Shape-aware as rigid as possible deformation

Alvaro Cuno, Claudio Esperança, Antonio Oliveira, Paulo Roma Cavalcanti
Federal University of Rio de Janeiro - PESC/COPPE

Abstract

We propose a formulation capable of deforming meshes in a shape-sensible way. We explain how to adapt the original space-deforming algorithm [2] into a skeleton-driven deformation scheme, where more sensitivity to the mesh geometry is achieved. The more natural results make it possible to use the technique for character animation.

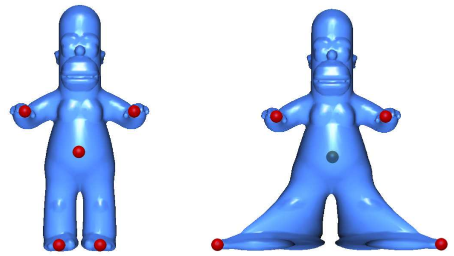

Figure 1. The standard as-rigid-as possible deformation gives non-natural results for meshes. (Left) The Homer model and five control points. (Right) Deformation result after moving two control points.

1. Introduction

Interactive deformation of 3D surface meshes is a process where the shape of a model is altered through constraints defined by the user. Usually, users manipulate the constraints using a 2D mouse and dragging the cursor on window canvas. No restrictions are given for mouse dragging, i.e., the constraints can be separated or stretched freely. However, the model shape should be preserved as much as possible giving the user the idea he is manipulating a real object.

The so-called “as rigid as possible” deformation techniques are of special interest for this task than other purely geometric deformation schemes [1, 3, 4, 5]. They avoid unnatural shearing and nonuniform scaling on the models since only optimal rigid-body transformations are used for driving the deformation process.

Recently, Schaefer et al. [4] used this paradigm for performing interactive image deformation. In summary, the problem for solving is how to find the rotation component for the optimal rigid-body transformation efficiently. They presented a closed-form solution using the relationship between similarity and rigid transformations and, thus, obtained attractive results. Their algorithm use as input a 2D image and a set of control points. After the user drags some of these control points, the image is “rigidly” altered. For each element of the image (not necessarily a pixel) is computed a transformation restricted to be “as rigid as possible”. They used a moving least squares optimization approach for computing a varying solution for each image element.

A proposal for deforming 3D meshes following the Schaefer’s criterium has been presented in [2]. Although they have presented an efficient formulation for computing the rotation component in 3D, non-natural results can be obtained when the deformation is applied on surface models, as the deformation process is blind to the shape of the model. Figure 1 illustrates this problem.

This work presents a proposal for applying “as rigid as possible deformations” sensitive to the model shape. We propose to solve the mesh deformation problem adapting the method presented in [2] into a skeleton-driven deformation scheme. Then, given a mesh model and its associated skeleton, we perform the deformation on the skeleton joints and project the transformations into the mesh. The shape sensitivity is realized by a distance metric defined over the skeleton bones. Figure 2 shows more natural results when compared with results on Figure 1.

2. Moving least squares deformation

Let $\{\mathbf{p}_i\}$ be a set of control points and $\mathbf{x} = [x\, y\, z]$ be a point in 3D. After new positions $\{\mathbf{q}_i\}$ are defined for the control points, the problem consists in finding the rigid-body transformation T that minimizes $\sum_i w_i |T_{\mathbf{x}}(\mathbf{p}_i) - \mathbf{q}_i|^2$, where w_i is a weight function defined as $w_i(\mathbf{x}) = |\mathbf{p}_i - \mathbf{x}|^{-2}$ and $|\cdot|$ denotes Euclidian distance metric.

It can be shown that the optimal MLS-based solution is defined as

$$T(\mathbf{x}) = R(\mathbf{x} - \mathbf{p*}) + \mathbf{q*}, \tag{1}$$

where $\mathbf{p*} = \frac{\sum_i w_i \, \mathbf{p}_i}{\sum_i w_i}$, $\mathbf{q*} = \frac{\sum_i w_i \, \mathbf{q}_i}{\sum_i w_i}$ and R is a rotation function defined by an angle α around an axis $\mathbf{a}$. In [2] it was shown that the vector that defines the rotation axis is given by the solution of the 3×3 system

$$(\mathrm{M} + \mathrm{M}^\mathsf{T} - \lambda \mathrm{I})\, \mathbf{u}^\mathsf{T} = \mathrm{V}^\mathsf{T},$$

where $\mathrm{M} = \sum_i w_i \hat{\mathbf{q}}_i^\mathsf{T} \hat{\mathbf{p}}_i$, $\mathrm{V} = \sum_i w_i \hat{\mathbf{p}}_i \times \hat{\mathbf{q}}_i$, $\hat{\mathbf{q}}_i = \mathbf{q}_i - \mathbf{q*}$, $\hat{\mathbf{p}}_i = \mathbf{p}_i - \mathbf{p*}$ and λ is the largest root of a depressed quartic equation. Additionally, expressions for computing angular parameters are given by

$$\cos(\alpha) = \frac{1 - \|\mathbf{u}\|^2}{1 + \|\mathbf{u}\|^2} \quad \text{and} \quad \sin(\alpha) = \frac{-2\|\mathbf{u}\|}{1 + \|\mathbf{u}\|^2}.$$

3. Shape sensitive deformation

The shape-aware deformation uses a skeleton to drive the deformation of the model. Once the skeleton is known, we define skin sets S_j composed of mesh vertices that will be influenced by the transformation of each bone b_j. Observe that, unlike classical skeletons used in character animation, in our skeleton there is no guarantee that each bone will be transformed rigidly, rather, we may think of each bone as a point set that will be transformed *as rigidly as possible* in a MLS sense. The process starts by applying transformations into the skeleton joints in much the same lines discussed in Section 2, except that Euclidian distances are replaced by path distances along the skeleton. The joint transformations, in turn, will define the transformations along the connected bones using linear interpolation. If we consider that a bone b_j contributes to the transformation of a vertex $\mathbf{v}$ by a factor proportional to the distance between them, say, $\rho(b_j, \mathbf{v})$, then that contribution is $\rho(b_j, \mathbf{v})T(c_j(\mathbf{v}))$, where $c_j(\mathbf{v})$ is the point of bone b_j closest to $\mathbf{v}$ and $T(\mathbf{x})$ stands for the optimal transformation of a point $\mathbf{x}$ of a bone.

All it remains is to find a weighting scheme ρ capable of ascertaining the smoothness of the deformed mesh. Let $h : [0, \infty) \to [0, 1]$ be any smooth function such that $h(0) = 1$, h decreases in $[0, r)$ and is null in $[r, \infty)$ for a given value $r > 0$ which acts as the maximum amplitude of a diffusion process. We use $\rho(b_j, \mathbf{v}) = h(g(\mathbf{v}, S_j))$, where the g is the geodesic distance between $\mathbf{v}$ and S_j and $h(d) = (1 - \frac{d}{r})^2$.

Finally, the transformation applied to any given mesh vertex $\mathbf{v}$ is given by

$$T(\mathbf{v}) = \frac{\sum_{b_j} \rho(b_j, \mathbf{v}) T(c_j(\mathbf{v}))}{\sum_{b_j} \rho(b_j, \mathbf{v})}.$$

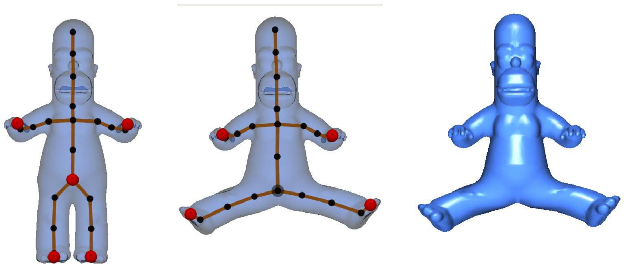

Figure 2. Original model, its skeleton and five control points defined on some joints of the skeleton (left) and their as rigid as possible skeleton-driven deformation (right).

Skin sets S_j may be defined by the user or using some algorithm. Vertex weights are computed by visiting all mesh points with a geodesic distance smaller than r from the border of S_j. The value of r is computed experimentally because it depends on the scale of the model.

4. Conclusions

Results on Figure 2 show a more phisically plausible deformation of the model when compared with the original formulation (Figure 1). The proposed skeleton-driven method is fairly simple and the deformation does not require inverse kinematics to be computed. Additionally, preliminar experiments show that this alternative substantially increase the frame rates of interactive deformation sessions of models with up to a few hundred thousand vertices.

Acknowledgement. The first author is supported by a CAPES/CNPq fellowship - IEL Nacional - Brasil.

References

[1] M. Alexa, D. Cohen-Or, and D. Levin. As-rigid-as-possible shape interpolation. *Proceedings of SIGGRAPH 2000*, pages 157–164, July 2000. New Orleans, USA, July 23-28.

[2] A. Cuno, C. Esperança, A. Oliveira, and P. R. Cavalcanti. 3D as-rigid-as-possible deformations using MLS. In *Proceedings of the 27th Computer Graphics International Conference*, pages 115–122, Petropolis, RJ, Brazil, May 2007.

[3] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. *ACM Trans. Graph.*, 24(3):1134–1141, 2005.

[4] S. Schaefer, T. McPhail, and J. Warren. Image deformation using moving least squares. *ACM Trans. Graph.*, 25(3):533–540, July 2006.

[5] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, July 2007.