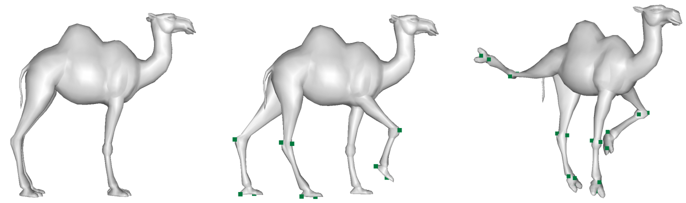
Shape Preserving Mesh Deformation

Alla Sheffer*
University of British Columbia

Vladislav Krayevoy†
University of British Columbia

1 Introduction

Deformation is the process of interactively transforming the surface of a model in response to some control mechanism. It is commonly used for model editing and animation. Typically, mesh deformation techniques require global knowledge of the model structure (such as a skeleton) and are quite time consuming. We propose a new approach for 3D mesh deformation based on a small number of user-specified *control vertices*. Given the positions of the control vertices, our method computes the positions of the rest of the vertices, in a manner that best preserves the shape parameters of the source model. As demonstrated by Figure 1, we generate natural looking deformations in seconds with minimal user interaction.



(a) (b) (c)

Figure 1: Examples of model editing; **a)** source model (4884 faces); **b)** walking camel - moderate deformation (9 control vertices, 3 sec); **c)** dancing camel - extreme deformation (18 control vertices, 12 sec). Computations were done on P4 2.4 Ghz computer.

2 Algorithm

The basic idea behind our algorithm is to describe the position of each vertex with respect to its neighbors in the mesh, rather than with respect to a global coordinate system. We would like the description to be invariant under rigid transformations in order to be able to move, bend or rotate parts of the model, such as limbs. One representation, describing a vertex with respect to its neighbors was introduced by Alexa [2001]. However, it is not invariant under rotation. Our representation is based on a set of angles and lengths relating a vertex to its immediate neighbors.

Representation: Let v be a mesh vertex in 3D and let $v_1, v_2, ..., v_m$ be its neighboring vertices. We define the projection plane

$$P = n_x x + n_y y + n_z z + d$$

using the normal at v, $n = (n_x, n_y, n_z)$, and $d = -\sum_{i=1}^{m} n \cdot v_i$. We define the projections of v and v_i to P as v' and v'_i respectively. The description of the vertex with respect to the neighbors consists of: a set of angles α_i between the projected edges $\langle v', v'_i \rangle$ and $\langle v', v'_{i+1} \rangle$; a set of angles β_i between n and the edges $\langle v, v_i \rangle$; and, a set of projected edge lengths $l_i = \|v' - v'_i\|$.

Reconstruction: Given those values we can uniquely define the vertex position given the neighbor vertices. We derive v' from v'_i using the reproduction property of the mean value coordinates [Floater 2003]:

$$v' = \sum_{i=1}^{m} w_i v'_i, \tag{1}$$

where the barycentric weights w_i are derived from α_i and l_i. To derive the position of v given v', we calculate a set of offsets along n, $h_i = \|v' - v'_i\| \cot(\beta_i) + (v_i - v'_i) \cdot n$. Finally we obtain v by offsetting v' by the average of h_i along n as follows:

$$v = v' + n \frac{1}{m} \sum_{i=1}^{m} h_i. \tag{2}$$

The presented shape description is invariant under rigid transformations. Given a 3D model, the angles and lengths are uniquely defined, though more than one model can fit a given combination. We use this redundancy to facilitate the deformation process, generating models which closely preserve the shape of the input models, subject to deformation.

Deformation: Given user prescribed positions for a set of control vertices V_c, the positions of the rest of the vertices are computed by iterating the following scheme:

1. For each vertex v, if v is not in V_c, update the position of v using the shape description in Equations 1 and 2.
2. For each control vertex v in V_c, compute the height offsets h_i and update the positions of the neighbor vertices:

   $$v_i = v'_i + n \frac{1}{m} \sum_{i=1}^{m} h_i.$$

   This sets the distance between v and the projection plane P to the average of the h_i values.
3. Repeat until convergence.

The shape preservation metrics that we use focus on angles. However, under deformation, the angles are sometimes preserved at the expense of stretch. To account for stretch we use a simple strategy of scaling the edge weights w_i by $\frac{\|v' - v'_i\|}{l_i}$. In addition to better length preservation, the use of a stretch component during vertex placement drastically speeds up the convergence of the reconstruction procedure.

3 Conclusions

We introduce a new, robust method for mesh deformation based on a local shape representation, which is invariant under rigid transformations. The sole input from the user consists of a number of control vertices defining the deformation. Given the input, our method provides natural deformations that include scaling, bending, and rotation of mesh parts, providing an effective tool for model editing and character animation.

References

ALEXA, M. 2001. Local control for mesh morphing. In *Proceedings of the International Conference on Shape Modeling & Applications*, IEEE Computer Society, 209.

FLOATER, M. S. 2003. Mean value coordinates. *Comput. Aided Geom. Des. 20*, 1, 19–27.

*e-mail: sheffa@cs.ubc.ca
†e-mail: vlady@cs.ubc.ca