A Prototype System for Character Animation Based on Real-time Deformations

Nikita Kojekine, Vladimir Savchenko*, Mikhail Senin**, Ichiro Hagiwara

Faculty of Engineering, Tokyo Institute of Technology, 2-12-1, O-okayama, Meguro-ku, Tokyo 152-8552, Japan

* Faculty of Computer and Information Sciences, Hosei University, 3-7-2 Kajino-cho Koganei-shi, Tokyo 184-584, Japan.

** Moscow Institute of Physics and Technology, Kerchenskaya str., house 1"A", building 1, Moscow 113-303, Russia.

Abstract

The problem described in this paper concerns transforming a given geometric shape into another in a continuous manner. Important examples of surface deformation have been intensively investigated in the past few years. Shape deformation of three-dimensional geometric objects is a time consuming operation, but at the same time for some applications, like computer games, an opportunity to produce plausible deformations for character animation fast is very important. For this application we have selected to use compactly supported radial basis functions, because as our benchmarks show they demonstrate their applicability for real time animation. Another very important issue is ability for the artists to define very easily (intuitively) the deformations. In this paper we report on the progress of our software system (available for free to download from our web page), providing an editor that assists in the design of animated objects, using only a moderate number of easy to define control vectors (possibly irregularly positioned).

Key words: *Radial basis functions, space mapping, animation.*

1. Introduction

Most existing shape transformation techniques fall into one of the three following categories (for more references, see [1]):

- mapping the space onto itself;
- metamorphosis;
- modification of defining functions.

The survey [2] discusses common mathematical foundations of the space deformation techniques. In this paper we will review works related to radial basis functions (RBF).

Vast literature is devoted to the subject of scattered data interpolation, which can be used for a space mapping, and if applied to some point set in the space, it changes this set to a different one. One of the approaches is to use methods of scattered data interpolation, based on the minimum-energy properties [3], [4], [5]. These methods are widely discussed in literature (see [6], [7]). As far as we know, the first publication on using discrete 2D landmark points is that of Bookstein [8], [9]. An approach (weighted least squares technique) to select destination points to adopt the configuration of minimum bending energy where the target landmarks or destination points are freed to slide along lines is discussed by Bookstein in [10]. The benefits of using RBFs have been recognized in many works and RBFs were adapted for 2D computer animation [11], medical application [12], [8], and reconstruction from 3D scattered data. Special methods to reduce the processing time were developed for thin plate splines and discussed in [13], [14], see also recent publications [15], [16].

Actually, the methods exploiting the RBFs can be divided into three groups. First group is “naive” methods, which are restricted to small problems, but they work quite well in applications dealing with shape transformation (see, for example [17], [18]). Second group consists of fast methods, which allow large data sets to be modeled [16], [13]. The third and last group is the Compactly

Supported Radial Basis Functions (CSRBF). Recently they were applied to reconstructing scattered data sets [19], [20].

We have selected CSRBF functions as a main tool for our animation system, because they provide smooth, plausible deformations in real-time, as described in the paper [21]. This optimization not only allows performing complex deformations in real-time, but also saves memory required for animation. For example, in the most popular 3D game engines (Quake, Unreal, Half-Life) for skeleton animation of character animation models all coordinates of all points of an object are stored for every frame of animation, what leads to huge amount of data stored for each model. When we use the described technique, only a small amount of additional information is needed.

Our next goal was to provide a simple, easy to use system for defining and viewing the animations based on this technology. The ability to transform the shape of a surface is useful in animation, especially for face simulation. The problems in this research area still remain among the most difficult, but it should be possible to use our system even in this field. For CSRBF the ability to define the radius of support r not only controls the speed of the deformation, but also controls the locality of the deformation, which enables to combine several CSRBF transformations on the one animation object with different radii of support and also to combine CSRBF transformation and other deformation techniques (like skeleton or inverse kinematics) easily. So our system should also provide the ability to combine several deformations on one object. And the system should be portable (available on the most of the current computer platforms) and should be free.

In the next chapter we are describing the developed software system (available to download for free from our web page [22]).

2. Software System

3D geometric modeling systems based on shape deformations have been pursued by many researchers and take mainly advantage of the simple idea that the user can define tangible geometry of deformations by defining starting and destination points (deformation vectors).

A software system was designed in C++ using open source software library "The Visualization Toolkit" (VTK) [23]. VTK is an open source, freely available software system for 3D computer graphics and consists of a C++ class library, and several interpreted interface layers including Tcl/Tk, Java, and Python. This toolkit has been tested on almost every Unix and PC (Windows) platform. Our C++ class for shape transformation can be used in the pipeline execution method (lazy mode) of VTK, that actually means that it can be combined with other shape transformation classes, including the other instances of the same class (for example, several CSRBFs shape transformations can be applied with different radii of support on the same object).

Here we give a short description on generation of a complete animated scene. In our software system the complete scene is represented as a collection of animation objects. Following three groups represent all supported movements of the object:

1. translation of the local coordinate system of an object along some trajectory (movement),
2. change of orientation local coordinate system of the object (rotation),
3. transformation of the object according to it's local coordinate system (transformation).

It is assumed that while applying transformation (3) the center of gravity of the object is not moving. The animation object can be constructed from several parts:

1. different independent transformations for various parts of an object could be applied,
2. different colors and textures for various parts of the object could be applied.

A prototype software system consists of two applications.

- "Picker" provides creation and translation of transformations. This program enables the user to interactively input control points and vectors for CSRBF spline calculation. The screenshot of the "Picker" level interface is shown on Figure 1 (a).

- “Animation composer”. This interface enables the user with a possibility to interactively define trajectory and provide orientation and timing marks (i.e., the schedule of the movement of an object along it’s trajectory). Kochanek-Bartels interpolating spline [24] is used for animating of the movement and quaternion calculus [25] is used for animating rotation. The screenshot of this interface is shown on Figure 1 (b).

The “Animation composer” program also supports an “Animation” mode. In this mode the program displays animated objects in accordance to their defined trajectories, rotations and transformations. This program also enables the user to save animation results as a movie in a personal computer movie file formats (avi).

Our interface level software was designed and tested for use on a PC under Windows (9x, ME, NT, 2000, XP). Command line interface version of the “Animation” program was also developed and it was tested on both Windows and Linux platforms.

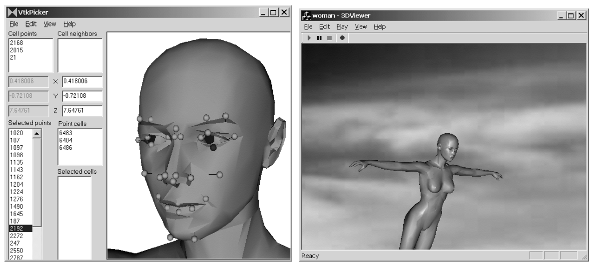
3. Tests and Benchmarks

To test our animation system we used a model of a woman’s head to test it as a model for a simple facial animation. Figure 2 shows five frames from generated set of animated frames for the simple deformation “smile” (see Figure 3), defined only by 32 control vectors. In order to localize this deformation we have selected a small radius of support r=0.2.

As Figure 3 shows even to define quite good-looking deformations only a moderate number of control vectors is needed. Ability to provide the locality allows us to define them very easily. In Figure 3 the resulting states of various simple deformations are shown in the first row (“smile”, “upset”, “kiss”, first image in the first row is the initial shape). The vectors defining corresponding deformation are shown in the second row.

Real time deformations are important not only for character and facial animations. For example, in Figure 4 two different deformations of a polygonal box are shown. Vectors from the center of the box using C2 -continuous Wendland’s function create these deformations. You can see that the resulting deformation is very large (vector lengths are about half of the box size) that shows an applicability of our method even for large transformations. For example, such real-time deformations can be used to model elastic environment (i.e. walls, ceiling) in computer games.

In our software system we have performed the speed benchmarks for this animations on our test computer: Athlon 1Ghz, 650MB RAM, ATI Radeon 250Mhz 8500LE 64MB video board VIA KT133 Chipset, Windows 2000 SP2. These results are shown in Table 1.



(a) (b)

Figure 1. User level interfaces: (a) “Picker” interface, (b) “Animation Composer” interface.

Name	**Number of polygons defining the model**	**Number of vectors defining the deformation**	**Selected by the user “Radius of support” parameter**	**Resulting speed benchmark (in frames per second – fps)**
Facial animation “smile”	7024	32	0.2	107
“Elastic box”	758	6	0.6	72

Table 1. Performance benchmarks.

4. Conclusions

We have described our software system that can be used to generate simple shape transformations, like “smile”, “kiss”, etc., and save them as a compiled transformation files. Those transformations files can be later used in conjunction with each other, to provide more complex facial gestures.

Figure 2. Several frames from the “smile” deformation movie.

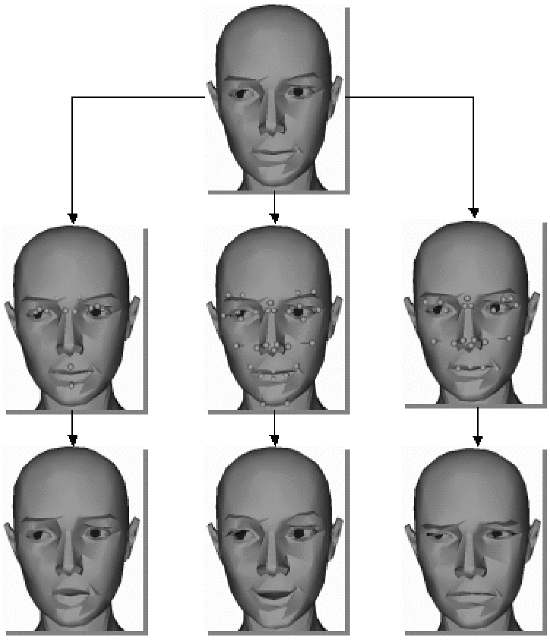


Figure 3. An example of a simple facial animations - "kiss", "smile" and "upset" transformations. Top row - original model, second row - defining vectors (selected by the user), third row - resulting deformation.

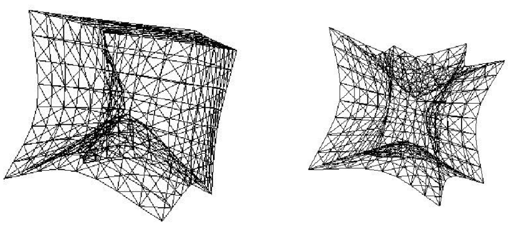


Figure 4. An example of the elastic environment deformation: an “Elastic box”.

There are several directions for future research. Firstly, we continue to design the end-user interface to create expressive 3D animations; currently we are planning to add other types of transformations (at least to use standard approaches of key frame and inverse kinematics for animating a character) to create an effective and complete animation system.

Secondly, we are also working on extending the interface system to create an artificial model with a full system of expressions, which can be easily modified and combined to create more complex facial gestures, like talking or expressing feelings in some defined situations.

References

[1] V. Savchenko and A. Pasko, Transformation of Functionally Defined Shapes by Extended Space Mappings, *The Visual Computer*, **14**, 257-270, 1998.

[2] D. Bechmann, Space deformation models survey, *Computers & Graphics*, **18(4),** 571-586, 1994.

[3] J. H. Ahlberg, E. N. Nilson, J. L. Walsh, *The Theory of Splines and Their Applications*, Academic Press, New York, 1967.

[4] J. Dushon, *Splines Minimizing Rotation Invariants Semi-Norms in Sobolev Spaces, Constructive Theory of Functions of Several Variables,* W. Schempp and K. Zeller (Eds.), Springer-Verlag, 85-100, 1976.

[5] V.A. Vasilenko, *Spline-functions: Theory, Algorithms, Programs*, Novosibirsk, Nauka Publishers, 1983.

[6] R. M. Bolle, B. C. Vemuri, On Three-Dimensional Surface Reconstruction Methods, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13(1),** 1-13, 1991.

[7] G. Greiner*, Surface Construction Based on Variational Principles, Wavelets, Images and Surface Fitting*, P. J. Laurent et al. (Eds.), AL Peters Ltd., 277-286, 1994.

[8] F. L. Bookstein, Principal Warps: Thin Plate Splines and the Decomposition of Deformations, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11(6)**, 567-585, 1989.

[9] F. L. Bookstein, *Morphometric Tools for Landmark Data*, Cambridge University Press, 1991.

[10] F. L. Bookstein, Two Shape Metrics for Biomedical Outline Data: Bending Energy, Procrustes Distance, and The Biometrical Modeling of Shape Phenomena, *Proc. Shape Modeling Conference (SMIA`97)*, March 3-6, Aizu-Wakamatsu, Japan, 110- 120, 1997.

[11] P. Litwinovicz, L. Williams, Animating Images with Drawing, *Computer Graphics, Proc. SIGGRAPH`94,* 409-412, 1994.

[12] J. C. Carr, W. R. Fright and R. K. Beatson, Surface Interpolation with Radial Basis Functions for Medical Imaging, *IEEE Transaction on Medical Imaging*, **16(1),** 96-107, 1997.

[13] R. K. Beatson, W. A. Light, Fast Evaluation of Radial Basis Functions: Methods for 2-D Polyharmonic Splines*, Tech. Rep. 119, Mathematics Department, Univ. of Canterbury, Christchurch*, New Zealand, Dec. 1994.

[14] W. Light, *Using Radial Functions on Compact Domains, Wavelets, Images and Surface Fitting*, P. J. Laurent et al. (Eds.), AL Peters Ltd., 351-370, 1994.

[15] J. C. Carr, T. J. Mitchell, R. K. Beatson, J. B. Cherrie, W. R. Fright, B. C. McCallumm, and T. R. Evans, Reconstruction and representation of 3D Objects with Radial Basis Functions, *Computer Graphics, Proceedings. SIGGRAPH`2001*, 67 – 76, 2001.

[16] L. Greengard, V. Rokhlin, A Fast Algorithm for Particle Simulation, *J. Comput. Phys*, **73**, 325-348, 1997.

[17] V. Savchenko and L. Schmitt, Reconstructing Occlusal Surfaces of Teeth Using a Genetic Algorithm with Simulated Annealing Type Selection, *Proc. 6th ACM Symposium on Solid Modeling and Application)*, Sheraton Inn, Ann Arbor, Michigan, June 4-8, 39-46, 2001.

[18] V.V. Savchenko, A.A. Pasko, T.L. Kunii, and A.V. Savchenko, Feature based sculpting of functionally defined 3-D geometric objects, T.S. Chua et al. (Eds.), *Multimedia Modeling, Towards Information Superhighway*, *Proc. MMM'95*, Nov., 341-348, 1995.

[19] B. Morse, T. S. Yoo, P. Rheingans, D. T. Chen, and K.R. Subramanian, Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions, *Shape Modeling conference, Proc SMI`2001*, Genova, Italy, May, 89-98, 2001.

[20] N. Kojekine, V. Savchenko, D. Berzin, I. Hagiwara, Software Tools for Compactly Supported Radial Basis Functions, *Computer Graphics and Imaging, Proc. IASTED CGIM`2001*, Hawaii, USA, August 13-16, 234-239, 2001.

[21] N. Kojekine, V. Savchenko, M. Senin, I. Hagiwara, Real-time 3D Deformations by Means of Compactly Supported Radial Basis Functions, *Short papers proceedings of Eurographics 2002*, Germany, Saarbrucken, September 2-6, 2002.

[22] http://www.karlson.ru/csrbf/

[23] *The Visualization Toolkit Textbook and open source C++ Library, with Tcl, Python, and Java bindings*, published by Kitware, 2001.

[24] D. Kochanek, R. Bartels, Interpolating Splines with Local Tension, Continuity, and Bias Control, *Computer Graphics*, vol. 18, no. 3, pp. 33-41, July 1984.

[25] K. Shoernake, Animating rotation with quaternion calculus, *ACM SIGGRAPH'87*, Course notes 10, Computer animation: 3-D Motion, Specification and Control, 1987.