Medial-Based Vertex Deformation

Jules Bloomenthal
Unchained Geometry, Inc.

Symposium on Computer Animation
San Antonio, July, 2002

Abstract

Vertex deformation is a popular technique to animate an erstwhile static object. It is difficult, however, to deform those vertices near multiple limbs of the controlling stick-figure skeleton while maintaining a natural-appearing surface.

By applying convolution to the medial axis/surface of the object, the weights associated with vertex deformation can be computed automatically. Fewer undesired artifacts are evidenced in the animated surface.

1. Introduction

Vertex deformation is a method of character animation in which the vertices of an object's surface are moved interactively in response to some control mechanism. The number of vertices and their inter-connectivity do not change.

The control mechanism may be a surrounding lattice or an internal stick-figure 'skeleton'. We consider the skeleton approach, which is called *axial deformation* [Lazarus *et al*. 1994] or *skeleton subspace deformation* [Lewis *et al*. 2000].

The stick-figure skeleton consists of individual *limbs*; two or more limbs meet at an articulable *joint*. Either the skeleton is posed via end effectors, in which case inverse kinematics is used to compute the joint angles, or the joint angles are specified directly by the animator.

It is straightforward to produce a 'rigid-body' rotation of a vertex associated with a single limb. It is considerably more challenging to move a vertex situated near two or more limbs such that the animated surface is smooth and natural in appearance.

Typically, the vertex location is given as a *vertex blend* (*i.e.*, a weighted sum) [Akenine-Möller and Haines 2002]. Each term of the summation is computed using a reference frame associated with a corresponding, animated limb.

Vertex deformation appears increasingly popular, in part due to increased hardware support for vertex blends [Akenine-Möller and Haines 2002], [Lander 2001].

As described in [Lewis *et al*. 2000], the default coefficients of the weighted sum often require tedious, manual adjustment and the results often evidence numerous artifacts. Indeed, these difficulties are considered 'notorious'. They are also regarded as inevitable as a consideration of the common case involving two limbs straddling a joint reveals the 'collapsing joint defect' in which the surface is pinched around a joint that is bent or twisted [Lewis *et al*. 2000].

The situation improves when intermediate reference frames are introduced. For example, a 180° twist between successive frames yields a zero radius (see figure below). But with a single intermediate frame the radius shrinks less than 30% and with three intermediate frames (*i.e.*, a 45° rotation between frames), the shrinkage is less than 8%.

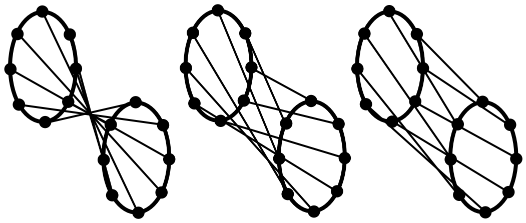

Figure 1. Pinch due to twist between frames (left to right: 180°, 90°, and 45°).

This paper addresses the collapsing joint defect with a method that, unlike conventional 'deformers', computes multiple frames (and associated weights) for each limb influencing an object vertex. The technique reduces artifacts such as pinching and bulging, and obviates the manual adjustment typical of vertex deformation.

The following section provides background on vertex deformation. Section 3 describes our method, which involves convolution of the medial axis/surface of the object. Section 4 presents details concerning the medial and section 5 provides details concerning convolution. Section 6 demonstrates results, and section 7 summarizes the paper and considers future work.

2. Vertex Deformation

This section summarizes the basic methodology of vertex deformation (an early description is given in [Magnenat-Thalmann *et al.* 1988, sec. 5]).

A reference frame (*i.e.*, coordinate system) F is established for each limb of the stick-figure skeleton. The frame consists of a location F_p and three orthonormal axes, F_t, F_n, and F_b (*i.e.*, the tangent, normal and bi-normal), with F_t typically directed from base towards tip of the limb. F_p may be positioned arbitrarily along the limb; in Figure 2 it is at the base.

In some systems [Alias/Wavefront 1999], the joints can be interpolated by curves, in which case the frame orientation changes along a limb and the at-rest frame F is typically positioned at $\boldsymbol{n}$, the point on the curve nearest to $\boldsymbol{q}$. F is augmented with t, the parametric location of $\boldsymbol{n}$.

With the object at rest (*i.e.*, no joint articulation), each surface vertex $\boldsymbol{q}$ is assigned a set of transformed $\boldsymbol{q_i'}$ that correspond with those *limb*$_i$ influencing $\boldsymbol{q}$. For a given limb L, $\boldsymbol{q'}$ is simply $\boldsymbol{q}$ with respect to the at-rest frame F. Referring to figure 2, $\boldsymbol{q'}$ is $(\boldsymbol{r{\bullet}t}, \boldsymbol{r{\bullet}n}, \boldsymbol{r{\bullet}b})$, *i.e.,* the projection of $\boldsymbol{r}$ onto the axes of frame F.

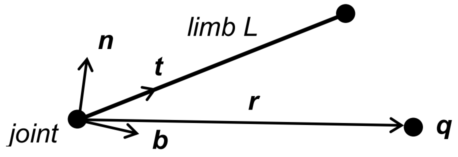

Figure 2. Frame *F* for *limb* ***L***.

Upon articulation, limbs move and rotate, which necessitates that frames F_i be recomputed. The new location of vertex $\boldsymbol{q}$ is given by:

$$\sum w_i\, (F_p + \boldsymbol{q'}_x F_t + \boldsymbol{q'}_y F_n + \boldsymbol{q'}_z F_b))$$

where F is the new frame for articulated *limb*$_i$ and $\boldsymbol{q}$' is $\boldsymbol{q}$ transformed by the at-rest frame F.

The initial w_i assigned to $\boldsymbol{q}_i$ are usually defaulted to some inverse function, such as $1/d_i^2$ where d_i is the distance between $\boldsymbol{q}$ and the nearest point on *limb*$_i$. Once computed, the weights are normalized, *i.e.*, $\sum w_i = 1$.

As detailed in [Lewis *et al.* 2000], these default w_i do not yield natural motions of the surface. Significant manual adjustment is usually required and good results difficult to achieve.

3. A New Method to Calculate Vertex Weights

To produce the reference frames and associated weights needed for skeletal vertex deformation, we utilize the *medial* of the three-dimensional object. The medial consists of two-dimensional surfaces and one-dimensional arcs, which connect along non-manifold seams and points. We represent the surfaces and arcs with triangles and segments, respectively, and refer to them as *primitives* P_i.

The original object may be reconstructed by defining a distance field around P_i. A similar but *differentiable* field is obtained by applying a *convolution filter* to P_i.

For a given vertex $\boldsymbol{q}$ of the object, a reference frame F_i is computed for each P_i that has at $\boldsymbol{q}$ a non-zero convolution value; this value is assigned to w_i. The actual method to align F_i within *Pi* is not significant but must be consistent; for triangle P, the normal and first edge, for example, may be used to construct the frame.

The stick-figure skeleton controls the medial. As the skeleton articulates, vertices of the medial are moved and new medial frames computed. Object vertices are recomputed using these new medial frames. Lists of affected vertices are maintained per stick-figure limb so that the locations of unaffected vertices are not recomputed.

This method is akin to that of [Stalpers and van Overveld 1997], in which rectangles control vertex deformation through reference frames associated with the rectangles. The rectangles are user-specified and the influence of a rectangle on an object vertex is determined by a network-based distance between rectangle and vertex. In contrast, the present work utilizes the medial, which may be derived from the object, and utilizes convolution to produce a differentiable field.

4. The Medial

The medial axis/surface (or simply 'medial') of an object is defined in [Blum 1967]. It is the locus of centers of all maximally sized spheres contained within the object. As such it is the 'center of support' for the object and provides a means to compute local object thickness.

One measure of the integrity of an animated shape is its retention of local thickness. This suggests that when computing w_i for vertex $\boldsymbol{q}$, the distance to the local 'center' of the object should be a factor. As shown below, the stick-figure skeleton is less accurate than the medial.

Aside from using object thickness to determine w_i, our method does not attempt to maintain a constant volume of the object.

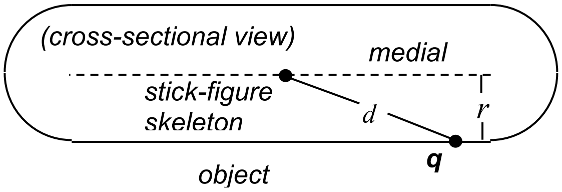

Figure 3. Distance d to the stick-figure skeleton (viewed head-on and represented as a single point) is a poor estimate of thickness r, as measured to the medial.

The medial exists unambiguously for any finite 3D object, but for even simple objects it can be quite complex and, in general, difficult to compute [Amenta *et al*. 1998]. In [Turkiyyah *et al*. 1997] methods of medial extraction are categorized.

More than one skeletal limb will influence those vertices of the medial that are near skeletal joints. This is accommodated by associating a frame F per influencing skeletal limb, and computing a weight via convolution of the limb. That is, the stick-figure skeleton controls the medial in a manner similar to the control of the object by the medial. This is illustrated in figure 4. An example object, its medial, and an articulated medial are shown in figure 5.

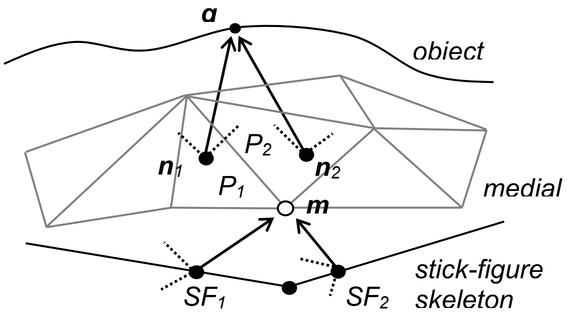

Figure 4. Reference frames (dashed) at $\boldsymbol{n_1}$ and $\boldsymbol{n_2}$, nearest points on triangles P_1 and P_2 to vertex $\boldsymbol{q}$; also shown are SF_1 and SF_2, stick-figure reference frames that control the medial vertex $\boldsymbol{m}$.

5. Convolution

A *convolution field* is produced around the medial by convolving a three-dimensional cubic filter (for example) with the medial. The width of the filter is proportional to and varies as the radius of the medial. Typically the filter is finite in extent, or windowed.

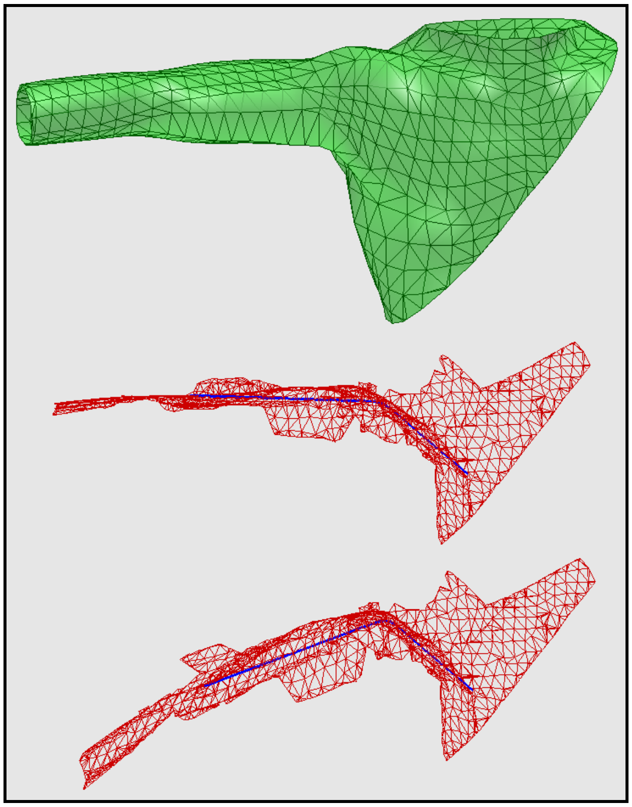

Figure 5. Top: object, middle: at-rest stick-figure skeleton and medial, bottom: articulated skeleton and medial.

The medial may be divided into manageably sized curve and surface primitives P_i. Let $Conv_i(\boldsymbol{q})$ be the value of the convolution field due to P_i, evaluated at $\boldsymbol{q}$. $Conv_i(\boldsymbol{q})$ is the 'influence' that the whole P_i (not simply a single point on P_i) exerts on $\boldsymbol{q}$, and is used for w_i. Because of the super-position property of convolution, the partitioning of the medial into P_i does not affect the overall convolution field at a vertex $\boldsymbol{q}$.

A discrete method to convolve curves and surfaces is developed in [Bloomenthal and Shoemake 1991]. An analytical method is given in [Sherstyuk 1999]. Convolution methods involve some effort to implement, but the computation itself is not unduly demanding. Once the vertex weights are computed, convolution is not used during the animation.

The following several figures employ a one-dimensional contour in two dimensions to compare three forms of deformation: unweighted, weighted by inverse-square, and weighted by convolution. The contour includes 52 points evenly distributed around the at-rest limbs.

The unweighted scheme, in which each contour point is associated with but a single limb, yields selfintersection when the joint is articulated.

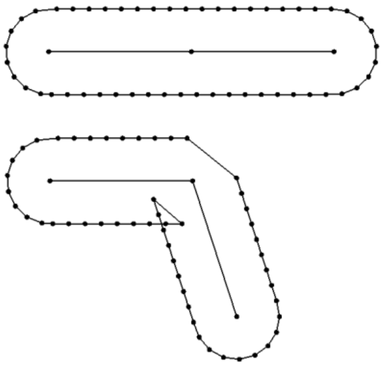

Figure 6. At-rest and articulated contours, unweighted

If, however, each contour point is computed as the weighted sum of two ***q’***, one for each of the two limbs, the results are much improved. In the following figure at left is a deformation in which $\boldsymbol{w_i} \propto 1/d_i^2$, and at right a cubic filter is applied to $limb_i$ and $\boldsymbol{w_i} \propto$ to the convolution value at ***q***.

With inverse-square weighting, the articulation distorts and shifts the shape; with convolution, the shape is pinched. Both functions are differentiable, which is desirable in that two surface vertices close to each other will have similarly close weights; otherwise, small joint rotations can induce abrupt surface motion

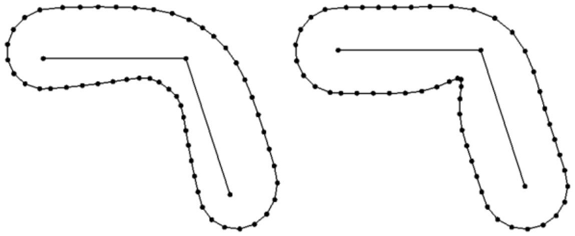

Figure 7. Left: weight based on $1/d^2$, where d is distance from ***q*** to limb; right: weight based on the convolution of a limb, measured at ***q***

7. Results

The setup time for computing the medial, convolving it, and producing the vertex weights for the torso shown in figure 5, is less than one minute on a 700 MHz processor. After setup, the following shapes are produced in real-time.

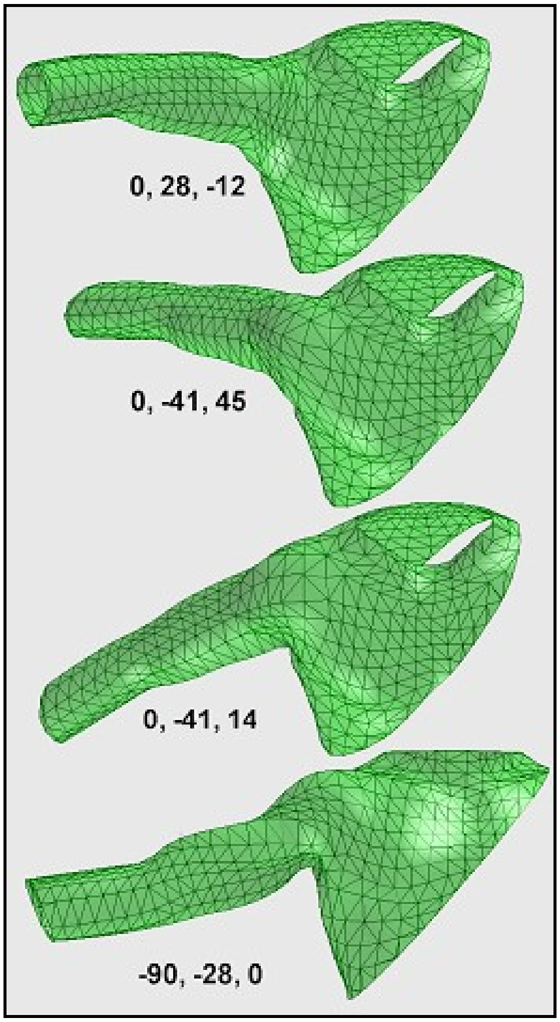

Figure 8. Articulation of the shoulder joint, with twist, pitch, and yaw given in degrees.

8. Conclusion

Conventional vertex deformation schemes typically employ a weighted blend of two points: one associated with the reference frame belonging to the articulating limb, the other associated with the reference frame of the parent limb. We have suggested that additional reference frames ameliorate the collapsing joint artifact, and that these frames may be obtained from the medial of the object.

The method automatically produces reasonable vertex weights, obviating the ‘tweaking’ otherwise needed. It shows good motion of the shoulder, a body part considered difficult to animate. Side by side comparisons with other methods would be a useful means to evaluate the method but, regrettably, such comparisons are beyond the scope of this paper.

Our deformation scheme is not a pose interpolation method such as described in [Lewis *et al.*] or [Sloan *et al.* 2001]. Nor is it a reconstruction scheme based on the medial, such as described in [Bloomenthal and Lim 1999], [Gagvani *et al.* 1998] or [Bittar *et al.* 1995]. It may, however, be used in conjunction with these methods.

There are many unanswered questions concerning the present technique, such as its robustness given a 'noisy' medial, its dependence on the resolution of the medial, its ability to use current hardware support, and so on. Especially difficult are numerous questions concerning the control of the medial by the stick-figure skeleton; the concept of 'skeletal distance' as developed in [Stalpers and van Overveld 1997] might ensure that object parts that are physically but not geodesically close (such as, *e.g.*, two fingertips) do not influence each other.

The method holds promise for the animation of objects modeled with conventional software or physically scanned. It also holds promise for interactive games, which require the speed of vertex deformation but often suffer from poor surface geometry near joints.

Acknowledgements

The author is indebted to the reviewers for their comments and suggestions.

The method reported here is patent-pending.

References

T.Akenine-Möller and E.Haines, *Real-Time Rendering*, 2nd ed., A.K. Peters Ltd., 2002.

Alias/Wavefront, *Learning Maya 2*, Toronto, 1999.

N.Amenta, M.Bern, M.Kamvysselis, *A New Voronoi-Based Surface Reconstruction Algorithm*, Proc. SIGGRAPH 1998.

E.Bittar, N.Tsingos, and M.-P. Cani, *Automatic Reconstruction of Unstructured 3D Data: Combining a Medial Axis and Implicit Surface*, Eurographics '95.

J.Bloomenthal and C.Lim, *Skeletal Methods of Shape Manipulation*., Proc. Shape Modeling Int'l, Aizu, Japan 1999.

J.Bloomenthal and K.Shoemake, *Convolution Surfaces*, Proc. SIGGRAPH 1991.

H.Blum, *A Transformation for Extracting New Descriptors of Shape*, Models for the Properties of Speech and Visual Form, Walter-Dunn, ed, MIT Press, 1967

N.Gagvani, D.Kenchammana-Hosekote, and D.Silver, *Volume Animation Using The Skeleton Tree*, IEEE Symposium on Volume Visualization, Research Triangle Park, NC, October, 1998, pp 47-53.

J.Lander, *A Heaping Pile of Pirate Booty*, Game Developer, v. 8, n. 4, pp. 22-30, April 2001.

F.Lazarus, S.Coquillart and P.Jancene, *Axial Deformations: an Intuitive Deformation Technique*, Computer-Aided Design v. 26, n. 8, 1994.

J.Lewis, M.Cordner and N.Fong, *Pose Space Deformation: a Unified Approach to Shape Interpolation and Skeleton-Driven Deformation*, Proc. SIGGRAPH 2000.

N.Magnenat-Thalmann and D.Thalmann, *Joint-Dependent Local Deformations for Hand Animation and Object Grasping*, Proc. Graphics Interface 1988.

A.Sherstyuk, Kernel Functions in Convolution Surfaces: a Comparative Analysis, The Visual Computer, v. 15, n.4, 1999.

P.Sloan, C.Rose III, and M.Cohen, *Shape by Example*, ACM Symp. Interactive 3D Graphics, 2001.

M.Stalpers and C.van Overveld. *Deforming Geometric Models based on a Polygonal Skeleton Mesh*. Journal of Graphics Tools, v. 2, n. 3, pp 1-14, 1997.

G.Turkiyyah, D.Storti, M.Ganter, H.Chen, and M.Vimawala, *An Accelerated Triangulation Method for Computing the Skeletons of Free-form Solid Models*, Computer-Aided Design, v. 29, n. 1, 1997.