Generic-Model Based Human-Body Modeling

Xiaomao Wu[+], Lizhuang Ma[+], Ke-Sen Huang*, Yan Gao[+], and Zhihua Chen[+]

[+]Department of Computer Science & Engineering, Shanghai Jiao Tong University
HuaShan Rd. No. 1954, Shanghai, P.R.C (200030)
{xmwu,ma-lz,gaoyan,zhchen}@cs.sjtu.edu.cn

*Department of Computer Science, National Tsing Hua University
Kuang Fu Rd. No. 101, Sec. 2, HsingChu, Taiwan 300 R.O.C (30013)
hks@glserver.cs.nthu.edu.tw

Abstract. This paper presents a generic-model based human-body modeling method which take the anatomical structure of the human body into account. The generic model contains anatomical structure of bones and muscles of the human body. For a given target skin mesh, the generic model can be scaled according to the skin, and then morphed to be fitted to the shape of the target skin mesh. After an anchoring process, the layered model can be animated via key-framing or motion capture data. The advantage of this approach is its convenience and efficiency comparing to existing anatomically-based modeling methods. Experimental results demonstrate the success of the proposed human-bodymodeling method.

Keywords: anatomically-based modeling, human body modeling, generic model

1 Introduction

Modeling and deformations of human bodies remain one of the main challenges in the field of computer graphics. The human body is biologically complex, which consists of more than 200 bones and 400 muscles [1]. Additionally, we are subconsciously sensitive to the details of human bodies because we are extremely familiar to our bodies.

Fortunately, users concern only about the skin deformations of the human body in most graphics applications. Human skin can be modeled as parametric surfaces by using existing commercial software, or be acquired via 3D laser scanners. Given a skin mesh in a static pose, animators could either employ the widely used method – skinning, or more complex anatomically-based method, to animate the skin.

Skinning is the most common method for generating of skin deformations in games and interactive systems, it has many names including SSD, enveloping, smooth skinning, transform blending, matrix blending and linear blend skinning. While this method is very fast and widely supported by commercial applications, it cannot represent complex deformations and suffers from "collapsing joint" and "candy-wrapper" defects [2, 3]. At the same time, adjusting weights of the skeleton to the skin vertices is a tedious and time-consuming task.

Alternatively, anatomically based approach can also be used to animate the skin. We can insert bones, muscles, and sometimes fat tissues below the given skin mesh, and utilize physically based method to simulate skin deformations [4–7]. This anatomically-based method mimics the layered deformation way of real humans, thus can produce detailed and realistic skin deformations. Anatomically based method is not so popular as skinning, because it requires the user designing the bones, muscles, and sometimes fat tissues individually, and fine-tuning the muscle shapes carefully, in order to fit those underlying structures to the skin mesh.

Example-based method is another competitive way for generate skin deformations. If we have obtained a series of scanned body poses via scanning, we can use the algorithm described in [8] to interpolate those poses, to produce very realistic skin deformations. While this approach can produce detailed and realistic skin deformation effects, it faces the retargeting problem: we can only capture the skin deformations of a specific human body at a time, and cannot easily retarget the captured deformations to the skin of a different human body. Furthermore, this approach is not suitable for the situation that a body shape is only in the animator's mind and doesn't exist at all, because we have no way to scan an imaginary body.

Inspired by related work on facial animation [4–6] and horse modeling [7, 9], we present an improved anatomically-based modeling method. We uses a generic model to simplify the construction process of human anatomical structures. Our method inherits of the advantages of anatomically-based modeling approach and is easier to implement.

Our method involves the following five steps: (1) Design the generic model; (2) Obtain the target skin mesh. The target skin mesh could either be modeled with commercial modeling software, or directly obtained by scanning a real person using 3D scanners; (3) Design the skeletons of the generic model and that of the target skin by using the interactive tool we have developed for skeleton design; (4) Fit the generic model to the skin mesh; (5) Anchor skin vertices to underlying components, then animate the skin mesh.

The next section introduces related work on human body modeling. Section 3 introduces how to prepare the generic model and the target skin mesh. Section 4 introduces the method we use to fit the generic model to the target mesh. In section 5, we introduce how the final model can be animated. Experimental results are shown in section 6. We conclude in section 7 with the future work.

2 Related Work

There are many ways to classify related research work that are most related to ours. We describe them in three classes: generating skin upon underlying components; deforming pre-existing skin; and deforming skin by examples.

The first class attempts to extract the skin from underlying components such as bones and muscles, it generally use implicit surfaces to represent the skin. Implicit surfaces are frequently utilized to model organic forms. Many researchers have used implicit surfaces to represent the skin. Blinn [10] used implicit surfaces generated by point skeletons with an exponentially decreasing field function to model his "blobby man". Bloomenthal [11] modeled a hand containing veins with convolution surfaces. He used polygons and lines as primitives. Yoshomito [12] used ellipsoidal metaballs to model the skin of a ballerina which produced realistic-looking shape. More recently, Scheepers *et al.* [13] and Wilhelms *et al.* [14] also used implicit functions to extract the skin from underlying structures.

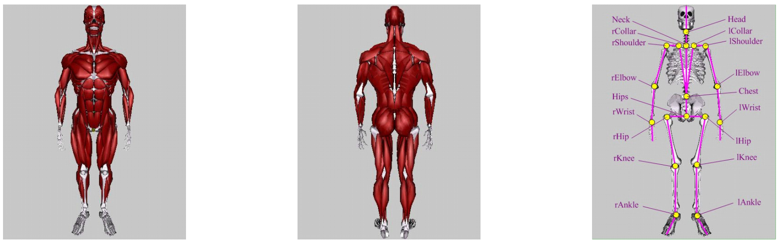
Deforming pre-existing skin is the second class of human body modeling. If we have modeled a skin with modeling system, e.g. Maya, or have acquired a skin mesh from a real person using 3D laser scanners, we may face the problem of how to deform that skin. Many researches have been done to solve this problem. Komatsu [15] applied a continuous deformation function with respect to the joint angles to deform the control points of a skin mesh represented by Bezier patches and Gregory patches. Magnenat-Thalmann *et al.*[16, 17] introduced the concept of *joint-dependent local deformation* or JLD, to deform existing skin algorithmically. Thalmann and Shen [18] formulated skin as cylindrical contours, they obtained smooth deformations of human trunk and limbs by setting the orientation and position of each contour. Schneider and Wilhelms [9] described a hybrid modeling method based on anatomy. They specified hierarchical skeleton, individual bones, muscles and generalized tissues below an existing skin, and used a physically based method to deform the skin. Simmons *et al.* [7] proposed generic-model based method to deform the pre-existing skin of a horse . They first deformed the skeleton and bones, and then fine-tuned the muscle shape, in order to fit them to the skin. Skinning [19, 2, 20, 3] is the most popular method to deform existing skin mesh, it defines the deformed vertex of the skin mesh as a weighted summation of the transformations of subspace.

The development of range scanning technologies has make it possible to scan a whole body in several minutes with satisfactory accuracy. Given a variety of scanned poses of a human body, Allen [8] successfully blended those poses, which produced smooth and highly-realistic skin deformations. Sand *et al.* [21] described a method for acquiring deformable human geometry from silhouettes. Body shape can be reconstructed and new body shape can be synthesized using a normalized radial basis function (NRBF).

Our method falls into the second class. We focus on how to deform a hand-made or range-scanned skin mesh efficiently and realistically, by using a generic model which contains bones and muscles of a human body. A detailed discussion on the unique of our method can be found in section 7.

3 Preparation of the Generic Model and Target Skin Mesh

The generic model plays an important role in our scheme. It consists of basic bones and muscles of the human body. The generic model will be used to semi-automatically reshaped according to a given skin mesh. The basic rules for establishing such a generic model are: Firstly, it should be general enough to represent the basic structure of bones and muscles of a human body; Secondly, anatomical accuracy of the generic model is not crucial, because our interest finally lies in the skin. More accurate anatomical structure will lead to slower speed in the final animation process; Thirdly, the generic model should stand at a muscle-relaxed pose, in which all muscles are placed on the bones in a state of relaxation. It is very helpful for simplifing the muscle animation process.



(a) (b) (c)

Fig. 1: The left two figures illustrate the generic model that we designed. (a) Frontal view. (b) Rear view. The right figure shows the designed skeleton of the generic model. The skeleton is designed interactively by using the graphics tool we developed. Muscles are not shown for clarity.

3.1 Generic-Model Design

Many ways can be used for designing the generic model (Fig. 1). It can be modeled with commercial modeling systems like Maya, SoftImage or 3DMax by artists. Basic knowledge of human anatomy [1] is required to model the generic model properly. Alternatively, we can also make use of existing volume database of human bodies [22], and use reconstruction algorithms to construct the generic model from scanned volume data [23].

We have modeled the generic model with 3DMax referring to anatomy books [1]. Each bone or muscle are individually modeled. The final generic model consists of 176 individual bones and 186 individual muscles. Bones and muscles are modeled as triangular meshes. Currently, we only model skeletal muscles. Cardiac muscles and smooth muscles are not modeled because we do not take facial and organic animation into account.

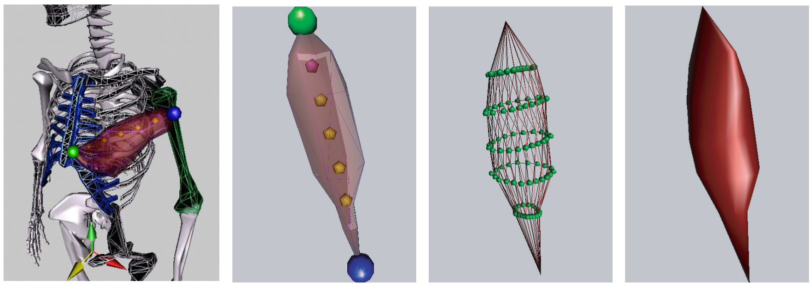
3.2 Skeleton Design

Skeleton can be used to name the bones of a human body that are attached to each other by joints, or the stick figure representing the positions and orientations of the joints that build up the articulated figure [24]. We follow the second meaning in this paper. A skeleton should be embedded into the generic model(Fig. 1), and will be used later to transform the generic model in order to fit the generic model to the skin.

The task of skeleton design is to specify the skeleton parameters $SC_i = \{P_i, O_i\}$, $i = 0, 1, ..., n-1$, where $P_i \in R^3$ is the joint position, $O_i \in S^3$ is the joint orientation, and n is the joint number. We have developed a tool which can be used to interactively specify the skeleton parameters.

3.3 Muscle Parameter Design

Muscles of the generic model are modeled as irregular triangular meshes, several parameters should be assigned to each muscle, because those parameters are important



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

Fig. 2: (a) Muscle parameter design. The muscle is shown in red. The small green and blue sphere represent the *origin* and the *insertion* respectively. The five small yellow spheres represent the points on the *action line*. The bones shown in blue and green are the *origin bone* and the *insertion bone* respectively. (b)-(d): The muscle resampling process. (b) The original shape of bicep muscle with seven points on the *action line*. (c) Intersections between rays emitted from the middle five point on the *action line* and the original muscle mesh. (d) The reconstructed muscle shape with Phong Shading.

in the muscle morphing and animating stages. Before we introduce the muscle parameters, we first introduce four terms referring to the characteristics of muscle deformations (Fig. 2(a)): *origin*, *insertion*, *belly* and *action line* [1].

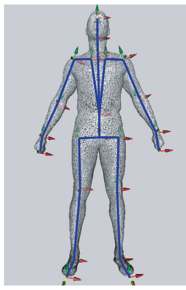
A skeletal muscle may attach a bone to another bone (often across a joint) or a bone to another structure, such as skin. When the muscle contracts, one of the structures usually remains stationary, while the other moves. The *origin* of the muscle is the muscle end that attaches to the stationary structure, usually bone. The *insertion* of the muscle is the muscle end that attaches to the moving structure.The *belly* of the muscle is that part of the muscle between the origin and insertion. The *action line* of the muscle is a spline starting from *origin* to *insertion*, it will be used to simulate muscle deformation.

The parameters of a muscle can be expressed as $M_i = \{O, I, Q_j, B_o, B_i, P_o, P_i\}$, where M_i is the parameters related to the ith muscle, O is the *origin*, I is the *insertion*, Q_js ($j \in [1, 5]$) are the five points on the *action line*, B_o is the *origin bone* which is the bone that the $origin$ attached to, B_i is the *insertion bone* which is the bone that the $insertion$ attached to, P_o and P_i are the *origin* and *insertion* of the muscle respectively. Those parameters are interactively designed by using the designing tool we have developed. Designing the parameter M_i for each muscle is a time-consuming task. But fortunately, it need be done only once.

3.4 Target Skin Mesh

The target skin mesh is the target mesh that we want to animate. It could be modeled with current modeling systems, or be obtained from digital skin model library produced by scanning real human bodies using 3D body scanners like CyberWare WB4 [25]. The previous method is more flexible, but is time-consuming, which requires the modeler has experience on modeling and art design; the latter method can produce more realistic

(a)



(b)

Fig. 3: (a) The target skin mesh. (b) The skin skeleton has been designed.

human skin, but is not so flexible as the previous one. Our generic-based modeling technique is suitable for both methods. In our work, we have modeled the target skin with the latter method as an example. The target skin mesh is downloaded from Cyberware [26].

4 From Generic Model to Specific Target Skin

In this section, we introduce the method that we use to fit the generic model to an existing skin mesh. Our method consists of 4 steps: Firstly, design the skin skeleton; Secondly, scale the generic model according to the skeletons of the generic model and the skin; Thirdly, resample the irregular triangular mesh of each muscle to regular shape that is suitable for morphing and animating; Finally, morph the scaled muscle shapes so that they will be properly fitted to the skin.

4.1 Skin Skeleton Design

A skeleton should also be embedded into the target skin, which is very important to the following process for generic scaling (section 4.2) and the animating stage (section 5). The design process for skin skeleton is similar to the one described in section 3.2. We interactively specify the parameters of the skin skeleton $SS_i = \{P_i, O_i\}$, $i = 0, 1, \cdots, n-1$, where $P_i \in R^3$ is the joint position, $O_i \in S^3$ is the joint orientation, and n is the joint number. During the skeleton designing process, the skin is rendered translucently to assist the user specifying the location and orientation of each joint clearly. Fig. 3(b) show the designed skeleton of the skin mesh.

4.2 Generic Model Scaling

In order to fit the generic model to the skin, we first scale the bones and muscles of the generic model. The scaling algorithm consists of the following five steps:

1. Group bones of the generic model.
2. Transform bones to local joint coordinates.

3. Scale the bones of each segment.
4. Change the skeleton of the generic model.
5. Scale muscles.

We start the scaling process by grouping bones of the generic model. The bones that lie between two joints are assigned to the parent joint , e.g., bones between $hips$ and $chest$ are assigned to hip joint, and bones between $chest$ and $neck$ are assigned to $chest$ joint.

In step 2, the grouped bones are transformed to the local coordinate system of their associated joints. If the accumulated matrix of the coordinate system of a local joint is M_i, then the transformed bone vertices are $M_i^{-1} \cdot v_j, j = 1, 2, \cdots, n$, where v_js are the bone vertex.

In step 3, the transformed bones of each segment are scaled according to the segment lengths of the generic-model skeleton and the skin skeleton. Segment length is defined as the Euclidean distance between the two joints that a segment attached to. Each bone is scaled by $S_{ij} = LS_i/LC_i$ along the segment direction pointing from the parent joint to the children joint, where S_{ij} is the scale factor of the jth bone in ith segment, LS_i and LC_i are the lengths of the ith segment of the generic-model skeleton and of the skin skeleton, respectively. The scale factor along the other two orthogonal axises of the local coordinate system are also scaled by S_{ij}, but they can be interactively modified by the user. Fig. 5(b) and Fig. 5(c) show the bones of the generic model before and after the transformation and scaling process.

In step 4, the generic-model skeleton should be changed to have the same parameters as the skin skeleton, i.e., $SC_i = SS_i,\ i = 0, 2, \cdots, n-1$, where SC_i and SS_i represent the skeleton parameters of the generic model and of the skin individually.

Finally, all muscles should be scaled after the scaling process of bones. Muscle shapes will be resampled and morphed. The according transformation matrix can be represented by equation 1.

$$M_i = M_R \cdot M_T \cdot M_0^{-1} \cdot M_s \cdot M_0 \cdot v \qquad (1)$$

where:

M_i is the according transform matrix of the ith muscle.

M_0 is the transform matrix that transforms the origin of the muscle to the global origin ,and the $origin-insertion$ vector to the Z axis of the global coordinate system.

M_S is the scale matrix, the scale factor of Z axis is S_z=LM_S/LM_0, with $S_x = S_y = 0.4 * S_z$. Here, LM_S is the Euclidean length between the associated *origin bone point* and *insertion bone point* of the ith muscle in the scaled generic model, LM_0 is the Euclidean length between the associated *origin bone point* and *insertion bone point* of the ith muscle in the original generic model. The scale factor in X and Y direction is set be small enough to make sure hat the muscle would lie between the skin. We set the scale factor to 0.4 in our experiment, which produced satisfactory results.

M_0^{-1} is the inverse matrix of M_0, which translates the muscle back to its original position from the global origin.

M_T is a transform matrix that translates the muscle from its unscaled *origin bone point* to the scale *origin bone point*.

M_R is the transform matrix that rotates the muscle along axis OA with angle θ, where $OA = OI \times O_s I_s$, and $\theta =\ acos(\frac{OI \cdot O_s I_s}{|OI| \cdot |O_s I_s|})$.

4.3 Muscle Resampling

Because muscles of the generic model are modeled as irregular triangular meshes, they should be resampled and changed to another form that is suitable for morphing and animation. In this section, we introduce our muscle resampling method that can sample each irregular triangular muscle shape to a regular one. It consists of five ellipses and two extreme points, say, the $origin$ and the $insertion$(Fig. 2(b)). The resampling algorithm consists of five steps:

1. Fit Bezier curve to the points on the *action line*.
2. Shoot rays from each point on the *action line*.
3. Calculate intersections between each ray and the muscle meshes.
4. Fit ellipse to the intersections.
5. Reconstruct and render the resampled muscle meshes.

First, we fit a Bezier curve to the seven *action line* points by solving a set of linear equationsusingth the Gauss-Jordan elimination method [28].

Then, we shoot rays from each point on the *action line* in their normal planes. Twenty rays are shot from each point, the angle between adjacent rays are $\alpha = 2\pi/n, n = 20$. More rays can produce more accurate shape but will also slow down the sampling speed.

In the following step, the intersections between the rays from each *action line* point and the skin mesh are calculated using Möller's method[29], which is one of the fastest ray-triangle intersection algorithms. A ray $R(t)$ with origin O and normalized direction D is defined as $R(t) = O + tD$, and a triangle is defined by three vertices V_0, V_1 and V_2. A point $T(u, v)$ on a triangle is given by $T(u, v) = (1 - u - v)V_0 + uV_1 + vV_2$. Computing the intersection between the ray, $R(t)$, and the triangle, $T(u, v)$, is equivalent to $R(t) = T(u, v)$, which yields:

$$O + tD = (1 - u - v)V_0 + uV_1 + vV_2 \qquad (2)$$

The final solutions can be obtained through the following equation using a trick and Cramer's rule:

$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \frac{1}{P \cdot E_1} \begin{bmatrix} Q \cdot E_2 \\ P \cdot T \\ Q \cdot D \end{bmatrix} \qquad (3)$$

where $E_1 = V_1 - V_0$, $E_2 = V_2 - V_0$ and $T = O - V_0$, $P = (D \times E_2)$ and $Q = T \times E_1)$.

In step 4, we fit ellipse to the sampled intersections calculated in the previous step using the method proposed in [30] which is ellipse-specific, extremely robust, and efficient.

In the final step, we reconstruct the resampled muscle shape and render it with Phong shading. Fig. 2(b)-(d) illustrate this muscle resampling process.

4.4 Muscle Morphing

After the scaling and resampling process, muscles have been aligned properly to the skin skeleton, and are ready for muscle morphing.

Currently, we simply use a iterative method to morph the muscles. Because muscles have been scaled down under the skin mesh, we dilate each elliptic slice of muscles with a small step δ_d until the slice is found to be out of the skin mesh, then we shrink the slice with a smaller step δ_s until the slice is shrunk into the skin mesh again. This process is very similar to the one described by Baraff to detect the colliding contact point during the process of physically-based simulation [31]. Fig. 5(e) shows the generic model and the skin mesh after the muscle morphing process.

5 Animating Target Skin Mesh

After finishing previous steps, we have fit a generic model to the target skin mesh. Then the anatomically based method proposed by Wilhelms *et al.* [14]could be used to animate the target skin mesh.

The target skin is anchored to the underlying bones and muscles, which is called an *anchoring* process. Each vertex in the skin represented with triangle meshes is associated with the closest underlying bones and muscles. And then skin vertices are transformed into the space among the planes of the two slices of a muscle through a *parametric trilinear transformation*. To simulate the elastic effect of skin, springs can be embedded into the skin vertices. Each edge of the triangle mesh of the skin is considered to be a spring with a certain rest length and a stiffness. These springs are brought into equilibrium by means of a series of relaxation operations. Users may refer to [14] for more detailed description.

6 Results

The graphics interface we have developed is written in C++. The generic model and skin are shadered with OpenGL and Cg.

Our program run on a Pentrium 4 computer with 1.8G CPU, 512Mb RAM and GeForce FX5200 graphics card. We spend about two minutes on designing the skeleton of the generic model, one hour on designing muscle parameters. These two steps need be done only once by experienced animators, and then the parameters of the skeleton and muscle are saved as text files on the disk, which can be reused by the user later. Less than one minutes is taken to scale the generic model according to the target skin. The muscle resampling process takes about three minutes, and muscle morphing process takes about four minutes.

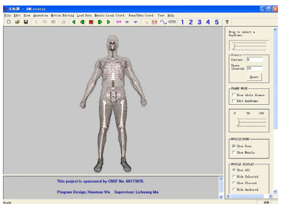


Fig. 4: The graphics interface we developed.

Fig. 4 shows the graphics interface that we have developed. Fig. 5 shows the main process of how the generic model is fitted to the target skin. Fig. 5(a) shows the skin model of John. Fig. 5(b)shows the skin model along with the initial generic model. Notice that the bones of the generic model are away from the corresponding parts of the skin. In Fig. 5(c), the bones of the generic model have been transformed and scaled according to the skin mesh of John. Fig. 5(d) shows the generic model in the skin after the muscle scaling process. Fig 5(e) shows the final model after the muscle morphing stage. Fig 6 demonstrate how the generic model is semi-automatically reshaped to Eric's Skin. These two examples show that the generic model had been successfully fitted to the target skin mesh.

7 Conclusion and Future Works

In this paper, we proposed a generic-model based human-body modeling method. The generic model need to be constructed only once by skilled modelers, users are not required to concern about this process. Given a new skin mesh, the generic model can be semi-automatically reshaped according to the shape of the skin. There are very little work remains for the users to do. Generic-model based method is helpful for constructing model human bodies with anatomical structure in a easy and fast way.

The limitations of our method include: (1) The muscle morphing algorithm is relatively slow and should be improved in the future; (2) We do not consider the deformations of head, hands and feet; (3) The shapes of bones and muscles are just an approximation, no proper criteria have been established to guide our generic model scaling and muscle morphing process.

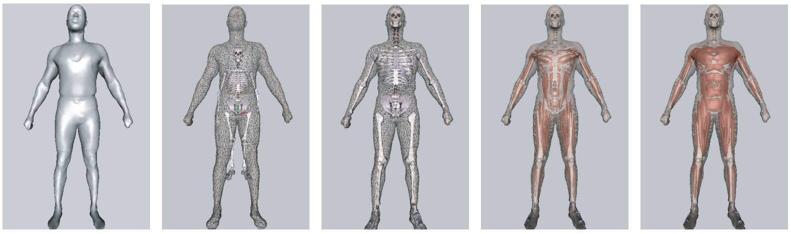
Our future work include: design new algorithm to accelerate the muscle morphing process, establish adequate criteria to guide the generic model scaling and muscle morphing process.

8 Acknowledgement

We would like to thank Cyberware for generously providing 3D whole body samples on their web. We would also like to thank Tomas Möller for his ray/triangle intersection code, thank Andrew Fitzgibbon for his ellipse-fitting Matlab code. Thank Brett Allen for his helpful advice. This work was supported by National Natural Science Foundation of China (grant No.60173035 and No.60373070), ,863 Program of China (grant No.2003AA411310) and Microsoft Research Asia (Project-2004-Image-01).

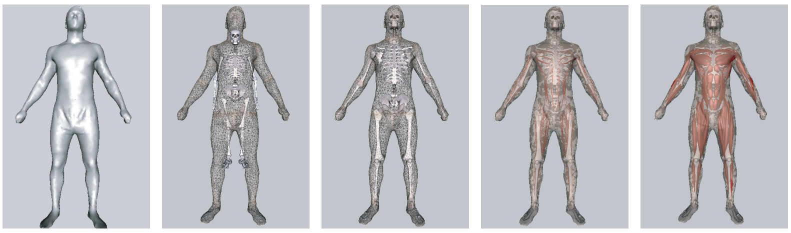
References

1. Phillip E. Pack. *CliffsQuickReview Anatomy & Physiology*. New York Cliffs Notes, 2001.
2. J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. *Computer Graphics(Pro. of SIGGRAPH'00)*, pages 165–172, August 2000.
3. A. Mohr and M. Gleicher. Building efficient, accurate character skins from examples. *ACM Transactions on Graphics*, 22(3):562 – 568, July 2003.



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

Fig. 5: Given a skin mesh, the generic model can be semi-automatically transformed according to the shape of the skin. (a) The skin model of John. (b) The skin model with the generic model before transformation. (c) After transformation and scaling of bones of the generic model. (d) After muscle scaling process. (e) The final generic model after muscle morphing.



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

Fig. 6: The same process as in Fig. 5, but with difference skin mesh of Eric's.

4. Y. Lee, D. Terzopoulos, and K. Waters. Realistic face modeling for animation. In *SIGGRAPH 95 Conference Proceedings*, pages 55–62, August 1995.
5. K. Kähler, J. Haber, and H.-P. Seidel. Geometry-based muscle modeling for facial animation. In *Proc. of Graphics Interface'01*, pages 37–46, 2001.
6. K. Kähler, J. Haber, H. Yamauchi, and H.-P. Seidel. Head shop: Generating animated head models with anatomical structure. In *Pro. of ACM SIGGRAPH Symposiumon Computer Animation (SCA) 2002*, pages 55–64, July 2002.
7. M. Simmons, J. Wilhelms, and A. Van Gelder. Model-based reconstruction for creature animation. In *Pro. of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer Animation*, pages 139–146, July 2002.
8. B. Allen, B. Curles, and Z. Popvić. Articulated body deformation from range can data. *ACM Transactions on Graphics*, 21(3):612–619, July 2002.
9. P. J. Schneider and J. Wilhelms. Hybrid anatomically based modeling of animals. In *Proc. of Computer Animation'98*, pages 161–169, June 1998.
10. J. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235–256, July 1982.
11. J. Bloomenthal. Hand crafted. *Siggraph Course Notes 25*, 1993.
12. S. Yoshimito. Ballerinas generated by a personal computer. *The Journal of Visualization and Computer Animation*, 3:85–90, 1992.
13. F. Scheeppers, R. E. Parent, W. E. Carlson, and S. F. May. Anatomy-based modeling of the human musculature. *Computer Graphics (Pro. of SIGGRAPH'97)*, 31:163 – 172, August 1997.

14. J. Wilhelms and A. Van Gelder. Anatomically based modeling. *Computer Graphics (Proc. of SIGGRAPH'97)*, pages 173–180, 1997.
15. K. Komatsu. Human skin model capable of natural shape variation. *The Visual Computer*, 3(5):265–271, March 1988.
16. N. Magnenat-Thalmann and D. Thalmann. The direction of synthetic actors in the film rendez-vous à montréal. *IEEE Computer Graphics and Applications*, 7(12):9–19, 1987.
17. N. Magnenat-Thalmann, R. Laperriere, and D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *Proc. of Graphics Interface'88*, pages 26–33, 1988.
18. D. Thalmann, J. Shen, and E. Chauvineau. Fast realistic human body deformations for animation and vr applications. In *Proc. of Computer Graphics International'96*, pages 166–174, June 1996.
19. J. Lander. Skin them bones: game programming for the web generation. *Game Developer Magazine*, pages 11–16, May 1998.
20. Xiaohuan Corina Wang and Cary Phillips. Multi-weight enveloping: least-squares approximation techniques for skin animation. In *Pro. of the 2002 ACM SIGGRAPH/Eurographics symposium on computer animation*, pages 129 – 138, July 2002.
21. Peter Sand, Leonard McMilla, and Jovan Popvić. Continuous capture of skin deformation. *ACM Transactions on Graphics*, 22(3):578–586, July 2003.
22. National Library of Medicine. The visible human project, *http://www.nlm.nih.gov/research/visible*.
23. Feng Dong, Gordon J. Clapworthy, Meleagros A. Krokos, and Jialiang Yao. An anatomy-based approach to human muscle modeling and deformation. *IEEE Transactions on visualization and computer graphics*, 8(2):154–170, April 2002.
24. Luciana Porcher Nedel and Daniel Thalmann. Anatomic modeling of deformable human bodies. *The Visual Computer*, 16:306–321, 2000.
25. Cyberware whole body color 3d scanner, *http://www.cyberware.com/products/wbinfo.html*.
26. Cyberware sample models on the web, *http://www.cyberware.com/samples/index.html*.
27. Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH 97 Conference Proceedings*, pages 209–216, August 1997.
28. William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C.* Cambridge University Press, 1992.
29. Tomas Möller and Ben Trumbore. Fast, minimum storage ray-triangle intersection. *Journal of Graphics Tools*, 2(1):21–28, 1997.
30. M. Fitzgibbon, A. W.and Pilu and R. B. Fisher. Direct least-squares fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, May 1999.
31. D. Baraff. Rigid body simulation. *Physically Based Modeling Course Notes 34, SIGGRAPH '95*, pages G1–G68, 1995.