Dynamic Skinning for Popping Dance

HongJung Son and JungHyun Han[**]

Game Research Center, College of Information and Communications,
Korea University, Seoul, Korea
`hj79337@gmail.com`, `jhan@korea.ac.kr`

Abstract. This paper presents an efficient technique to enhance the realism of character animation by adding muscle dynamics. Focusing on the isometric contraction of muscles, the proposed algorithm takes normal mesh and clenched mesh, and uses the disparity between them to simulate the skin vibration. The skin simulation algorithm is integrated with an example-based skinning, and shows real-time performance. The proposed approach proves to be useful for animating popping dance.

Key words: character animation, skinning, mesh deformation.

1 Introduction

Skin movement plays a key role in computer animation. The dominant approach to skin animation has been Linear Blend Skinning(LBS)[5]. The LBS algorithm is fast enough to be processed in real-time, is widely used among computer artists, and is also widely supported by commercial applications. However, binding the skin to the bones restricts dynamic effects caused by muscle movement, which is important to make realistic animation. Our goal is to simulate the dynamic skin caused by muscle movement.

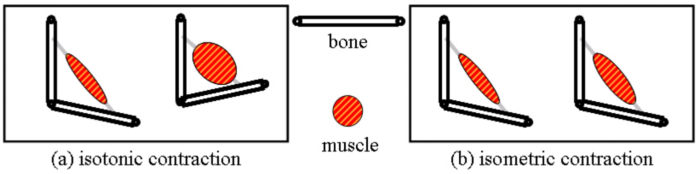


Fig. 1. Muscle contraction

The factors that bring changes to skin include bones, muscles, tendon, fat, etc. With respect to muscles, anatomists distinguish between two types of muscle contraction[1]: *isotonic* and *isometric*, as shown in Fig. 1. Upon isotonic contraction, the belly changes shape, often bulging, while the total length of the muscle

---

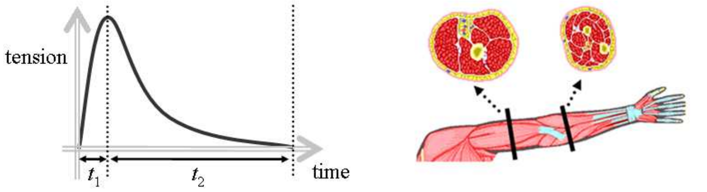
[**] Corresponding Author.

diminishes so that the bones to which the muscle is attached are pulled towards each other. Upon isometric contraction, the shape of the belly also alters because of the tension in the muscle but the length of the muscle does not change, and therefore no skeletal motion is produced.

The dynamic skin effect caused by the isometric contraction of muscles can be easily found in *popping dance*, where one pumps up the muscles on purpose to bulge muscles. This paper focuses on the isometric contraction to add dynamic effect to the skin, and presents an algorithm for simulating popping dance.

2 Related Work and Background

The skinning algorithm is notorious for its failings including *collapsing joint* and *candy-wrapper* problems. A representative effort to resolve the problems is *example-based skinning*, which uses a set of examples typically made by designers[4, 5, 7, 11]. There have been attempts to add dynamic effect to the character skin since Chadwick *et al.*[2]. Turner and Thalmann[10] and Wilhelms and Gelder[12] used an elastic surface to represent a deformable skin layer. James and Pai[3] pre-computed modal analysis of dynamic elastic models and combined vibration modes in real-time using graphics hardware. While this technique gives a physical realism for animation, it requires the animator to have advanced physics knowledge to set up the finite element simulation and to provide a volumetric mesh for the model.



(a) muscle twitch (b) cross sections of human arm (from [6])

Fig. 2. Properties of human muscle

Fig. 3. Skin vibration examples from video

Fig. 2-(a) illustrates a muscle's twitch upon isometric contraction[9]. The time-to-peak denoted by t_1 leads to instant *bulging* of muscle. Fig. 2-(b) shows

cross sections of human arm, where a complex structure of the muscles can be found. Such a complex structure leads to *skin vibration* when muscles bulge altogether. Skin vibration can be easily found in popping dance. Fig. 3 shows snapshots of skin vibration for a fixed skeletal configuration.

3 Skin Vibration and its Integration with Skinning

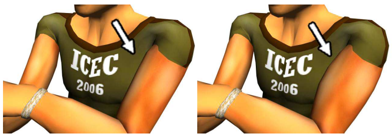


Fig. 4. Normal and clenched meshes

To capture the popping effect, two meshes are taken as input from the designer, as shown in Fig. 4. One is *normal mesh* of a character, and the other is the mesh with isometric contraction, which we call *clenched mesh*. When one pumps up the muscles, all vertices of the normal mesh are supposed to move to the surface of the clenched mesh, i.e. the *goal position* of a vertex is set to the corresponding vertex position in the clenched mesh.

For simulating the skin vibration, a modified version of the integration scheme proposed by [8], which guarantees unconditionally stable simulation, is used to calculate the velocity(v) and position(x) of a vertex i with time t and time step h:

$$v_i(t+h) = (1-\alpha)v_i(t) + \alpha\frac{g_i(t) - x_i(t)}{h} \tag{1}$$

$$x_i(t+h) = x_i(t) + hv_i(t+h) \tag{2}$$

where α=(0...1] simulates the *stiffness* of a skin vertex, and g denotes the goal position toward which the vertex is pulled. Given two vertices, $x(t)$ and $g(t)$, the iterative scheme yields the following update rule:

$$\begin{bmatrix} v(t+h) \\ x(t+h) \end{bmatrix} = \begin{bmatrix} 1-\alpha & -\alpha/h \\ (1-\alpha)h & 1-\alpha \end{bmatrix} \begin{bmatrix} v(t) \\ x(t) \end{bmatrix} + \begin{bmatrix} \alpha g(t)/h \\ \alpha g(t) \end{bmatrix}$$

The first term on the right hand side is the system matrix. Its eigenvalues are $1-\alpha \pm i\sqrt{\alpha(1-\alpha)}$, and the magnitude of both eigenvalues is $1-\alpha$. If α=1, the vertex instantly adheres to the goal position, i.e. there is no vibration. If $\alpha < 1$, the vertex sequentially translates towards the goal position. With accumulated translations, the vertex may pass by the goal position. Then, it retreats. When such a back-and-forth move is repeated, the vibration effect is achieved near the goal position.

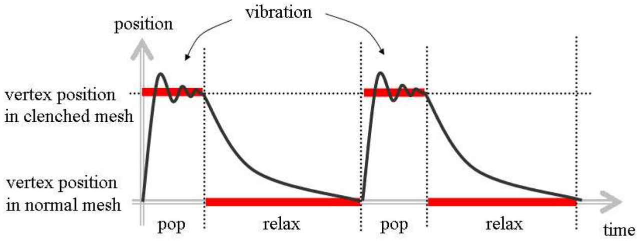
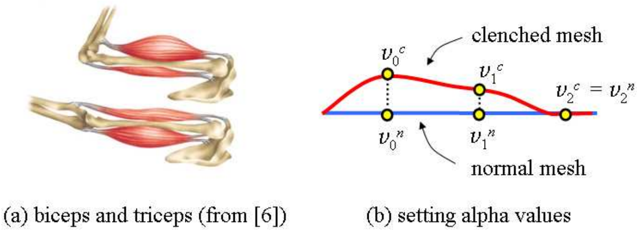


Fig. 5. Skin popping

Fig. 5 visualizes the skin vibration or popping effect. The horizontal bold line segments marked by 'vertex position in clenched mesh' represent the goal positions, and the curved graph traces the vertex movement.



(a) biceps and triceps (from [6]) (b) setting alpha values

Fig. 6. Setting stiffness values

Most of skeletal muscles in the limbs are long *fusiform* muscles[1]. The biceps and triceps in Fig. 6-(a) are good examples. When the muscles are pumped up, the central parts experience more vibration effect than the parts near the tendon. It can be simulated by assigning different stiffness values to vertices. Given two example meshes (normal and clenched meshes), the distances are computed for all pairs of corresponding vertices. Suppose that the maximum distance is $dist_{max}$. For the i-th vertex in the normal mesh with distance $dist_i$, its stiffness α_i is computed as follows:

$$\alpha_i = (\alpha_{min} - 1) \times \frac{dist_i}{dist_{max}} + 1 \tag{3}$$

where α_{min} is the minimum stiffness (between 0 and 1) set by the animator. Fig. 6-(b) shows examples. For vertex v_0, $dist_0$=$dist_{max}$ and therefore α_0=α_{min}. Vertex v_0 is assigned the minimum stiffness, and experiences the most vibration. For all other vertices, the stiffness values are greater than α_{min}, which results in faster convergence. The extreme case is found in v_2, where $dist_2$=0 and α_2=1.

Note that, when one relaxes the muscles, the normal mesh is used to set up the goal positions. As visualized in Fig. 5, the skin surface smoothly moves from the clenched mesh to the normal mesh without vibration. Such a monotonically decreasing curve can be simulated as follows:

$$x_i(t+h) = x_i(t) + \beta(g_i(t) - x_i(t))h \tag{4}$$

where β is a scalar value between 0 and 1. The bigger β is, the more sharply the curve decreases. When the distance to the goal position becomes less than a threshold, the vertex is clamped to the goal position of the normal mesh.

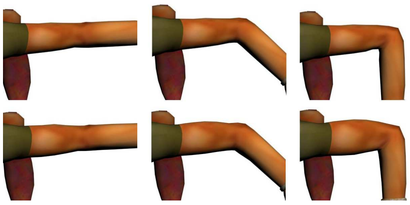


Fig. 7. Example meshes

The proposed algorithm works as a secondary animation for LBS, and is added to the result of LBS. However, they cannot be separated because the translation vector for popping effect should be determined considering the bone configuration in LBS especially for the joint part of a character. Example-based skinning algorithm can resolve the problem. As shown in Fig. 7, three pairs of example meshes are used. A pair is per a distinct bone configuration. The normal meshes are illustrated in the upper row, and the clenched meshes are in the lower. The sparse data interpolation technique[5] is applied both to the normal mesh group and to the clenched mesh group, and the results are fed into the proposed dynamic skin simulation algorithm.

4 Implementation

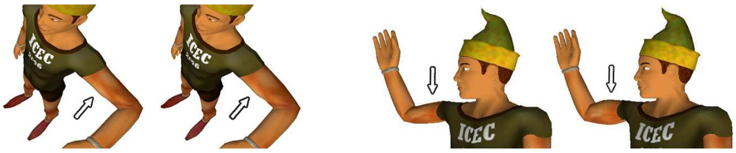


Fig. 8. Results

The proposed approach has been implemented in 3.6GHz Intel CPU and 2GB RAM. Fig. 8 compares the skinning results with and without dynamic skin effect. The character has 4,380 vertices, and is animated at 60 FPS on average when integrated with the sparse data interpolation technique for skinning.

5 Conclusion

This paper presents an efficient technique to enhance the realism of character animation by adding muscle dynamics. Aiming at real-time animation of popping dance, the proposed algorithm takes normal mesh and clenched mesh, and uses the disparity between them to simulate the skin vibration. The skin simulation algorithm is neatly integrated with an example-based skinning. The experimental results show the feasibility of animating the dynamic skin effect in real-time.

Acknowledgements

This research was supported by the Ministry of Information and Communication, Korea under the Information Technology Research Center support program supervised by the Institute of Information Technology Assessment, IITA-2005(C1090-0501-0019).

References

1. Aubel, A.: Anatomically-based human body deformations. Ph.D Thesis No. 2573, EPFL. (2002)
2. Chadwick, J. E., Haumann, D. R., Parent, R. E.: Layered construction for deformable animated characters. ACM Computer Graphics. **23** (1989) 243–252
3. James, D. L., Pai, D. K.: DyRT: Dynamic response textures for real time deformation simulation with graphics hardware. ACM Transactions on Graphics. **21** (2002) 582–585
4. Kry, P. G., James, D. L., Pai, D. K.:Eigenskin: Real time large deformation character skinning in hardware. In ACM SIGGRAPH Symposium on Computer Animation. (2002) 153–160
5. Lewis, J. P., Cordner, M., Fong, N.: Pose space deformations: A unified approach to shape interpoalation and skeleton-driven deformation. In Proceedings of ACM SIGGRAPH. (2000) 165–172
6. Macintosh, B. R., Gardiner, P. F., McComas, A. J.: Skeletal Muscle - Form and function. Human Kinetics Press. (2005)
7. Mohr, A., Gleicher, M.: Building Efficient, Accruate Character Skins from Examples. ACM Transactions on Graphics. **22** (2003) 562–568
8. Muller, M., Heidelberger, B., Teschner, M. Gross, M.: Meshless Deformations Based on Shape Matching. ACM Transactions on Graphics. **24** (2005) 471–478
9. Phillips, C. A., Repperger, D. W., Neidhard-Doll, A. T., Reynolds, D. B., Biomimetic model of skeletal muscle isometric contraction. I. An energetic viscoelastic model for the skeletal muscle isometric force twitch. Computers in Biology and Medicine. **34** (2004) 307–322
10. Turner, R., Thalmann, D.: The elastic surface layer model for animated character construction. In Proceedings of Computer Graphics International. (1993) 399–412
11. Wang, X., Phillips, C.: Multi-weight enveloping : Least-squares aproximation techniques for skin animation. ACM SIGGRAPH Symposium on Computer Animation. (2002) 129-138
12. Wlhelms, J., Gelder, A. V.: Anatomically based modeling. ACM Computer graphics. (1997) 173–180