Stretch It - Realistic Smooth Skinning

Xiaosong Yang Jian J. Zhang
National Centre for Computer Animation, Bournemouth Media School
Bournemouth University, Unite Kingdom
{xyang, jzhang}@bournemouth.ac.uk

Abstract

Smooth Skinning is still the most popular method for the animation of deformable human and creature characters. It has became almost an industry standard because of its intuitiveness to use and efficiency to compute. However it suffers from a number of problems, such as the collapsing elbow and candy-wrapper effect. In this paper, we present a new method, which is able to solve these defects with very little extra stretching computations. The advantage of this method is its compatibility with the current animation workflow. The animator can change the skin weight any way they like and therefore it favours creativity. The aim of our method is trying to retain the realism and computational efficiency at the same time. It is suitable for situations, where real time response is necessary, like computer games. A practical implementation in the form of a Maya plug-in is created to demonstrate the viability of the method.

1. Introduction

Convincing skin deformation is a very important issue for realistic character animation. It has received a great deal of attention from the animation research community over the last two decades. But this research area is still one of the most difficult and challenging because animated human or creature characters are very complex objects, and we are very familiar with the subject in our life. Our eyes are very sensitive to even very little unnatural artifacts. Currently there are two prevalent approaches, one is based on the anatomy structures of the human/creature characters like muscles and fat, and the other is based on the direct deformation of the character skin.

The anatomy based technique tries to simulate the movement of the underlying organic tissues in a human or creature body. Normally three layers of anatomical structures are used, the skeleton, the musculature and possibly fat, and the skin layer [1]. This approach usually works by layering individual muscles on the skeleton. These muscles stretch or bulge following the motion of the skeleton. The topmost layer, the skin, takes the overall shape of the muscle and fat layer of the animated character body. Because the final shape of the skin is derived from the underlying structures, this approach affords good graphical realism. But because it is tedious and unintuitive to use, not many people in the animation industry employ this approach due to these drawbacks.

The second approach, often known as the *sub-space deformation (SSD) or smooth skinning*, makes no direct use of the anatomical elements, such as muscles and fat. The skin shape is controlled by the transformations of associated joints of an articulated skeleton. Its main advantage is that it affords real time deformation even on very low-end hardware. This technique is simple to use and leaves enough creative space for animators to get the desired result. Due to these advantages, it became the most popular method in animation production and has been incorporated into many animation packages. But because it uses a very simple linear shape blending technique to simulate an inherently highly non-linear relationship between skeleton and skin, it is understandable that unusual deformation artifacts appear in the skin during animation. Some of the commonly seen problems in smooth skinning during deformation are the so called collapsing elbow and candy-wrapper when the twist angle of a joint is close to 180 degrees. Insertion of extra joints is one good choice currently adopted by many animators. Although it improves the result, it doesn't really solve the problem. It needs the reconstruction of other structures in the animation and will take a lot of effort to produce a good outcome.

In this paper we introduce a novel method called the stretch smooth skinning to overcome the persisting drawbacks of joint skeleton skinning. Our method builds upon the same animation pipeline as the traditional smooth skinning. It doesn't involve any changes to the whole structure of the animation scene, not even the painted skin weight, which normally takes the animator a long time to get the desired result. It only adds a little extra stretch operation before and just after the smooth skinning operation. The artifacts including collapsing elbow and candy-wrapper can be completely removed.

1.1. Previous work

Skin deformation is a popular method of performing real time characters deformation by way of associated bones/joints of an articulated skeleton. According to the involvement of tissues other than the skeleton in the computation, two dominating techniques exist as stated above, one is called *anatomy based skinning* and the other is *skeleton driven skinning*.

For anatomy-based method, normally some physical properties [2] are considered in the simulation of individual anatomical elements, such as muscles and other internal elements. When the skeleton moves, these anatomical elements deform accordingly resulting in realistic deformation of the skin surface. A lot of research work including Wilhelms [3], Scheepers et al. [4], and Nedel [5] has been undertaken in this direction. Normally this technique is able to achieve very good graphical realism on the skin surface. However, it has not proven popular in animation production, due to the above mentioned disadvantages.

The skeleton driven method, which involves no knowledge of anatomy, goes by various names, such as s*keleton subspace deformation* by Lewis [6] and *smooth skinning* by some software packages (e.g. Maya). Here in this paper we will also adopt this name. An introduction of the background of this technique can be found in Lewis et al [6]. Catmull [7] introduced the first skeleton-driven technique. And then Magnenat-Thalmann et al [8] presented another 3D skeleton-driven technique that deforms character meshes using the motion of particular joints. Later research works have been aiming to both improve realism and reduce tedious manual intervention. One direction is called example based skinning. It uses a large number of examples, which may be obtained either by manual modelling or by range scanning. Wang and Phillips [9] proposed the least squares multi-weight technique to compute the weights of the elements of the transformation matrices from examples. Allen et al [10] presents a method to interpolate the scanned key shapes to derive a fuller set of shapes. The example-based approaches have the advantage over anatomical approaches by being computationally faster and having easy control over the result at specific poses. This technique is able to produce very realistic results, but all rely on the availability of a large number of pre-obtained models. This represents a major disadvantage, as this in itself is an expensive and time-consuming process.

Skeleton and skin relationship in the present smooth skinning pipeline is strictly linear, whereas observation of the various geometry artifacts like *candy-wrapper* and *collapsing joints* intuitively point to the fact that linear blending falls short in accurately depicting skin deformations because of their non-linear nature. Kavan[11] explored this non-linear nature and proposed a spherical blending method. The translation and rotation factor are treated differently in the computation of the skin vertices. Some animators try to add additional joints [12] near the main joint, to get rid of the *collapsing joints* problem. Although it weakens the artifact, it doesn't really solve the problem. And the "improved" result demands much extra efforts. The skeleton structure has to be modified by the insertion of a new joint, and the skin weight has to be redistributed to accommodate the new joint. And if the skeleton is binding to some motions or dynamic motion computations, these have to be reconstructed accordingly. Yang and Zhang [13] presented a geometrical method to solve the problems. Although the method gives convincing result, the additional computation load was high, and the weight is fixed in order to get the realistic result leaving little freedom for the animator, for example, to create the effect of muscle bulge, or other detailed skin features. And the biggest restriction of the method is that for each skin vertex only two influencing joints can be considered, which is very restricting. Most animation software packages, such as Maya, 4-5 influencing joints are used for smooth skinning.

1.2. Overview

Our method presented in this paper builds upon the same structure of smooth skinning, without involving extra joints, without restrictions on weight painting, without limitations of maximum influencing joints numbers. We attain the realistic skin deformation with only very little extra computations.

The aim of this paper is to retain the strength of the popular smooth skinning technique and keep the current production pattern unchanged, in the meanwhile allowing the graphical realism to be significantly improved. Our method is based on skin weighting, thus the animator can continue with their pattern of practice which they are used to for a long time. Similar to [13], our method deforms the skin vertex in a nonlinear manner. Current techniques either require heavy manual input or blend shapes in a linear manner, which is the major cause for many unpleasant defects.

The remainder of the paper is structured as follows. In Section 2 we present our stretching smooth skinning method. We will first introduce briefly the traditional technique. Then we discuss how it is improved to produce visually pleasant results. Section 3 shows our implementation in Maya and some result of skin deformation.

2. Stretch smooth skinning

As mentioned above, the defects of the smooth skinning technique is to some extend caused by the linear interpolation mechanism involved. In this Section, we present a new technique to remedy this problem, stretch out the skin mesh against the collapsing effect like in Figure 1. The key problems here are to decide how much the stretch operation will affect each skin vertex, and how the dimension of the stretch operation will be smoothed out around the collapsed joint. In order to leave the weights out of the stretch operation, i.e. no matter how the animator paints the weight, the stretch operation is the same. In other words, the weight will not be used in any other means in the entire process other than the traditional way of interpolation in smooth skinning. So theoretically the stretch operation will only be related to the transformation of the collapsing joint, not with the exact pattern of weighting and the skin local features. In the following subsections, we will describe how to stretch out from the collapsing joint and the candy-wrapper.

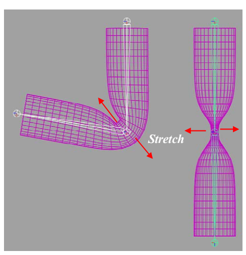


Figure 1. Stretching the collapsing effect

2.1. Smooth Skinning

Traditional smooth skinning method works by binding a skeleton with a character skin model, usually in a neutral pose. Each skin vertex is then assigned a set of influencing joints together with a weight factor corresponding to each influencing joint. This process is called skinning which establish the relationship between a skeleton and a skin mesh. Deforming the character into a different pose involves blending of the transformed vertex from the initial pose by all the influencing joints. At a skeletal pose c, a deformed vertex Vc, can be computed by [12]:

$$V_c = \sum_{i=1}^{n} w_i M_{i,c} M_{i,d}^{-1} V_d \qquad (1)$$

where V_d is the location of the vertex at its initial pose, w_i are the weights, $M_{i;c}$ denotes the transformation matrix associated with the ith joint in pose c and $M_{i;d}^{-1}$ the inverse of the transformation matrix associated with the ith influencing joint at the binding pose.

This smooth skinning method is very fast and widely adopted by animation software packages. However as mentioned above, it suffers from a number of defects such as collapsing elbow and candy-wrapping effect.

2.2. Stretch skin deformation

The most important step in smooth skinning is the weight distribution. Most animation software packages, e.g. Maya, leaves this work to the animator who paints the weights by hand. Understandably, assigning weights for all the skin vertices totally by hand is very time-consuming. Some research work was made to relieve the animator from the drudgery, such as [9] and [12] by trying to reverse-engineer the weights from a well defined pose space. However, problems remain if such a pose space, usually constructed from a large number of existing example models at different poses, is not available.

In [13], the author defines a separate set of weights for each skin vertex. The skin deformation is restricted to this weighting pattern. The animator is not allowed to change it.

In this paper, we present another method for direct assignment of weights to the vertices according to its position around the joint. One thing needed to be mentioned here is, this weighting is used only as a default assignment, which can be further changed as the animator wishes either by weight painting or other means, so retaining the creativeness for the animator.

2.2.1. Weight computation. Before we give the weight for each skin vertex, let's define a distance D between the skin vertex and joint.

Let's first consider the distance between a skin vertex and a bone (joint link). If the mapping of the vertex on the link line falls within the length of the bone, the distance will be the distance from the vertex to the bone; otherwise the distance will be the distance from the vertex to the nearest bone endings (Figure 2).

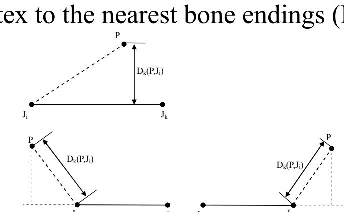


Figure 2. Distance between skin vertex *P* and Joint J_i

Assume P is the skin vertex, joint J_i is one of the corresponded joint, joint J_k is a child joint of J_i. (J_i, J_k) is a bone in the skeleton structure. So the distance $D_k(P, J_i)$ is defined as:

$$\alpha = \frac{\overrightarrow{J_iP} \bullet \overrightarrow{J_iJ_k}}{\left|\overrightarrow{J_iP}\right|^2} \quad (2)$$

$$d = \frac{\left|\overrightarrow{J_iP} \times \overrightarrow{J_iJ_k}\right|}{\left|\overrightarrow{J_iJ_k}\right|}$$

$$D_k(P, J_i) = \begin{cases} \left|\overrightarrow{J_iP}\right| & \alpha < 0 \\ \left|\overrightarrow{J_kP}\right| & \alpha > 1 \\ d & 0 \le \alpha \le 1 \end{cases}$$

This distance between P and J_i associated with the bone (J_i, J_k) is actually the distance from P to the line segment of (J_i, J_k).

In a normal character skeleton structure, some joints have multiple joint children. In this case, the final distance between P and J_i should be given as:

$$D(P, J_i) = \min(D_k(P, J_i), J_k \in Child(J_i))$$

For the weight of skin vertex P associated with the joint J_i is defined as

$$W_i(P) = \frac{\prod_{j \ne i, j=0}^{n} D\ (P, J_j)^m}{\sum_{k=1}^{n} \prod_{j \ne k, j=0}^{n} D\ (P, J_j)^m} \quad (3)$$

where n is the maximum number of influencing joints for each skin vertex. Normally it takes the value of *4-5*, m is usually called the drop rate. J_j is the joint influencing the deformation of P. The weight here is calculated by the Shepard's method [14] to be in inverse proportion to the m order power of the distance between the skin vertex and influencing joints. According to Shepard's method, the weight defined above has following qualities:

(4)

1. $W_i(P) \ge 0$
2. $W_i(P) \in C^0$
3. $\sum W_i(P) = 1$

Which just meet the requirement of the weighting operation.

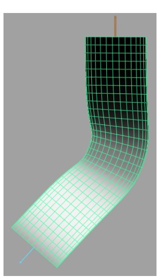


Figure 3**. Weight Definition**

Figure 3 shows the weight distribution for the second joint on a cylinder mesh, which is bound to a three joints skeleton.

2.2.2. Compensation for a collapsing joint. The problem of [13] is that it only considers the joint structure at limbs; only two joints can be considered at the same time. And the weight painting operation will affect the skin deformation unexpectedly. Actually in our method, the animator can modify the weight as they wish. What should be noted is that the weighting we give only serves as a default weighting. The following operations are not restricted by any specific weighting patterns defined so far.

The idea of the stretch operation is pretty simple. After the blending of the transformations, we add an extra operation to stretch the skin vertex out from the collapsing joint center to compensate for the collapsing result.

The following equation defines the stretch action.

$$P' = \frac{\overrightarrow{J_iP}}{\left|\overrightarrow{J_iP}\right|} * l' \quad (5)$$

where

$$l_2 = |\overrightarrow{PJ}|$$
$$\beta = 1 - 4 * (w_i - 0.5)^2 \quad (6)$$
$$l_1 = \sum_{i=0}^{n} w_i \left|\overrightarrow{JP_i}\right|$$
$$l' = l_1 * \beta + l_2 * (1 - \beta)$$

The stretch operation will be undertaken by the animator who chooses a specific collapsing joint. Assume J_i is a chosen joint, P is the skin vertex position after traditional smooth skinning. βcontrols the smoothness of the stretch operation on the skin vertices around the collapsing joint, l' is the new distance from the collapsing joint to compensate the loss of the volume, i.e the dimension of the stretch operation. In equation 6, l_1 is the weighted distance from the joint centre. At the place far away from the collapsed joint, the distance l_2 takes the dominant position, where getting closer to the collapsed joint, l_1 gradually take the control.

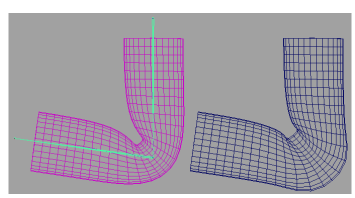


Figure 4 **Stretching out the skin mesh at the collapsing joint, left shows the traditional smooth skinning result. Right figure shows stretch result.**

2.2.3. Compensation for candy-wrapper effect. Candy-wrapper is a notorious problem for the smooth skinning method. The key reason is that the twist angle is computed together with all the other transformation operations. We will separate this part of computation out by adding an extra step before the real smooth skinning computation.

Firstly let us get the information of twist angle of each joint, such as α_i for joint *i*,

$$\alpha_i = \alpha_i^c - \alpha_i^b \quad (7)$$

where α_i^c is the current twist angle of the joint, α_i^b is the twist angle at the binding pose. Normally in software like Maya the twist angle is the rotation around the local *x*-axis. Then the final twist angle for the skin vertex *P* will be computed with the same weighting parameters:

$$\alpha' = \sum_{i=0}^{n} w_i \alpha_i \quad (8)$$

So before transforming skin vertex *P* from local space of joint *i* to the world coordinate system, we add a local rotation transformation around *x* axis with the angle of

$$\Delta\alpha_i = \alpha' - \alpha_i^b \quad (9)$$

So the smooth skinning equation will became

$$V_c = \sum_{i=1}^{n} w_i M_{i,c} M_{i,r} M_{i,d}^{-1} V_d \quad (10)$$

where

$$M_{i,r} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\Delta\alpha_i) & -\sin(\Delta\alpha_i) & 0 \\ 0 & \sin(\Delta\alpha_i) & \cos(\Delta\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (11)$$

This added rotation guarantees that after the transform of each influencing joint, the skin vertex has been set at the exact final twist angle position. Thus at the final smooth skinning blending step, no twist will be involved in the computation. The candy-wrapper effect will no longer occur. Figure 5 shows the compensation for the candy-wrapper on a cylinder with three joints binding inside, and the middle joint is given a twist rotation with the angle of *180* degrees.

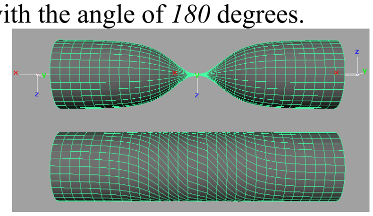


Figure 5. Twist result compared with smooth skinning, top figure shows candy-wrapper effect by using smooth skinning, the bottom shows our stretching result

3. Results

We have implemented our method into a plug-in for the Maya Environment alongside the traditional smooth skinning technique. A Stretch node was developed and inserted into the animation pipeline. It takes both the weighting and joint transformation information from the skinCluster node as the input connection, and also gets the mesh data at the binding pose. All the computation was resided in the Stretch node. It gives out the new skin mesh instead of the one supplied by skinCluster. Figure 6 shows the pipeline in Maya's hypergraph.

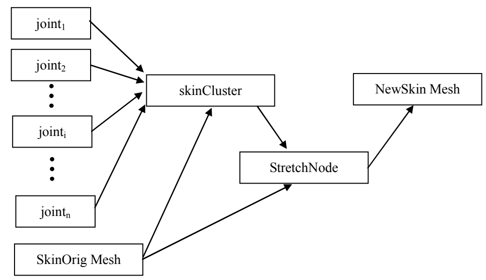
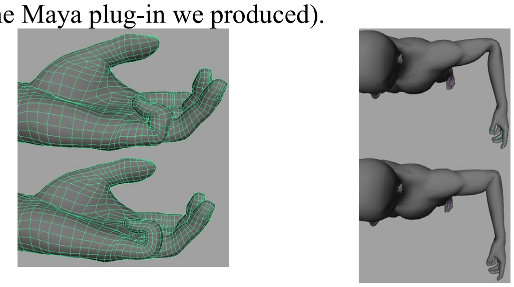


Figure 6. Flowchart for the Maya implementation

Compared with smooth skinning, we supply realistic result with very little extra computation. Before smooth skinning, a local rotation involving four multiplications and two additions are added for the joint and skin vertex involving twist operations. For each skin vertex related to the selected collapsing joint, after smooth skinning, the extra computation will be *11+6n* additions, *14+4n* multiplications, *n+1* square root computations, where *n* is the maximum influences number of joint on each skin vertex. Normally *n* takes the value of *4* or *5*. If the algorithm is implemented by using GPU programming language such as Cg, most of the vector related computations can be easily accelerated.

With a little more computation overhead, we solve the notorious collapsing joint and candy-wrapper problems for the smooth skinning method. Following are some examples using the new method (produced by the Maya plug-in we produced).



(a) (b)

Figure 7. a) Finger bending example, the top figure shows our result, the bottom figure shows the smooth skinning result; b) Forearm bending example, the top figure shows the collapsing joint from smooth skinning, the bottom figure shows stretch result

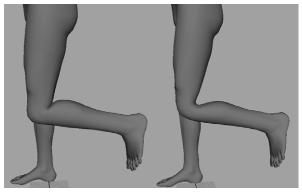


Figure 8. Leg bending example, the left figure shows our result, the right figure shows the collapsing result

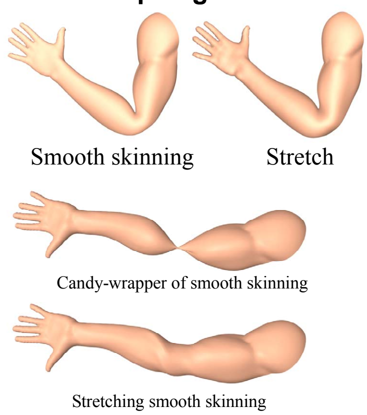


Figure 9. Forearm bending and twisting example on a more muscular model

4. Conclusion

We have presented a new stretch smooth skinning method for character skin deformation. Current linear blending techniques suffer from a number of defects, such as the collapsing elbow, candy-wrapper effects. In this paper, we studied the cause of such problems and presented an effective remedy. The idea is to identify and stretch the skin around a possible defected joint. Using this technique, the animator is able to produce realistic skin deformation easily and still be able to deliver his/her creative intent. Because our method is compatible with the traditional smooth skinning operation structure, it can be easily implemented into computer games and even GPU- based real time systems.

Acknowledgements

We would like to thank Arun Somasekharan for his valuable suggestions. This research is funded by the British Arts and Humanities Research Council grant B/RG/AN5263 /APN12727. We are grateful to Autodesk for their donation of the Maya software licences.

References

[1] Scheepers, F., R. E. Parent, et al. (1997). Anatomy-based modeling of the human musculature. Proceedings of the 24th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co.: 163--172.

[2] Chen, D. T. and D. Zeltzer (1992). Pump it up: computer animation of a biomechanically based model of muscle using the finite element method. Proceedings of the 19th annual conference on Computer graphics and interactive techniques, ACM Press: 89--98.

[3] Wilhelms, J. and A.V. Gelder, Anatomically based modeling, in Proceedings of the 24th annual conference on Computer graphics and interactive techniques. 1997, ACM Press/Addison-Wesley Publishing Co. p. 173—180

[4] Scheepers, F., R. E. Parent, et al. (1997). Anatomy-based modeling of the human musculature. Proceedings of the 24th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co.: 163--172.

[5] Nedel, L. and D. Thalmann, Modeling and Deformation of the Human Body using an Anatomically-Based Approach, in Proceedings of the Computer Animation. 1998, IEEE Computer Society. p. 34.

[6] Lewis, J. P., M. Cordner, et al. (2000). Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. Proceedings of the 27th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co.: 165—172

[7] Catmull, E. E. (1972). A system for computer generated movies. Proc. ACM Annual Conf.

[8] Magnenat Thalmann, M., Laperrire, R., And Thalmann, D. (1988). Joint dependent loal deformations for hand animation and object grasping. Proceedings of Graphics Interface'88, 26-33.

[9] Wang, X. C. and C. Phillips (2002). Multi-weight enveloping: least-squares approximation techniques for skin animation. Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, ACM Press: 129--138

[10] Allen, B., B. Curless, et al. (2002). Articulated body deformation from range scan data. Proceedings of the 29th annual conference on Computer graphics and interactive techniques, ACM Press: 612--619.

[11] Kavan L., Zara J.: Spherical Blend Skinning: A Real-time deformation of Articulated models. In Proceedings of the 2005 symposium on Interactive 3D graphics and games (2005), pp. 9-16.

[12] Mohr A., Gleicher M.: Building efficient, accurate character skins from examples. ACM Transactions on Graphics. Vol. 22, Issue 3 (2003), pp. 562-568.

[13] Yang X., Zhang J J.: Realistic skeleton driven skin deformation. In Third Technical Session on Computer Graphics and Geometric Modeling, TSCG'2005 Suntec Singapore, Singapore: Springer-Verlag (2005).

[14] Shepard D. A Two Dimensional Interpolation Function for Irregularly Spaced Data. Proceedings of ACM 23rd national Conference, 517-524