



SMI 2013

Shape modeling for animated characters using ordinary differential equations

E. Chaudhry^a, L.H. You^a, X. Jin^b, X.S. Yang^a, J.J. Zhang^{a,c,*}^a National Centre for Computer Animation, Bournemouth University, United Kingdom^b State Key Lab of CAD & CG, Zhejiang University, China^c Traction Power State Key Laboratory, Southwest Jiaotong University, China

ARTICLE INFO

Article history:

Received 18 March 2013

Received in revised form

1 June 2013

Accepted 1 June 2013

Available online 11 June 2013

Keywords:

Shape modeling

Static skin deformations

Fourth order ordinary differential equations

Finite difference solution

Curve-based surface modeling and character animation

ABSTRACT

In this paper, we develop a new approach to animate skin deformation of character models. It aims to combine the strengths of joint-based approaches, physics-based algorithms and curve-based surface modeling methods together for efficient and realistic animation of skin deformation. We first transform the deformation of skin surfaces of character models into that of the curves defining the skin surfaces, and introduce a mathematical model consisting of a vector-valued fourth order ordinary differential equation and boundary conditions to describe the curve deformation. In order to achieve capacity and high animation efficiency, we propose an efficient finite difference solution of the mathematical model, and apply our proposed solution to animate skin deformation of character models. The application examples demonstrate that our proposed approach can create realistic skin deformations for real-time character animation.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Skin deformation plays a very important role in creating realistic and efficient character animation. How to create high-fidelity skin deformations quickly is one of the most challenging areas of computer animation. To this end, various skin deformation techniques have been developed. Among them, joint-based [1], physics-based [2], and example-based techniques [3] are widely applied, and curve-based surface modeling [4] also has some applications.

Joint-based techniques are the most popular in the animation industry because they seem intuitive. However, the animator must spend time and effort to manually manipulate the skin surface in relation to the motion of the skeleton. This is because the relationship between skin deformation and skeleton movement must be manually tuned by applying proper weights. Physics-based approaches consider anatomy and biomechanics and can create more realistic skin deformations. However, they involve a lot of numerical calculations, and reduce the efficiency of animating skin deformations. Example-based methods use example skin shapes to improve the realism of skin deformations. Although they do not require any manual skills to specify the weights required by

skeleton-based techniques, sufficient example skin shapes must be used to achieve realistic skin deformations. Curve-based surface modeling transforms modeling and deformations of surfaces into those of curves. It is very efficient but requires investigations into the relationships between surfaces and the curves defining the surfaces.

As discussed above, there is not one complete method for skin deformation that can meet all the requirements of the animation industry like realism, effectiveness and less computational time. Each method has its strengths and weaknesses. This paper aims to combine the strengths of joint-based approaches, physics-based algorithms, and curve-based surface modeling methods, and uses two example skin shapes for efficient and realistic animation of skin deformations. It uses surface curves to define the skin shape, example skin shapes and the underlying physical law of curve deformations to improve realism, and employs curve shape changes to drive skin deformations for reducing computational cost. In order to maximize the capacity of our proposed approach, an efficient finite difference solution of the proposed mathematical model is developed. It can effectively tackle the problems of line distribution forces in local regions and/or concentrated forces which are difficult to solve with analytical approaches.

Our proposed approach first transforms a skin surface into a set of curves defining the skin surface. Then, the forces acting on the set of curves which drive the skin surface to deform from an initial pose to a final pose are determined using our proposed physics-based deformation algorithm similar to that of beam bending.

* Corresponding author at: National Centre for Computer Animation, Bournemouth University, United Kingdom. Tel.: +44 1202 965055.

E-mail addresses: jzhang@bournemouth.ac.uk, ehtzaz@yahoo.com (J.J. Zhang).

Next, the forces at any poses between the initial and final poses are obtained using interpolation. With the obtained forces, our proposed physics-based algorithm will generate the deformation of all the curves and create the deformed skin surface.

2. Related work

Skin deformation techniques can be classified into three major categories: surface-based, volume-based and curve-based techniques [5].

Among various surface-based techniques, joint-based techniques deform skin surfaces through the transformations associated with the joints of the skeleton of a character model. Since the transformations do not involve the physics of skin deformations, skin shapes are achieved by manually applying appropriate weights to modify the transformations [6–13].

With volume-based techniques, biomechanics and anatomical structures are introduced to develop more accurate models and to achieve realistic skin deformation. These models usually comprise a skeleton, muscles, other tissues, and skin. The involvement of muscles and other tissues causes volumetric deformation [14–16].

Curve-based techniques [17–21] use the curves on skin surfaces to define character models. When animating skin deformation, these curves are deformed and the deformed curves are used to change skin surfaces into other shapes.

Example-based techniques are surface-based. In example-based techniques [10,22–26], the artist models different shapes and poses of the characters or reconstructs them from the data consisting of range scans of a human body in a variety of poses. New poses are interpolated from these poses.

Physics-based techniques can be surface-based or volume-based. They used anatomy, elastic mechanics, or biomechanics of skin deformation originating from the movements of muscles and tendons [27–33].

Recently, Kavan and Sorkine developed a closed form skinning method that combines joint-based techniques with physics-based techniques. This method minimizes an elastic energy function to optimize skinning weights [34].

The work given in this paper aims to take advantage of the strengths of skeleton-based skinning, physics-based approaches and curve-based surface modeling, and use two example skin shapes to improve realism and efficiency of animating skin deformations. In what follows, we first discuss the relationships between skin surfaces and curves defining the skin surfaces. Then we introduce curve-based static skin deformations. Next, we present a number of examples of skin deformations with our proposed approach followed by a conclusion.

3. Defining geometric models

Geometric models can be defined by four types of surfaces in Autodesk Maya: polygons, subdivision surfaces, patch modeling and with curves. In Autodesk Maya, a surface model can be changed into a wireframe model or created from a set of curves through skinning these curves. Curves can also be represented by Bézier, B-spline and NURBS.

In addition, the solution to a vector-valued fourth order ordinary differential equation subjected to suitably specified boundary conditions also gives a curve. Different from Bézier, B-spline and NURBS curves, the curves created by the solution to a vector-valued fourth order ordinary differential equation can be regarded as physics-based since the differential equation can be derived from the same methodology as that of beam bending. In

the subsections below, we will use such curves to define surface models.

3.1. Basic principle

For curve-based surface modeling, the shape of a surface can be described by the curves on the surface. In order to use surface curves to animate a surface, we must find out the relationships between surface curves and the surface described by these curves. They include how to obtain the surface curves describing a surface from a polygon model, and how to transfer the deformations of the surface curves to the polygon model. We will discuss these two issues in the following subsections.

3.2. Converting a polygonal model into a curve defined surface model

First, we divide a model into different parts or surface patches (complicated parts should be decomposed into two or more simpler surface patches). Second, we create the curves defining each part or surface patch. To do this, we draw a rough central curve within each of the parts or surface patches. Then, some points at equal intervals are selected on the curve. From each point, we create a plane perpendicular to the curve. We use this plane to find out the intersecting points between the plane and the part or surface patch which forms a curve on the part or surface patch. With this method, we obtain all the curves defining the part model or surface patch and the central curve determined from the curves as shown in Fig. 1a. Then we transform these curves into those in the direction of the arm as indicated in Fig. 1b. Using the above curve creation method, we obtained the curves defining a human arm, human hand and complete human model with segments as shown in Fig. 1.

3.3. Converting a curve defined model back to a polygonal model

After the curves describing a polygon model are obtained, we can deform these curves and transfer the deformations of these curves to the polygon model to achieve the new shape of the polygon model. With this method, we can animate a polygon model efficiently.

There are two approaches which can be used for this purpose. One is to create a new model from these curves through the skinning method available in Maya software package and the other is to relate all surface points of the polygon model to these curves [20,21]. In this paper, we have used both approaches to generate deformed models.

4. Static skin deformation

In addition to the relationships between surface curves and the surface described by these curves, another main issue is how to deform the curves describing the surface realistically.

For static deformations, the mathematical model will consist of a time-independent vector-valued fourth order ordinary differential equation and boundary conditions. The purpose of introducing a fourth order ordinary differential equation to describe the deformations of curves is that the equation is similar to that of beam bending and it considers physical properties with an externally applied force to change the shape of curves. Therefore, it can produce more realistic deformation of curves than purely geometric shape manipulation.

Since the finite difference solution of the ordinary differential equation dealing with static deformations is very efficient and it has a capacity in tackling the problems of line distribution forces in local

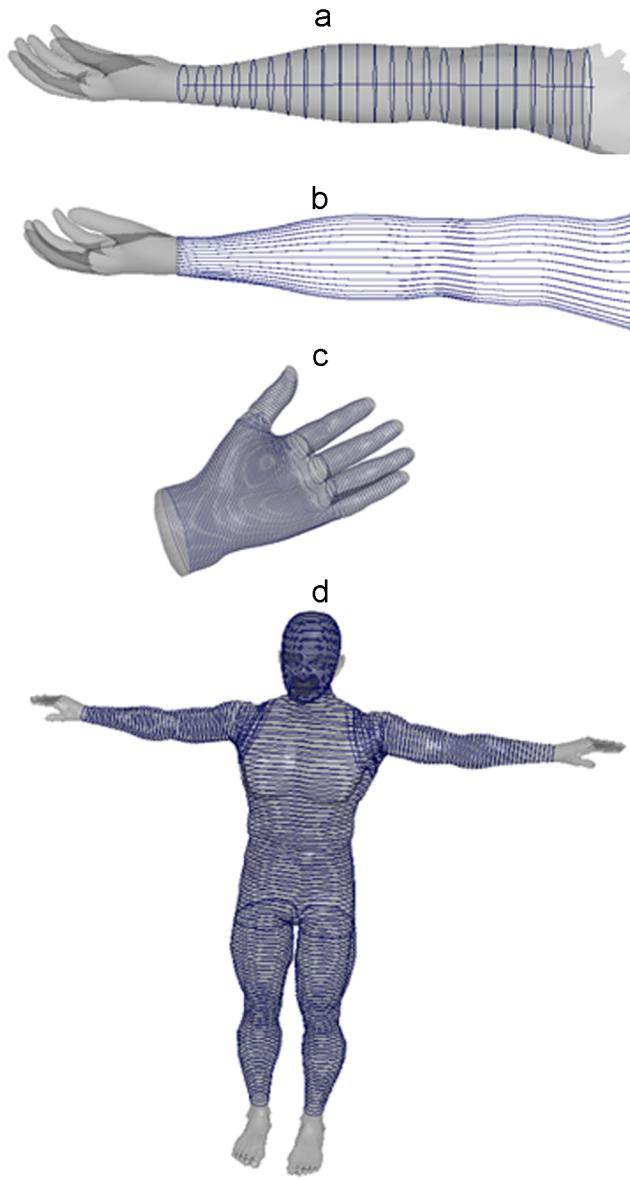


Fig. 1. Curves creation from the polygon mesh and surface curves.

regions and/or concentrated forces which are difficult to cope with by analytical solutions, we develop such a solution below.

4.1. Mathematical model

The curves on skin surfaces will change their shapes with the joint rotation of the skin surfaces. These shape changes are caused by the varying forces acting on the curves. We will investigate how to relate the varying forces to the joint rotation in our future work. In this paper, we focus on the deformation determination of the curves subjected to the varying forces which will be determined from example skin shapes as discussed below.

Similar to the equation of beam bending, the static deformations of a curve can be represented by a vector-valued static fourth order ordinary differential equation which has the following form:

$$a_1 \frac{d^4 q(v)}{dv^4} + a_2 \frac{d^2 q(v)}{dv^2} = F(v) \quad (1)$$

where a_1 and a_2 are vector-valued shape control parameters which can be determined from physical and geometric properties of skin

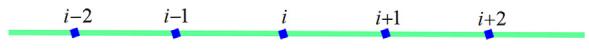


Fig. 2. Representative node i .

surfaces, $q(v)$ is a vector-valued deformation function, and $F(v)$ is a vector-valued externally applied varying force.

The boundary conditions of beam bending consist of the positions and rotations at the two ends of curves. Similarly, the boundary conditions for the curve deformation can be represented by the following equation:

$$\begin{aligned} v=0, \quad q=\bar{q}, \quad \frac{dq}{dv}=0 \\ v=1, \quad q=\tilde{q}, \quad \frac{dq}{dv}=0 \end{aligned} \quad (2)$$

where v is a scalar parametric variable, $v=0$ and $v=1$ indicate the two ends of a curve. The rotations at the two ends are set to zero to achieve tangent continuity for smooth transition between two adjacent surface patches.

Eqs. (1) and (2) define the mathematical model of deformations of the curves describing a skin surface.

4.2. Solution of mathematical model

According to the central finite difference approximation, we can find the finite difference formula (3) of the first derivative, (4) of the second derivative, and (5) of the fourth derivative at a representative node i as shown in Fig. 2.

$$\left(\frac{dq}{dv} \right)_i = \frac{1}{2h} [q_{i+1} - q_{i-1}] \quad (3)$$

$$\left(\frac{d^2 q}{dv^2} \right)_i = \frac{1}{h^2} [q_{i+1} - 2q_i + q_{i-1}] \quad (4)$$

$$\left(\frac{d^4 q}{dv^4} \right)_i = \frac{1}{h^4} [6q_i - 4(q_{i-1} + q_{i+1}) + q_{i-2} + q_{i+2}] \quad (5)$$

where h is the interval between two adjacent nodes. Substituting Eqs. (4) and (5) in Eq. (1), we obtain the finite difference equation of ODE (1) at the representative node i which is given in the following equation:

$$\begin{aligned} \frac{a_1}{h^4} [6q_i - 4(q_{i-1} + q_{i+1}) + q_{i-2} + q_{i+2}] \\ + \frac{a_2}{h^2} [q_{i+1} - 2q_i + q_{i-1}] = F_i \end{aligned} \quad (6)$$

For each curve, we uniformly divided it into N equal intervals as indicated in Fig. 3. In the figure, nodes 0 and N are boundary nodes, -1 and $N+1$ are two virtual nodes used to indicate the nodes outside the boundary nodes 0 and N , and nodes $1, 2, \dots, N-1$ are inner nodes. Using Eq. (6), we can obtain the finite difference equations for all the inner nodes. Put the finite difference equations for all inner nodes together, we obtain $N-1$ linear algebra equations which can be used to determine $N-1$ unknown constants $1, 2, \dots, N-1$.

For inner nodes 2 and $N-2$, their finite difference equations involve boundary nodes 0 and N . For inner nodes 1 and $N-1$, their finite difference equations involve both boundary nodes 0 and N and virtual nodes -1 and $N+1$. Therefore, we require four more equations to determine the unknown constants at nodes $-1, 0, N$ and $N+1$. This can be done by considering boundary conditions (2) and finite difference formula (3) whose finite difference equations are as follows:

$$q_0 = \bar{q}$$

$$q_1 = q_{-1} = 0$$

$$\begin{aligned} q_N &= \tilde{q} \\ q_{N+1} &= q_{N-1} = 0 \end{aligned} \quad (7)$$

Combining Eqs. (6) and (7) together and writing them in the form of matrix, the following equation is reached:

$$KX = F \quad (8)$$

We obtain the solution of Eq. (8) by using the inverse of matrix K to multiply the both sides of Eq. (8).

$$X = K^{-1}F \quad (9)$$

For the initial pose 0 as shown in Fig. 4, we can determine the force at the pose, which is

$$\begin{aligned} F_{i,0} &= \frac{a_1}{h^4}[6q_{i,0}-4(q_{i-1,0}+q_{i+1,0})+q_{i-2,0} \\ &\quad +q_{i+2,0}]+\frac{a_2}{h^2}[q_{i+1,0}-2q_{i,0}+q_{i-1,0}] \quad (i=1,2,\dots,N-1) \end{aligned} \quad (10)$$

where the second subscript 0 indicates the initial pose, and the first subscript indicates the node index.

For the final pose J as shown in Fig. 5, we can also determine the force at the pose, which is

$$\begin{aligned} F_{i,J} &= \frac{a_1}{h^4}[6q_{i,J}-4(q_{i-1,J}+q_{i+1,J})+q_{i-2,J} \\ &\quad +q_{i+2,J}]+\frac{a_2}{h^2}[q_{i+1,J}-2q_{i,J}+q_{i-1,J}] \\ &\quad (i=1,2,3,\dots,N-1) \end{aligned} \quad (11)$$

Then, we can calculate the difference of the forces between the initial and final poses, using the following equation:

$$\Delta F_i = F_{i,J} - F_{i,0} \quad (i=1,2,\dots,N-1) \quad (12)$$

Having known the force difference, we can calculate the deformation at any poses between the initial and final poses. For example, if we want to calculate the deformation at the pose j ($0 < j < J$), we can first generate the following equation:

$$\begin{aligned} \frac{a_1}{h^4}[6q_{i,j}-4(q_{i-1,j}+q_{i+1,j})+q_{i-2,j} \\ &\quad +q_{i+2,j}]+\frac{a_2}{h^2}[q_{i+1,j}-2q_{i,j}+q_{i-1,j}] = F_{i,j} \\ &\quad (i=1,2,\dots,N-1; j=1,2,\dots,J-1) \end{aligned} \quad (13)$$

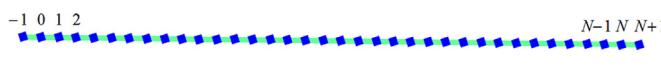


Fig. 3. Nodes of a curve.

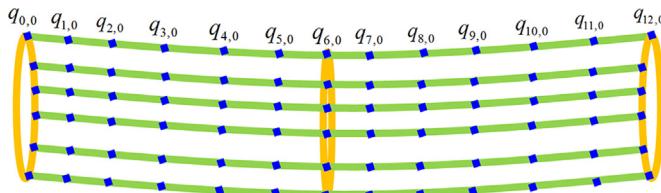


Fig. 4. Initial pose 0.

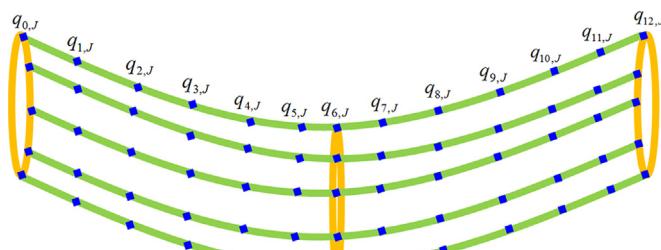


Fig. 5. Final pose j .

where

$$F_{i,j} = \frac{j \times \Delta F_i}{J} \quad (i=1,2,\dots,N-1; j=1,2,\dots,J-1) \quad (14)$$

Substituting Eq. (14) in Eq. (13), producing Eq. (8) using Eq. (13) and boundary conditions (7), and solving Eq. (8), we obtain the curve deformation given by Eq. (9) at the pose j , which can be used to create the skin deformation at the pose.

In addition to automatic determination of skin deformations using the above physics-based algorithm, our proposed approach also allows animators to apply their control over skin deformations. If animators are not satisfied with skin shapes at some poses, they can manually modify the skin surfaces at these poses. Then the modified skin surfaces at these poses are taken to be example skin shapes and used to determine skin deformations at other poses through our proposed physics-based deformation algorithm.

5. Application examples

We have implemented our proposed method with C++ and OpenGL. Several application examples are presented to demonstrate our proposed approach.

In the first example, we have used our proposed approach to animate two models of a human arm as shown in Figs. 6 and 7. The arm model used in Fig. 6 consists of 31 curves and each curve has 103 nodes. The arm model used in Fig. 7 consists of 34 curves and each curve has 27 nodes. Here we demonstrate how to determine skin deformations with Fig. 7. The skin shapes at the initial and final poses are given in Fig. 7a and f, respectively. The skin shapes at the poses between the initial and final poses are depicted in Fig. 7b–e. These images indicate that our proposed approach creates plausible skin shapes.

The second example is to animate skin deformations of a hand finger with our proposed approach. The first and the final images shown in Fig. 8 are from the initial and final poses, respectively, and those between the first and final poses are from our calculated results.

In the third example, a human leg is animated using our proposed approach. The first and final images shown in Fig. 9 are from the initial and final poses, respectively, and those between the first and final poses are from our calculated results. It can be observed that our proposed approach generates a realistic skin deformation of the human leg.

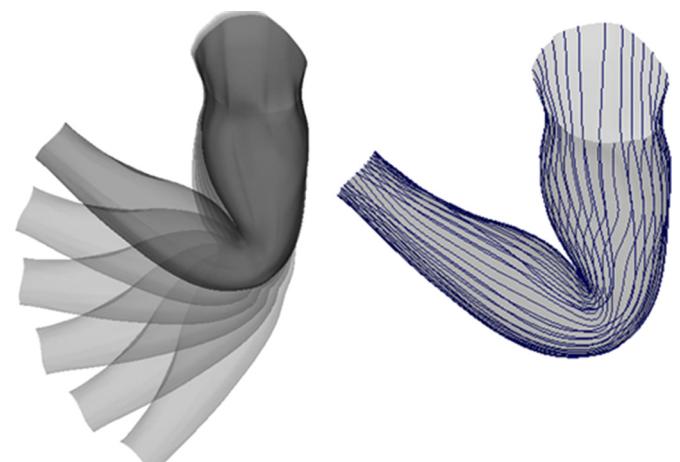


Fig. 6. Human arm animation.

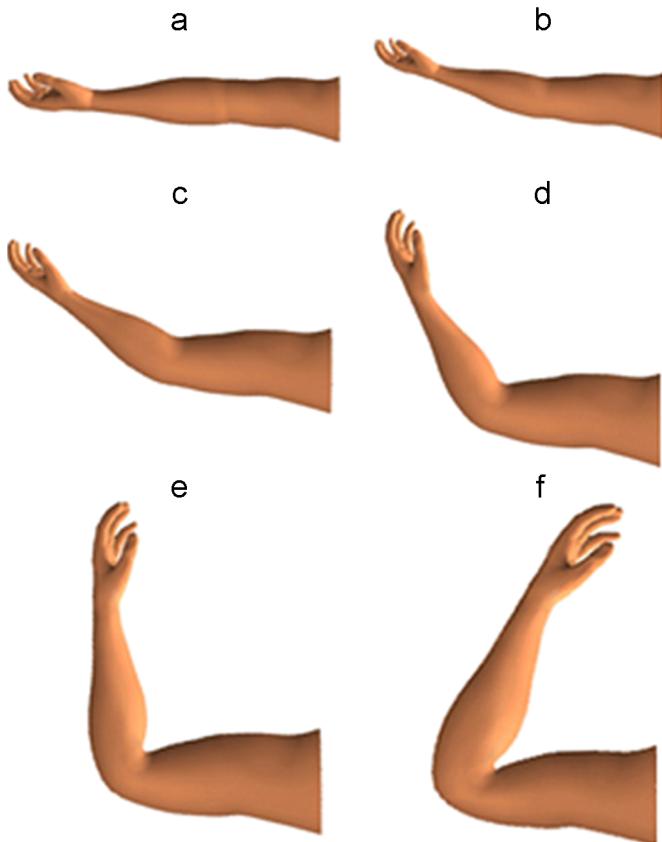


Fig. 7. Human arm animation and skin deformation from initial pose $j=0$ to final pose $j=5$.

We have made a comparison in Autodesk Maya between two skinning methods and our proposed approach in Fig. 11a–c where Fig. 11a and b are from classic linear skinning method and dual quaternion skinning method, respectively, and Fig. 11c is from our proposed approach. Clearly, classic linear skinning method without using the painting tool to apply the weights causes the problem of a collapsing joint, and dual quaternion skinning method solves the problem but still creates less realistic skin deformations at the joint. In contrast, our proposed approach creates more realistic skin deformation of the human leg.

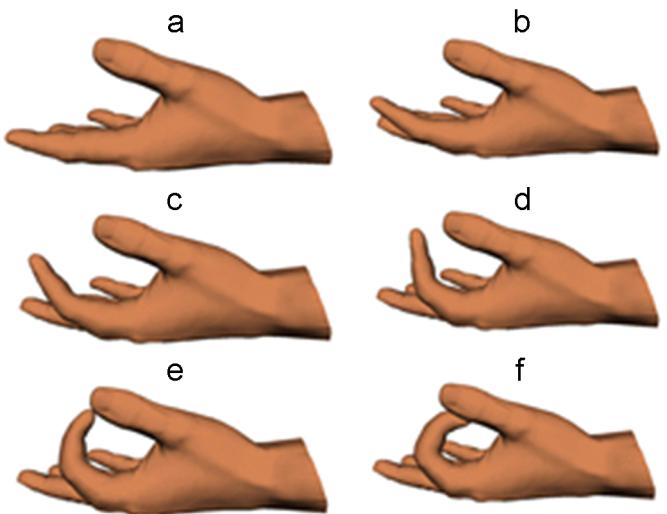


Fig. 8. Human finger animation and skin deformation from initial pose $j=0$ to final pose $j=5$.

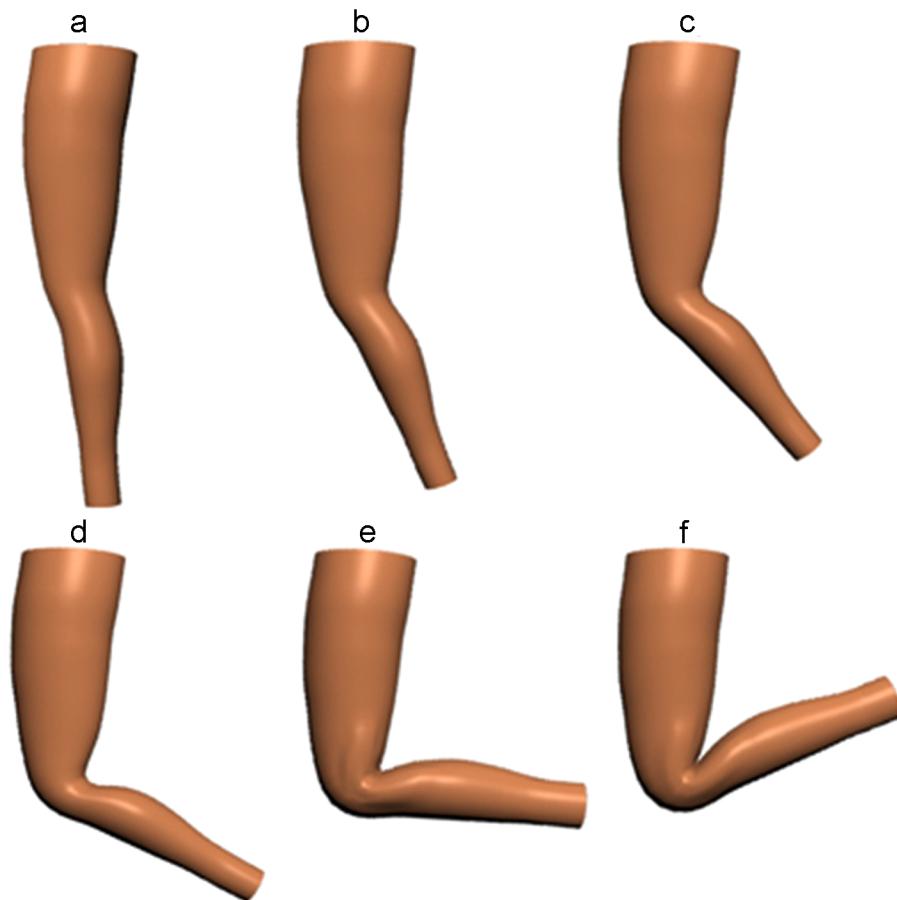


Fig. 9. Skin deformation of a human leg (a) initial pose 0 (b) pose 1 (c) pose 2 (d) pose 3 (e) pose 4 and (f) final pose 5.

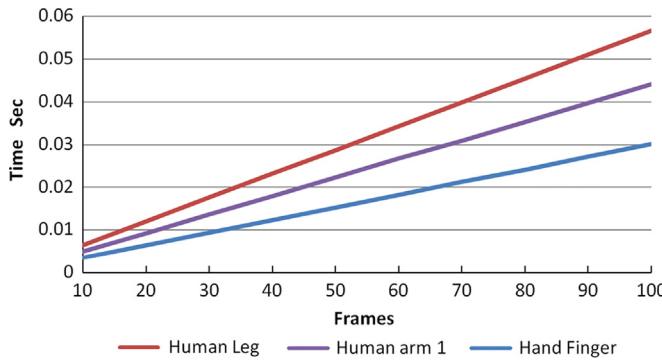


Fig. 10. Relationship between the total time and frames.

Table 1
Computation time of four models.

Application examples	No. of curves	No. of nodes	Time (s) to get K^{-1}	Time (s) for 30 frame	Total time for 30 frame
Hand finger	64	22	0.00028	0.00898	0.00926
Human leg	36	31	0.00066	0.01680	0.01746
Human arm1	34	27	0.00045	0.01305	0.01350
Human arm2	31	103	0.00204	0.17815	0.18019

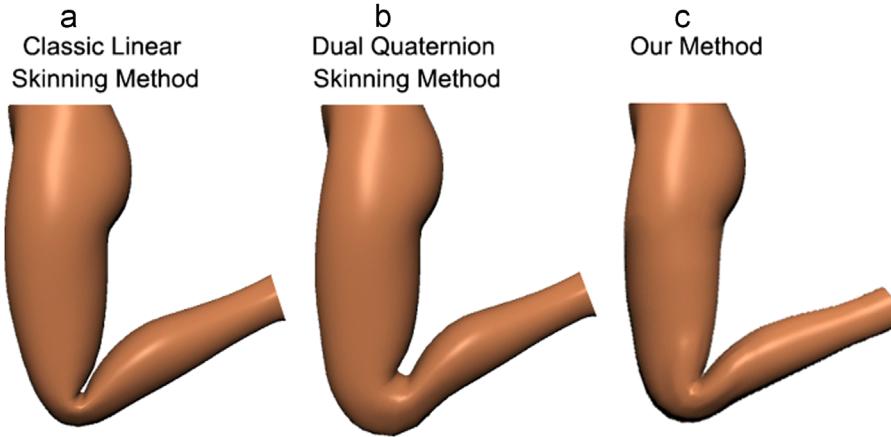


Fig. 11. Comparison between our skin deformation method and other methods. (a) This method creates less realistic skin deformations at the joint. (b) This method causes the skin surface at joints to become bigger. (c) Human leg animated by our method.

Our timing test demonstrates high efficiency of our proposed approach effectively. The test was carried out on a 3 GHz Intel Core2 Duo PC with 2.98 GB of memory. We used the Gauss–Jordan elimination to find the inverse K^{-1} of the matrix K . For each of the four models shown in Figs. 6–9, we used different numbers of surface curves and different numbers of vertices to demonstrate the efficiency of our proposed method. The obtained results were given in Table 1. In the table, the second column indicates the number of the curves used to define each of the 4 models. The third column denotes the number of nodes of each curve. The fourth column gives the time used to find the inverse K^{-1} of the matrix K . The fifth column is the time spent on Eq. (9) for 30 frames and the sixth column is the total time used to create 30 frames for each of 4 models. How the total time changes with the increase of the frames for the first three models is plotted in Fig. 10. It can be seen from the table that when the order of the matrix is small, say for example 27 by 27, the time used to find the inverse K^{-1} of the matrix K is very small (0.00045 s for a matrix of 27 by 27). However, when the order becomes large enough, say for example 103 by 103, the time used to find the inverse K^{-1} of the matrix K increases to 0.00204 s.

Fig. 10 indicates that the total time increases linearly with the increase of the frames for all the models. However, the rate of the increase for different models differs largely. The models require more time to create one frame increase more quickly than those requiring less time to create one frame.

6. Conclusions and future work

In this paper, we have presented a new approach to curve based skin deformation which is based on a mathematical model and a fast finite difference solution of static skin deformation. We have applied our proposed approach to animate the skin deformation of human arms, fingers and legs. We have also made a comparison among two important skinning methods and our proposed one. These application examples demonstrate that our proposed approach can create realistic skin deformation of character models. We have tested the computational efficiency of our proposed approach. The test demonstrates that our proposed method is very efficient and can be used to create skin deformation for real-time character animation.

The work described in this paper does not tackle the collision problem. We will investigate how to solve it in future work. The limitation of this paper requiring two example skin shapes can be overcome by determining the force using the information of the joint rotation. In our future work, we will investigate the relationship between the force and joint rotation. Our method can be used for any complex models such as complete human models including hands and feet. In future, we will generalize our method so that it can work on all kind of models. We also intend to integrate the modeling of character models and the animation of skin deformation together, and introduce time-dependent ordinary differential equation to consider dynamic skin deformations.

Acknowledgments

This research is supported by the grants of the UK Royal Society International Joint Projects/NSFC 2010 and the Sino-UK Higher Education Research Partnership for Ph.D. Studies Project. Xiaogang Jin was supported by the National Natural Science Foundation of China (Grant nos. 61272298, 60933007), and the Major Science and Technology Innovation Team (Grant no. 2010R50040).

References

- [1] Wang XC, Phillips C. Multi-weight enveloping: least-squares approximation techniques for skin animation. In: Proceedings of the 2002 ACM SIGGRAPH/eurographics symposium on computer animation, ACM; 2002. p. 129–138.
- [2] Capell S, Burkhardt M, Curless B, Duchamp T, Popović Z. Physically based rigging for deformable characters. In: Proceedings of the 2005 ACM SIGGRAPH/eurographics symposium on computer animation, ACM; 2005. p. 301–310.
- [3] Allen B, Curless B, Popović Z. Articulated body deformation from range scan data. In: Proceedings of SIGGRAPH'02; 2002. p. 612–619.
- [4] Shin SY, Pyun H, Shin HJ, Sung S, Shin Y. On extracting the wire curves from multiple face models for facial animation. *Comput Graph* 2004;28:757–65.
- [5] Chaudhry E, You LH, Zhang JJ. Character skin deformation: a survey. In: Proceedings of seventh international conference on computer graphics, imaging and visualization. IEEE Press; 2010. p. 41–48.
- [6] Thalmann NM, Laperrière R, Thalmann D. Joint-dependent local deformations for hand animation and object grasping. In: Proceedings on graphics interface'88. Canadian Information Processing Society; 1988. p. 26–33.
- [7] Lander J. Skin them bones: game programming for the web generation. *Game Dev Mag* 1998:11–6.
- [8] Lander J. Over my dead, polygonal body. *Game Dev Mag* 1999:11–6.
- [9] Weber J. Run-time skin deformation. In: Proceedings of game developers conference; 2000.
- [10] Mohr A, Gleicher M. Building efficient, accurate character skins from examples. *ACM Trans Graph (SIGGRAPH'03)* 2003;22(3):562–8.
- [11] Kavan L, Žára J. Spherical blend skinning: a real-time deformation of articulated models. In: Proceedings of the 2005 symposium on interactive 3D graphics and games, ACM; 2005. p. 9–16.
- [12] Yang X, Somasekharan A, Zhang JJ. Curve skeleton skinning for human and creature characters. *Comput Anim Virt Worlds* 2006;17(3–4):281–92.
- [13] Kavan L, Collins S, Žára J, O'Sullivan C. Geometric skinning with approximate dual quaternion blending. *ACM Trans Graph* 2008;27(4):105:1–23.
- [14] Scheepers F, Parent RE, Carlson WE, May SF. Anatomy-based modeling of the human musculature. In: Proceedings of SIGGRAPH'97; 1997. p. 163–172.
- [15] Wilhelms J, Van Gelder A. Anatomically based modeling. In: Proceedings of SIGGRAPH'97; 1997. p. 173–180.
- [16] Yang XS, Zhang JJ. Automatic muscle generation for character skin deformation. *Comput Anim Virt Worlds* 2006;17:293–303.
- [17] Shen J, Thalmann NM, Thalmann D. Human skin deformation from cross sections. In: Proceedings of computer graphics international; 1994. p. 9–4.
- [18] Hyun DE, Yoon SH, Chang JW, Kim MS, Jüttler B. Sweep-based human deformation. *Visual Comput* 2005;21:542–50.
- [19] Yoon SH, Kim MS. Sweep-based freeform deformations. *Comput Graph Forum* 2006;25:487–96.
- [20] You LH, Yang XS, Zhang JJ. Dynamic skin deformation with characteristic curves. *Comput Anim Virt Worlds* 2008;19(3–4):433–44.
- [21] You LH, Chaudhry E, You XY, Zhang JJ. Physics and example based skin deformations for character animation. In: Proceedings of 11th IEEE international conference on computer-aided design and computer graphics, IEEE; 2009. p. 62–67.
- [22] Lewis JP, Cordner M, Fong N. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In: Proceedings of SIGGRAPH'00; 2000. p. 165–172.
- [23] Rhee T, Lewis JP, Neumann U. Real-time weighted pose-space deformation on the GPU. *Comput Graph Forum* 2006;25:439–48.
- [24] Kurihara T, Miyata N. Modeling deformable human hands from medical images. In: Proceedings of the 2004 ACM SIGGRAPH/eurographics symposium on computer animation. Eurographics Association; 2004. p. 355–363.
- [25] Le BH, Deng Z. Smooth skinning decomposition with rigid bones. *ACM Trans Graph (SIGGRAPH Asia'12)* 2012;31(6):199:1–10.
- [26] Weber O, Sorkine O, Lipman Y, Gotsman C. Context-aware skeletal shape deformation. *Comput Graph Forum* 2007;26(3):265–74.
- [27] Wilhelms J, Gelder AV. Anatomically based modeling. In: Proceedings of SIGGRAPH'97; 1997. p. 173–180.
- [28] Nedel L, Thalmann D. Modeling and deformation of the human body using an anatomically-based approach. In: Proceedings of the computer animation. IEEE Computer Society; 1998. p. 34–40.
- [29] Nedel L, Thalmann D. Anatomic modelling of deformable human bodies. *Visual Comput* 2000;16:306–21.
- [30] Aubel A, Thalmann D. Interactive modeling of the human musculature. In: Proceedings of fourteenth conference on computer animation; 2001. p. 167–255.
- [31] Simmons M, Wilhelms J, Van Gelder A. Model-based reconstruction for creature animation. In: Proceedings of fourteenth conference on computer animation. IEEE; 2001. p. 167–255.
- [32] Guo Z, Wong KC. Skinning with deformable chunks. *Comput Graph Forum (EUROGRAPHICS'05)* 2005;24(3):373–81.
- [33] Venkataraman K, Lodha S, Raghavan R. A kinematic-variational model for animating skin with wrinkles. *Comput Graph* 2005;29(5):756–70.
- [34] Kavan L, Sorkine O. Elasticity-inspired deformers for character articulation. *ACM Trans Graph (SIGGRAPH Asia'12)* 2012;31(6):196:1–8.