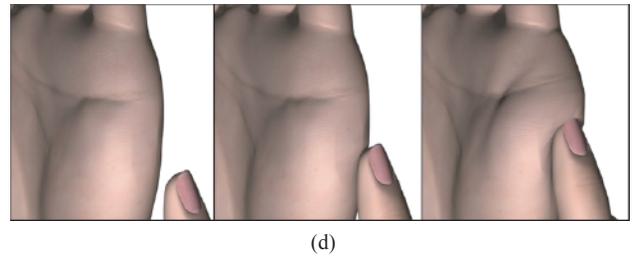
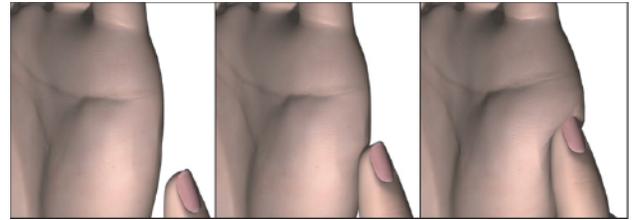
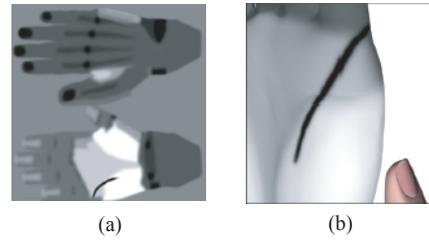


Fig.9: Examples of lines drawn on the channel image. Similarly to Fig.8, (a) shows the channel image, (b) shows an example of mapping, (c) shows the animation without elastic displacement and (d) shows the animation with elastic displacement. These figures show the possibility of intuitive control of wrinkles formed by a force.

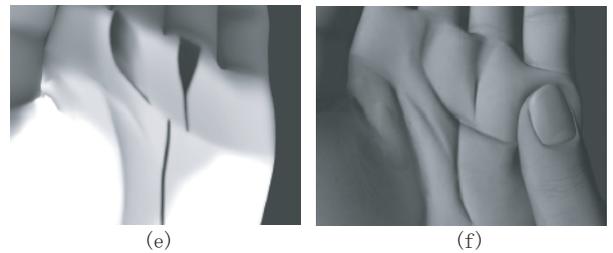
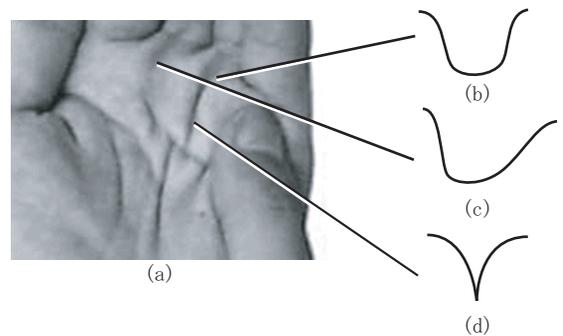


Fig.10: Examples of comparison with actual image. (b), (c) and (d) show the cross-sectional shapes estimated at the respective positions, (e) shows the map image for representing the status of (a), and (f) shows the result of simulation using (e).

in the channel images (Fig.9(a)). The technique of forming wrinkles by drawing lines may seem to resemble the technique proposed by Bando et al.[13], but this technique expresses the wrinkle shapes exclusively using the shape shown in Fig.10(d). Nevertheless, since wrinkles on the hand, particularly the palm, have a large variety of form variations as shown in Figs.10(b) and (c), some flexibility is required for their shapes as well as their positions and depths. This will make it possible to achieve more finely detailed expression by specifying the wrinkles by means of images rather than using line data only.

### 3.4. Color Change Simulations

The visible phenomenon caused by the touch of human hand includes color change as well as deformation. Skin

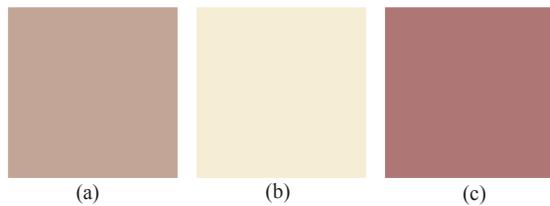


Fig.11: Three kinds of synthesized textures used in Figs.12 and 13. (a) shows the basic texture, (b) the texture of the colliding region and (c) the texture around the colliding region.

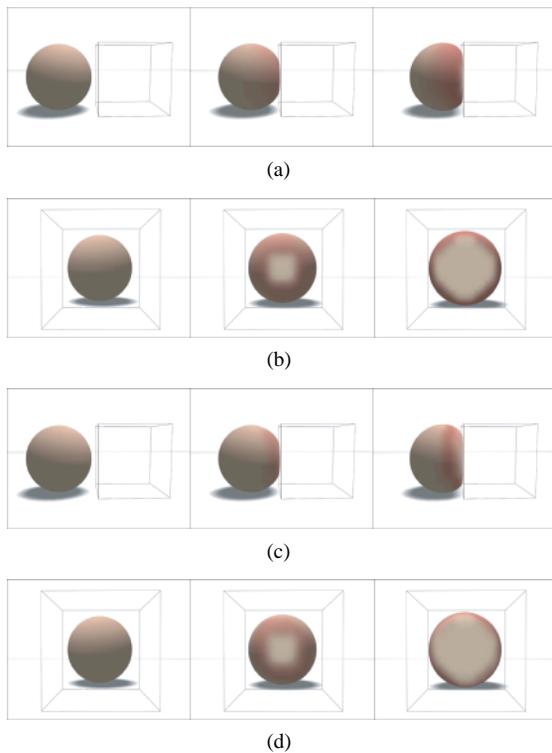


Fig.12 Result of color change simulation using the texture shown in Fig.11: (a) and (b) are the same images rendered from different angles, and similarly for (c) and (d). In (c) and (d), the texture areas of the colliding region are set more widely than those in (a) and (b), and the texture areas of the peripheral regions are set more narrowly.

color is determined mainly by the concentrations of melanin and carotene contained in the skin, the concentration of hemoglobin in the capillaries, and the oxygen saturation (that is, the percentage of oxidized hemoglobin in the total hemoglobin[16]). With regard to the effects of hemostasis on the skin, there is no reported change in melanin content, but hemoglobin concentration and oxygen saturation vary noticeably[19]. This suggests that skin color changes caused by a load such as touching can be classified as a change in the hemoglobin concentration due to pressure on the skin. A yellowish color change would be due to a drop in the oxygen saturation caused by a change in the hemoglobin concentration due to pressure on the internal capillaries, while a reddish color change would be due to a rise in oxygen saturation. Consequently, this paper proposes a technique for expressing color changes due to skin touching by synthesizing three kinds of textures as described below.

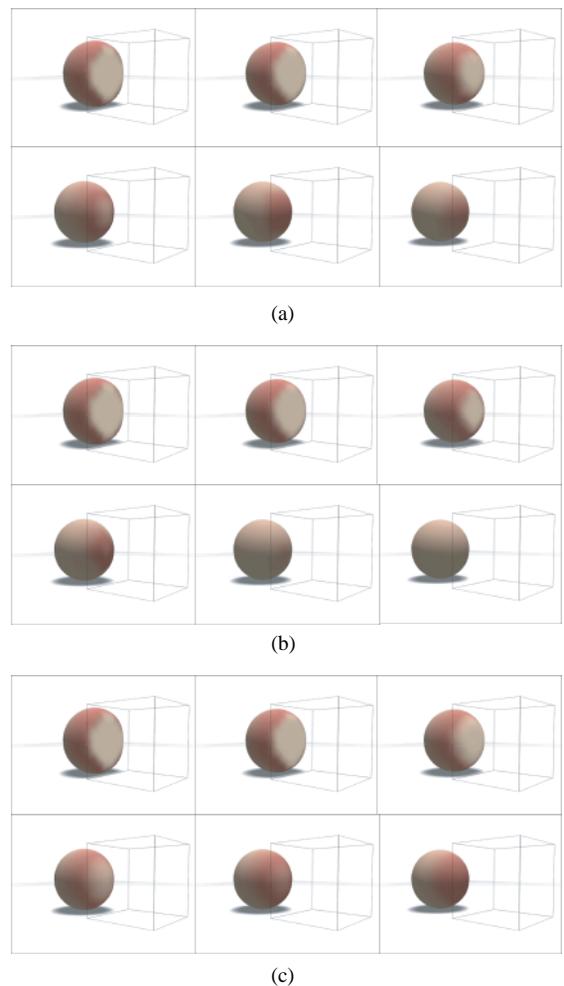


Fig.13: Animation showing the recovery from the status of Fig.12 to the basic texture. (b) is set to take longer to return to the basic texture than (a), and (c) is set to take less time.

The outline of the algorithm is as follows. At first, three kinds of textures are prepared (Fig.11), the basic texture, the texture of the colliding region, and the texture around the colliding region. The user can arbitrarily set the area where each texture appears. Then, the collision peak is obtained from the simulation data, and the peak color is determined using the values set at the collision vertex and the peripheral vertexes. Lastly, using these data, channel images synthesized from the three kinds of textures are prepared to produce an animation with the color change as illustrated in Fig.12[23]. When the colliding object is separated from the target, the colliding region maintains the collision texture for a certain period, but then changes to the surrounding texture and eventually returns to the original basic texture (Fig.13). This represents the phenomenon in which hemoglobin concentration and oxygen saturation rise significantly as a result of the sudden blocking of blood flow by pressurization. The duration of each texture can be set as desired by the animator.

#### 4. Simulation Results

Fig.14 shows results of simulation using the algorithm described above[23]. Figs.14(a), (b), and (c) show an original animation, its deformed version, and version with a color change, respectively. These images show that the proposed method can effectively express the reaction to touching for joint motion with a high degree of flexibility. The number of vertexes is 2495 for these hands. Each step in the simulation is assumed to be 0.02[s] and computation takes about 20.0[s] using a machine with Xeon 3.2GHz with 3.0GB of memory. Since the primary goal of this method is not to create a real-time model, but to improve the user's freedom and control and to ensure maximum image quality, it does not take full advantage of increased computation speed. As a result, there is still room for further improvement in terms of proper utilization of computation speed. Although Fig.14(d) shows an overstated hand, the expression of touching via the complex interplay of associated elements should be effective not only for animation of real human characters, but also for overstated characters. This algorithm can be applied not only to the human hand but also to any other elastic matter. Moreover, the expression technique introduced here, which expresses the deformations and color changes due to touching, can also be applied in fields other than entertainment. For example, Fig.15 shows images as tools to learn techniques for playing guitar. A user will be able to better understand by seeing both the front view (Fig.15(a)) and the back view (Fig.15(b)). Fig.16 shows a

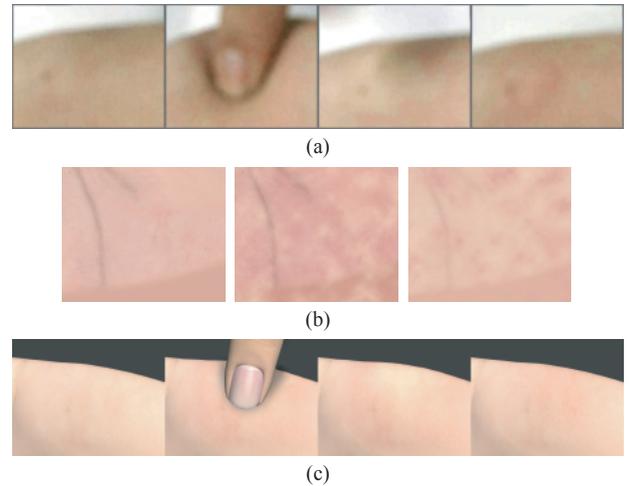


Fig.16: Comparison of color change animation with an actual image sequence. (a) shows the actual sequence, (b) shows the three kinds of textures use in reproduction, and (c) shows the color change animation using textures (b).

comparison of the color change simulation sequence with an actual image. The actual color change of the hand is not as uniform as shown in Fig.12, but the proposed method employs the texture synthesis technique so it is capable of producing complex color change animations as desired by the animator.

#### 5. Future tasks

In this paper[20], an effective technique for the representation of hand touching has been proposed, and results of the simulation have been reported. These results show that the algorithm proposed enables operation with a high degree of freedom for the user, and is effective for visual simulations of the hand. However, improvement of the following points is necessary.

1. When the objects are excessively intersected, they dent deeply. Thus, there exists a limit to the magnitude of the collision in order to avoid their penetration.
2. When multiple parts collide simultaneously, it requires a great deal of labor to individually set parameters such as color change.

The skeletal subspace deformation technique[21], used for various kinds of software at present, is not accurate anatomically, and thus not suitable for the deformation of the hand[10]. On the other hand, when hand deformation is looked from the viewpoint of collision, the skin collides to expand inside the deformation, and the collision between the skin and bone can emboss the shape of the bone outside the deformation. By simulating them and applying a color change effect due to pressing and extrusion, an algorithm enabling intuitive control and precise expression can be expected. A study regarding restriction of the joint movement

to actions occurring simultaneously has been reported[22], and we have examined the effect of the extreme intersection mentioned in the preceding chapter to the joints. Thus, considering these previous studies, we are planning to develop a more advanced algorithm.

## References

[1]Foundation Machine System Promotion Society, "Investigation Research Report Concerning Getalhand Technology", System technology development investigation research, 15-R-2, <http://www.hql.or.jp/gpd/jpn/www/etc/dhyoushi.pdf>, HQLnews, No.1059, 2004, in Japanese.

[2]The ANIMATRIX <http://www.intothematrix.com/>, 2003.

[3]D. Terzopoulos, J. Platt, A. Barr and K. Fleisher, "Elastically Deformable Models", *Computer Graphics*, 21 (4), pp.205-214, 1987.

[4]M. Muller, J. Dorsey, L. McMillan, R. Jagnow and B. Cutler, "Stable Real-Time Deformation", *Proceedings of SIGGRAPH 99*, pp.65-72, 1999.

[5]M. Muller, L. McMillan, J. Dorsey and R. Jagnow, "Real-Time Simulation of Deformation and Fracture of Stiff Materials", *Eurographics 2001 Computer Animation and Simulation Workshop*, pp.27-34, 2001.

[6]M.Kato, Y.Sakamoto, "Generation and Transformation Operator of Mass-Spring Model from Volume Data", *IEICE technical report, EID98-167, ID98-158*, pp.19-24, 1999, in Japanese.

[7]T.Ikegami, N.Saiwaki, S.Nishita, "Automatic Generation of Movement of Object that uses 3D Spring Model", *Technical report of IEICE, MVE96-70*, pp.15-20, 1997, in Japanese.

[8]J.Gourret, N.M.Thalmann, D.Thalmann, "Simulation of Object and Human Skin Deformations in a Grasping Task", *Proceedings of SIGGRAPH '89*, 1989.

[9]I.Albrecht, J.Haber, and H.Seidel, "Construction and Animation of Anatomically Based Human Hand Models", *Proceedings of Eurographics /SIGGRAPH Symp. on Computer Animation 2003*, pp.98- 109,368, 2003.

[10]T.Kuriahara , N.Miyata, "Modeling Deformable Human Hands from Medical Images", *ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2004*, pp.357-364, August 27-29, Grenoble, France, 2004.

[11]N.S.Pollard and V.B.Zordan, "Physically Based Grasping Control from Example", *ACM SIGGRAPH / Eurographics Symposium on Computer Animation, Los Angeles, CA, 2005*.

[12]Y.Bando, T.Nishita, "Human Skin Simulation by Generating Wrinkles along Vector Field", *Visual Computing*

*Graphics & CAD joint symposium 2001*, pp.7-12, 2001, in Japanese.

[13]Y.Bando, T.Kuratate and T.Nishita, "A Simple Method for Modeling Wrinkles on Human Skin", *Pacific Conference on Computer Graphics and Applications*, pp.166-175, 2002.

[14]Y.Wu, P.Karla and N.M.Thalmann "Physically-based Wrinkle Simulation & Skin Rendering." *Eurographics Workshop on Computer Animation and Simulation '97*, pp.66-79,1997.

[15]T.Ishii, T.Yasuda, S.Yokoi, J.Toriwaki, "Human Skin Rendering Based on the surface Microstructures", *Proceedings of Nicograph 1991*,pp.68-76, 1991, in Japanese.

[16]H.Nakai, Y.Manabe, S.Inokuchi, "Simulation and Evaluation of Human Skin-Like Quality", *IEICE TRANSACTIONS D-II, J84-D-II No.2*, pp.321-327, 2001, in Japanese.

[17]T.Kawai, Y.Kurioka, "Volumetric Visualization of Human Skin", *The transactions of the Institute of Electrical Engineers of Japan, C,123,11*, pp.1936-1943, 2003, in Japanese

[18]N.Tsumura, et. al., "Image-based skin color and texture analysis / synthesis by extracting hemoglobin and melanin information in the skin", *Proceedings of SIGGRAPH 2003*, pp.770-779, 2003.

[19]D.Nakao, N.Tsumura, Y.Miyake, "Real-time multi-spectral image processing for mapping pigmentation in human skin", *Medical Imaging Technology Vol.20 No.2*, pp.120-133, 2002, in Japanese

[20]Y.Ishiguro, T.Kikuchi, "Visual Simulation of Hand Touching Introducing Elastic Deformation and Color Change", *Proceedings of Nicograph International 2005*, pp.55-60, 2005

[21]N.M.Thalmann, R.Lperriere and D.Thalmann, "Joint-dependent local deformations for hand animation and object grasping", *Graphics Interface '88*,pp.26-23, 1988

[22]G.ElKoura, K.Singh, "Handrix: Animating the Human Hand", *Proceedings of Eurographics/SIGGRAPH Symp. on Computer Animation 2003*, 2003

## Appendix

[23]Sample movies  
<http://www.id.takushoku-u.ac.jp/~tkikuchi/nicointer2005/>

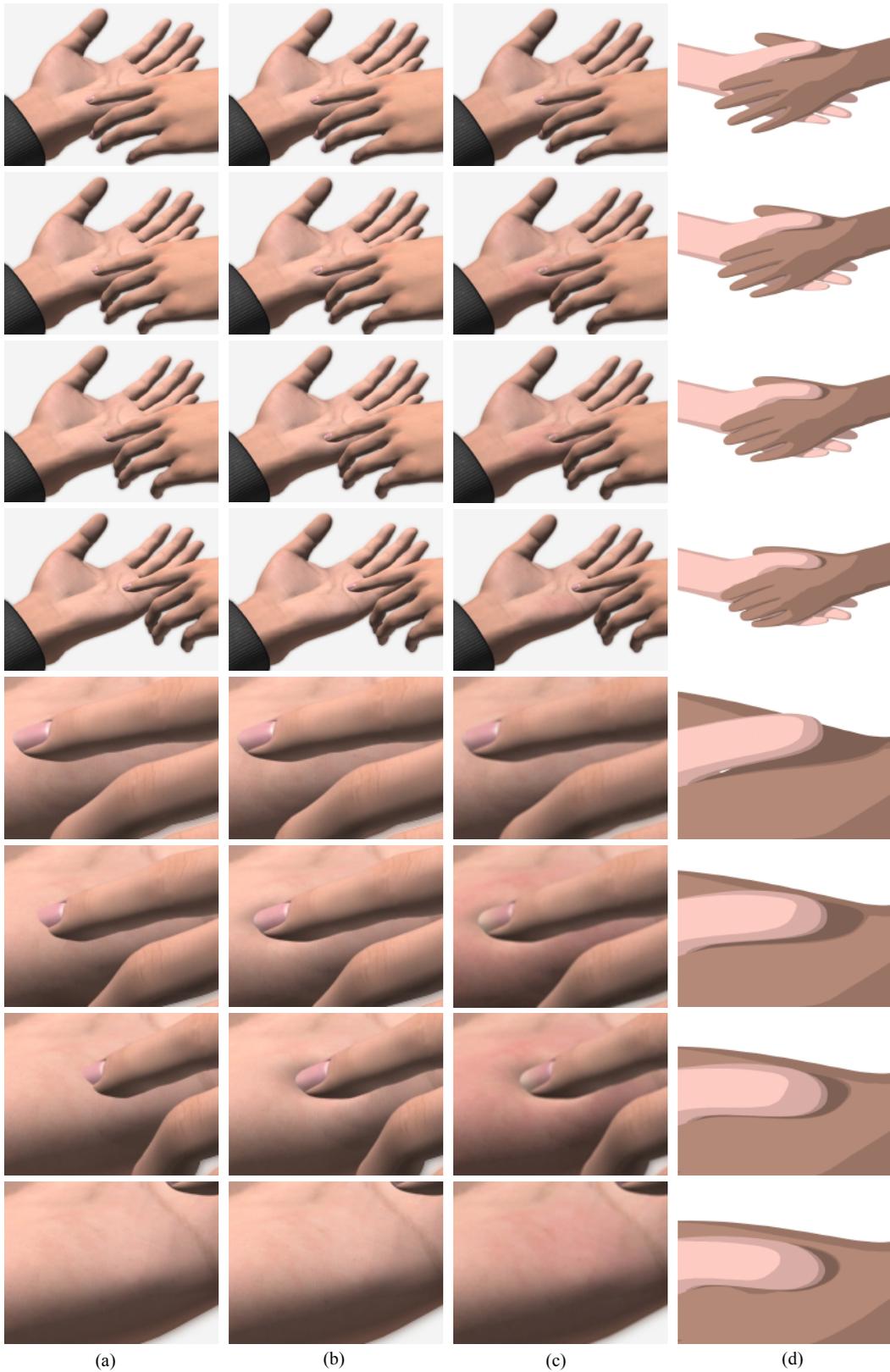


Fig.14: Examples of application of the algorithm to the hand model. (a) shows the user-input animation before simulation, (b) shows the result of deformation simulation, (c) shows the result of color change simulation applied to (b), and (d) shows an example of application to the deformed hand model.

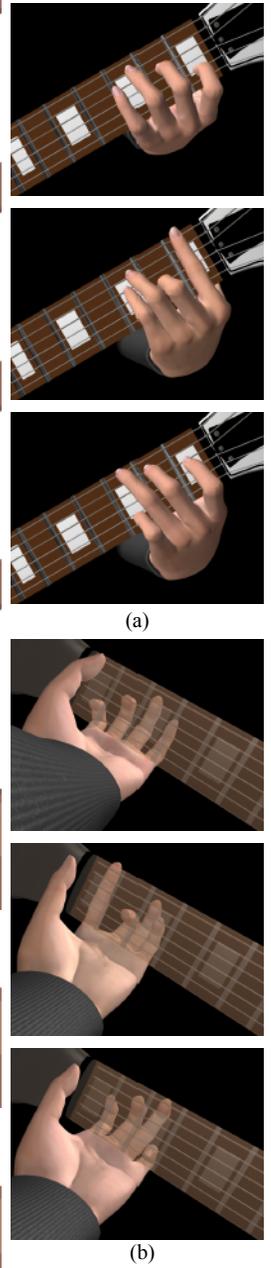


Fig.15: Animation of a hand playing a guitar. (a) shows the frontal view and (b) shows the rear view by turning the neck semi-transparent.