A Control Cluster Approach to Non-linear Deformation

A. A. Bukatov
SFU Computer Center
Bolshaya Sadovaya 105, 344000,
Rostov-on-Don, Russia
baa@sfedu.ru

E. E. Gridchina
Informatics and Computational Experiment Department, SFU,
BolshayaSadovaya 105, 344000,
Rostov-on-Don, Russia
helen.gridchina@gmail.ru

D. A. Zastavnoy
Informatics and Computational Experiment Department, SFU,
BolshayaSadovaya 105, 344000,
Rostov-on-Don, Russia
dzast@sfedu.ru

ABSTRACT

Modeling plausible deformation of the objects has been an important task in computer animation and game design industry. The approach proposed in the paper deals with a polygonal mesh deformation splitting the vertices of the mesh into two types: cluster vertices and free vertices. With the user defining the shape of the mesh key areas with the help of cluster vertices, the algorithm takes advantage of non-linear geometric deformation for calculating free vertices position. The approach could be used both for creating a sequence of altered model shapes to produce a character animation (with the help of user-created control cluster data) and for visualizing some ecological processes.

Keywords

Control cluster, non-linear deformation, skeletal animation, polygonal mesh deformation.

1. INTRODUCTION

Deforming model meshes has become an active field of research [1-5]. It can be used both for modeling characters and for creating a sequence of altered model shapes to produce an animation.

Most of the algorithms can present either physical or geometric approach. We concentrate in the geometric approach domain. Most of the geometric algorithms proposed demonstrate shortcomings connected with their linear approach. So this research focuses on the non-linear geometric approach to deformation.

In this paper a geometric method for plausible polygonal model deformation is proposed. It makes use of control clusters to facilitate the work of the animator in creating realistic animation of the character without involving much user-input data. The method proposed provides intuitive control and it is easy to use because it allows the user to influence only a small group of vertices leaving the non-linear deformation of the rest of the vertices to the algorithm.

In the previous papers an approach based on a control clusters technique for correcting 2D models skinning deformation was presented [6]. This paper is to describe a generalized and more versatile approach to deforming 3D polygonal models – Control Cluster Method (CCM). The generalized approach can be implemented in a wide variety of applications including physical processes simulation, such as ecological processes (e.g., computational domain flooding and dewatering as a result of wind upsurge-downsurge; fire spreading; prevalence of air and water pollution).

2. RELATED WORK

Among the geometric deformation technologies connected with skeletal animation the most widely used is Linear Blend Skinning (LBS). Being simple and straightforward LBS is notorious for its well-studied shortcomings [7].

To avoid well-known artifacts Pose Space Deformation proposed in [8] takes advantage of using sample shapes of the model, the technique requiring much input from the user [9].

Dual-quaternion skinning is a more recent skeletal animation technique [10], describing both rotations and translations using quaternions. Having no shortcomings of LBS, dual-quaternion skinning demonstrates some new artifacts connected with too much volume.

It was also proposed in [11] to pre-compute optimized skinning weights for linear and dual-quaternion skinning at joints to approximate the skin transformations produced by nonlinear variational deformation methods.

[12] exploits the advanced compositions mechanisms of volumetric implicit representations for correcting the results of geometric skinning techniques.

3. PROPOSED APPROACH

3.1 General description

Control cluster method assumes all the vertices of the model are split into two types: cluster vertices and so-called free vertices. The position of the cluster vertices is defined by some input data (see 3.4). The key areas of the mesh defined by cluster vertices form the shape of the model, thus cluster vertices are used for controlling the deformation of the model. The position of free vertices is calculated automatically.

3.2 Cluster vertices transformation

Cluster vertices can be transformed in a great variety of ways, either by mesh editing or by applying some deformers including lattices, cage or skeleton. Cluster vertices transformation could be also defined by data from some type of measuring equipment, for example water level measuring device.

3.3 Free vertices transformation

If a vertex does not belong to the cluster, it is considered a free one and its position is recalculated based on its position in the model topology and its nearest cluster vertices position. Free vertices position can be defined in several non-linear ways, including cubic cardinal splines generalizing cubic Catmull-Rom splines [13] so that it corresponds to the position of the cluster vertices.

Using interpolation, C^1-continuity and local control, cardinal splines are an acceptable way to solve the task. The C^1-continuity of the spline provides a smooth natural look of the character, with the spline remaining flexible. Also the spline interpolates its control points giving direct control over the points of the curve. With local control the spline has every control vertex provide a slight impact on the overall look, so the model details are preserved. Therefore cardinal splines allow the achievement of realistic deformation of commonly difficult parts of the model.

The shape of the curve the spline gives depends heavily on the parameterization defined [14]. Choosing a spline parameterization for our method we considered three types: uniform parameterization, chord-length parameterization, centripetal parameterization.

Uniform parameterization is considered the most popular choice for cardinal splines, though for curves with segments of different length this parameterization often leads to artifacts such as self-intersections within short curve segments [14], which is as usual unacceptable. Cusps and intersections are also possible when using chord-length parameterization, the curve "overshoots" within longer curve segments. Centripetal parameterization is the only not to generate such artifacts. Moreover, among these parameterizations the centripetal version appears to produce a curve that is closer to the control polygon than the others.

Despite the fact that it is mathematically proven that centripetal parameterization lacks traditionally undesired features such as cusps and intersections within a segment [14], the CCM uses chord-length parameterization. Our reasoning behind this preference differs from the standard one as CCM can deal with character models. Firstly, the curvature of the relatively long curve segments is larger in comparison with the results of the other parameterizations considered, and it tends to remain small within shorter curve segments. It helps achieve a more realistic look of the character, as human-like models will not lose much volume when animated. Secondly, chord-length parameterization is considered the best as it provides a very well-conditioned linear system of equations compared to other parameterization types. So if $\boldsymbol{p}_i$, $\boldsymbol{p}_{i+1}$are spline control points the parameterization $t_{i+1} = t_i + |\boldsymbol{p}_{i+1} - \boldsymbol{p}_i|$ is used.

The cardinal spline curve segment shape depends on four neighboring data points: $\boldsymbol{p}_{i-2}$, $\boldsymbol{p}_{i-1}$, $\boldsymbol{p}_{i+1}$, $\boldsymbol{p}_{i+2}$. The tangent is calculated as

$$m_k = (1-q)\frac{\boldsymbol{p}_{k+1} - \boldsymbol{p}_{k-1}}{t_{k+1} - t_{k-1}} \qquad (1)$$

The tension parameter q defines the curvature of the spline, $q \in [-1,1]$. The method sets q=-0.5 as the default parameter value.

3.4 Fields of application

Control cluster method could be used in different spheres of visualization ranging from animation to ecological processes simulation.

The cluster vertices deformation can be set using user-input data, including different deformers such as a skeleton to produce a character animation or some certain pose of the model.

Provided cluster vertices positions are defined by some scientific equipment, a wind upsurge-downsurge flooding and dewatering simulation could be achieved with the help of control cluster method. Among possible spheres of control cluster method application in ecological processes simulation fire spreading and prevalence of air and water pollution can be mentioned.

In this paper application of the method to animation of human-like character models is presented.

 ISBN 978-80-86943-72-5

4. APPLICATION TO SKINNING

4.1 Skinning cluster vertices transformation

Skinning application of CCM can be used for creating character animation. Generally speaking, cluster vertices transformations can be defined in different ways that are not strictly set. To demonstrate it we consider three different ways for cluster vertices transformation in creating arm rotation animation: Linear Blend Skinning, Pose Space Deformation and procedural dependence.

The most efficient, versatile and wide-spread way of creating a character animation is Linear Blend Skinning. So base cluster vertices can be deformed with LBS.

Let $B=\{\boldsymbol{b_i}\}$ be a skeleton, i.e. a hierarchy of bones. Every bone $\boldsymbol{b_i}$ is assigned a coordinate system and a 3D transformation (translation, rotation and scaling), defined by matrix W_i. The transformation of a child node of the hierarchy inherits its parent node transformation. Every vertex $\boldsymbol{v} \in V_P$ is associated with a set of weights $\{w_i\}$, $\sum_i w_i = 1$, where w_i is the weight of the bone $\boldsymbol{b_i}$. Weight w_i defines the extent to which the vertex position is influenced by the bone $\boldsymbol{b_i}$. Let $\{B_i\}$ be the skeleton configuration in the bind pose. The skeleton being in an arbitrary pose $\{W_i\}$, the transformed position of cluster vertex $\boldsymbol{v'}$ is calculated according to the formulae

$$\boldsymbol{v'} = \mathrm{LBS}(\boldsymbol{v}) = \sum_i w_i W_i B_i^{-1} \boldsymbol{v} \qquad (2)$$

Being versatile and computationally efficient, LBS demonstrates some undesired artifacts, such as volume loss. One of the defects, so-called “collapsing elbow”, results in unnatural look of the character. To correct those undesired artifacts PSD can be used. It requires using sample pairs <X, S>, where X is a user-input sample shape of the model corresponding to skeleton configuration S. With the flexibility of control cluster method it is possible to avoid defining sample shapes of the whole model. Instead only the problem vertices of the elbow area of the model can be defined as cluster ones and only one sample pair is used for deforming those vertices likewise PSD thus achieving the necessary amount of volume. So cluster vertices of the elbow problem area undergo additional transformations with the help of the displacements

Let us assume that $\boldsymbol{v}$ is a vertex position in the sample pose of cluster vertices in the pose X^i, $\boldsymbol{v^0}$ is vertex position in the bind pose B. Then displacement $\boldsymbol{d_j}$ in the bone $\boldsymbol{b_j}$ local coordinate system is calculated as follows:

$$\boldsymbol{d_j} = LBS^{-1}(\boldsymbol{v}, \boldsymbol{b_j}) - B_j^{-1}\boldsymbol{v^0} \qquad (3)$$

If the current pose $X=X^i$, the current position of the control cluster vertex $\boldsymbol{v}$ is calculated as:

$$\boldsymbol{v'} = LBS(\boldsymbol{v}, \boldsymbol{d}) = \sum_j w_j W_j (B_j^{-1}\boldsymbol{v} + \boldsymbol{d_j}) \qquad (4)$$

Another welcome feature for realistic animation is muscle bulging. It could also be achieved using PSD but it would require for the user to create some sample pairs to imitate realistic muscle bulging. As an alternative way cluster vertices corresponding to biceps area of the upper arm can be deformed procedurally depending on the angle between upper arm and lower arm bones.

5. RESULTS

For the sake of example cylinder model (a simplified “hand” model) with rigging is used (see Fig. 1). All the vertices are split into 4 groups: 3 cluster groups and free vertices.

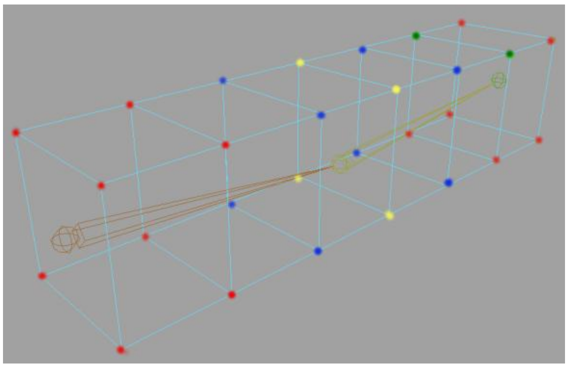

Figure 1. The model of a cylinder with a two bone skeleton. Red – cluster LBS deformed vertices; yellow – cluster likewise PSD deformed vertices; green – cluster procedurally deformed vertices; blue – free vertices.

The base cluster vertices of the model are deformed with help of LBS, they define the overall shape of the model. Cluster vertices of the “elbow” problem area of the model are transformed using one sample pair for the four cluster vertices, the sample pair being extracted from PSD sample shape of the whole model. To create biceps bulging effect corresponding cluster vertices position is calculated procedurally based on the angle between the two bones.

Comparing the three methods (LBS, PSD and CCM) in a rotation animation without muscle bulging it is possible to say that yielding close results neither PSD, nor CCM demonstrates the volume loss common for LBS (see Fig. 2). As far as user input data is concerned, PSD requires to create at least one 12 vertices sample pair for the animation, while it is enough to store one 4 vertices sample for CCM.

Moreover, with a greater angle of rotation due to using cardinal splines CCM forms a cusp in the area

 ISBN 978-80-86943-72-5

of the elbow that is natural for human like characters (see Fig. 3).

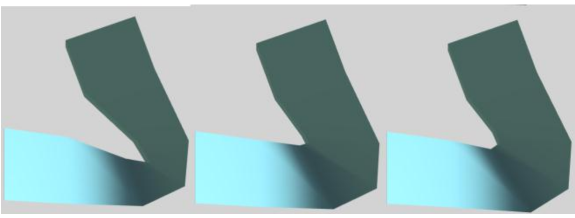

Figure 2. Side view of the model, frames of the child bone rotation animation. From left to right: LBS, PSD, CCM.

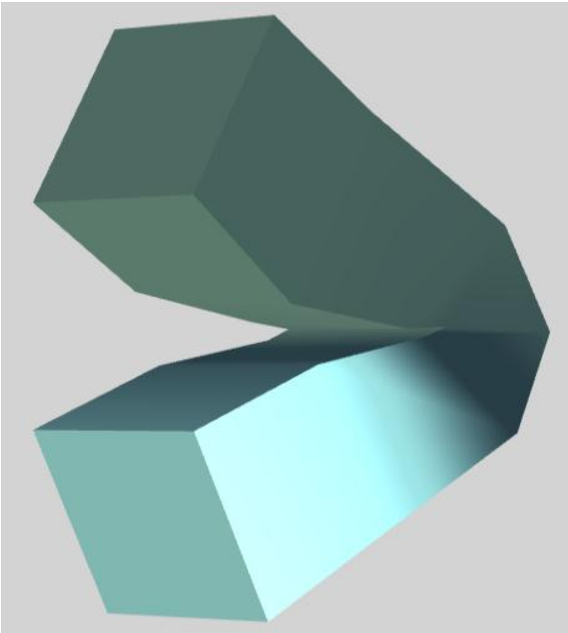

Figure 3. CCM rotation animation of the model with a slight muscle bulging effect.

6. CONCLUSION AND FUTURE WORK

Control cluster method is a novel approach for geometric non-linear deformation of the model in a labor-saving way. In this paper its application to skinning is described. Applications of the approach to other fields as physical processes simulation is in the scope of the future research.

7. REFERENCES

[1] Gain J., Bechmann D., A survey of spatial deformation from a user-centered perspective. ACM Trans. on Graphics, 27/4, 107:1–107:21 (2008).

[2] Cohen-Or D. Space deformations, surface deformations and the opportunities in-between. J. Comput. Sci. Tech., 2009,24: pp. 2–5.

[3] Botsch M, Sorkine O. On linear variational surface deformation methods. IEEE Trans. Vis. Comput. Graph., 2008, 14: pp. 213–230.

[4] Deformation Models: Tracking, Animation and Applications Series: Lecture Notes in Computational Vision and Biomechanics, Vol. 7, González Hidalgo, Manuel, Mir Torres, Arnau, Varona Gómez, Javier (Eds.), 2013, XIV, 297 p.

[5] Botsch M., Kobbelt L., Pauly M., Alliez P. and Levy B., Polygon Mesh Processing. AK Peters - 2010.

[6] Bukatov A.A., Gridchina Y.Y., Zastavnoy D.A. A Spline-Based Approach to Control Cluster Deformation, World Applied Sciences Journal, 26 (6): 724-727, 2013.

[7] A. A. Bukatov, E. E. Gridchina, D. A. Zastavnoy, 2012. Skeletal animation techniques for deforming polygonal surfaces of 3D models. Inzhenerniy Vestnik Dona, 3: 59-74 (in Russian).

[8] J. P. Lewis, M. Cordner and N. Fong, 2000. Pose space deformation: a unified approach to shape interpolation and skeleton driven deformation. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, pp: 165-172.

[9] G. S. Lee and F. Hanner, 2009. Practical experiences with pose space deformation. In ACM SIGGRAPH ASIA 2009 Sketches, pp: 43:1-43:1.

[10] Skinning with dual quaternions / L. Kavan, S. Collins, J. Zara, C. O'Sullivan // Proceedings of the 2007 symposium on Interactive 3D graphics and games.— I3D '07.— New York, NY, USA: ACM, 2007.— Pp. 39–46.

[11] Kavan, L. Elasticity-Inspired Deformers for Character Articulation / L. Kavan, O. Snorkie // ACM Trans. Graph. 31(6): 196 (2012).

[12] Implicit skinning: real-time skin deformation with contact modeling / Vaillant R., Barthe L., Guennebaud G., Cani M., Rohmer D., Wyvill B., Gourmel O. and Paulin M. // ACM Trans. Graph. 32, 4, Article 125 (July 2013), 12 pages.

[13] E. Catmull and R. Rom, 1974. A class of local interpolating splines. In Computer Aided Geometric Design, Eds., R. E. Barnhill and R. F. Reisenfeld. New York: Academic Press, pp: 317-326.

[14] C. Yuksel, S. Schaefer and J. Keyser, "On the parameterization of Catmull-Rom curves", ACM Joint Conference on Geometric and Physical Modeling (SPM '09). New York: ACM, 2009, pp. 47–53.