4D Objects for Animation: Immersion on Virtual Reality

Sylvain Brandel[1]

1. Université Louis Pasteur Strasbourg I, LSIIT, Pôle API, Bd Sébastien Brant, F-67400 Illkirch
brandel@lsiit.u-strasbg.fr

Abstract

4D object modeling, also called space-time modeling, allows to build animations with topological modifications, like metamorphosis or implicit surfaces, with high level control. Starting from a 4D objects modeler for animation, we propose a tool to manipulate 4D objects embedded on imported scenes, running on various OS and environments. Since the perception of 4D objects is complex, we propose new concepts of 4D visualization, using power of our immersive equipment.

Key-words: space-time animation, virtual reality, immersion.

1 Introduction

4D object modeling allows to generate a large field of animation types, with various topological modifications. But fine control and smooth results necessitate heavy 4D objects, which are complex to manipulate on workstations. For this reason we propose to use virtual reality to improve 4D objects perception and control.

We establish that projecting a 4D object on a 3D space is as natural as projecting a 3D object on a 2D screen. Moreover, interaction metaphors associated to immersive visualization system, like projected large screens or head-mounted display, help in the perception of 4D objects and controlling the resulting animation.

Section 2 gives a brief state of the art on geometric animation. Section 3 describes 4D modeling, problems with perception and visualization and presents our modeler of 4D objects for animation. Section 4 presents the contribution of virtual reality to the problems of immersion and visualization of 4D objects, and interaction methods implemented in a multi viewer.

2 Research Status

Geometric modeling and physical model simulation are two usual ways to compute animations on computer graphics. On the geometric modeling field, we are interested with 3D animations with topological modifications, like smooth merging, disconnection, appearance or disappearance of objects. These kinds of effects can only be obtained using metamorphosis [LV98], implicit surfaces [Blo97], and models like F-Rep (for Function Representation) [FPA00] or 4D objects [BBB98].

Simple techniques can define metamorphosis between objects with same topology: for example fusion of grids [KCP92], recursive approach [Par92] or use of axis [LV98]. Other techniques allow topological modifications during the metamorphosis: holes handling [DG96], region correspondence [TPG01] or intuitive scheme definition [ZSH00].

Implicit surfaces are common way to define animations with topological modifications [WMW86, SP01], but need a heavy post-processing for displaying animations, using polyhedral tessellation [NB93] or ray-tracing [GA98].

Function Representation [PAS01, PPI+02] allows to represent hyper volumes modeled by implicit surfaces, CSG, B-Rep or solid modeling, and to homogenously manipulate these objects. F-Rep Functions are powerful but computationally demanding and difficult to manipulate.

4D modeling, also called space-time modeling, was initially introduced by using DOGME, an FFD model [BB91] to deform 4D objects with time-constant constraints [BD93] (Figure 3.a) or non time-constant constraints [AB97] (Figure 3.b), extended with a multidimensional deformation tool [BE01]. Space-time objects can also be used without deformation to obtain animations by direct modeling [BBB98, BBB00] or Cartesian product [SL01].

4D objects allow to obtain a large field of animations with a simple data structure, easy to manipulate using standard graphic routines without additional material or technologies. We chose them for all these reasons.

3 4D Objects

3.1 4D Modeling

4D object modeling consists in integrating geometric (3 dimensions) and time information in only one object, called space-time or 4D object (Figure 1), in which any axis can be associated to time axis. We use a topological model, the Generalized Maps [Lie89] (Figure 2) to represent these 4D objects. G-Maps define a unique basic element, the dart (Figure 2.a), with several relations, the α-links (Figure 2.b), allowing to easily extract various cells (Figure 2.c, Figure 2.d and Figure 2.e).

To build an animation, we use a cutting algorithm to intersect these objects by a hyperplane orthogonal to the chosen time axis. So the animation results from the succession of intersected objects at different times (Figure 3), or by fixing a time and manipulating the 4D objects.

3.2 Perception and Visualization

We have facilities to build, represent and manipulate 3D objects, because we are living on a 3D world. The problem of the perception of 4D objects (Figure 4) is more complex, since we are not used to working with them. A common way is to proceed by analogy: we consider 3D objects and imagine how to add a dimension. This concept works often fine, but, in some cases, adding a dimension is not intuitive: for example, a rotation axis has two dimensions in a 4D space, and it's hard to imagine how to rotate a 4D object around a plane (Figure 5).

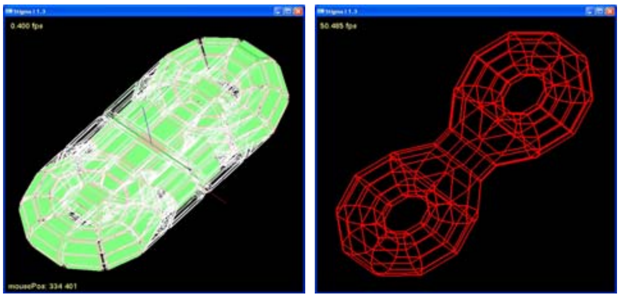


Figure 1. *4D object and animated object.*

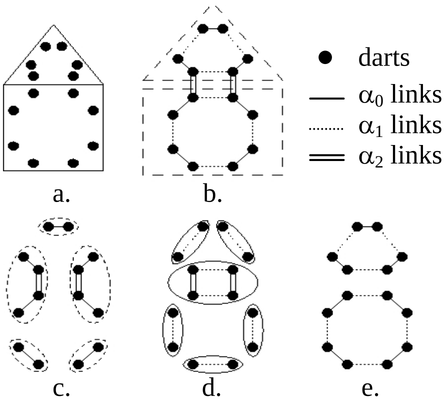


Figure 2. *2-GMap description: a) darts corresponding to geometric object; b) relations between darts; c) vertex orbit; d) edge orbit; e) face orbit.*

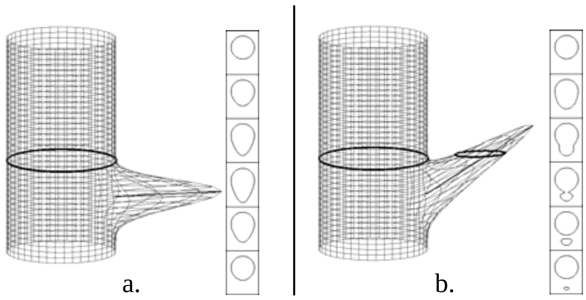


Figure 3. *Intersections and resulting animation. The space-time object is deformed with a time-constant constraint (a) or non constant-time constraint (b).*

3D modelers offer usually 3 views corresponding to projection on the three orthotropic planes. Thus, a 4D modeler proposes 4 views for each projection on the four orthotropic spaces. Then it's possible to view a whole 4D object by projecting it on a 2D plane (Figure 6), or by representing the fourth dimension by translation or scale (Figure 7). But all these modes offer only partial view and we need other tools to perceive these 4D objects.

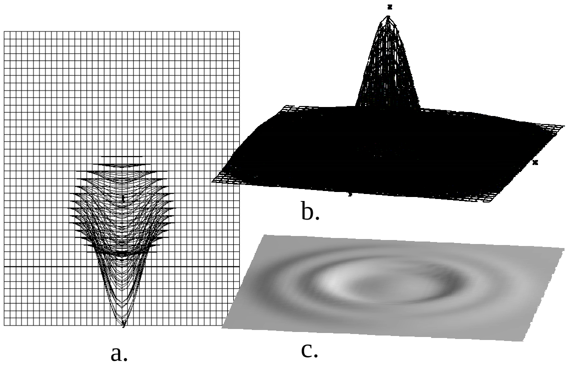


Figure 4. *A 4D object (a and b) corresponding to a water drip falling on a liquid, and animation (c).*

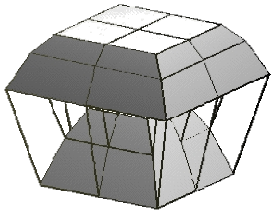


Figure 5. *4D rotation of a hypercube.*

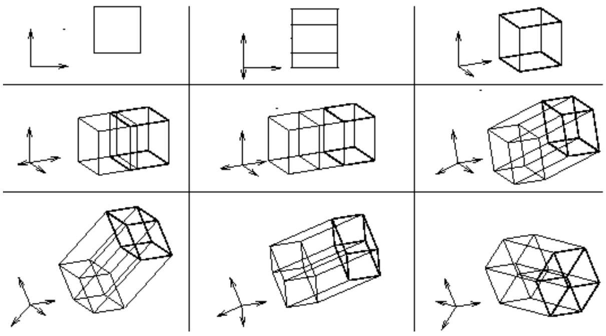


Figure 6. *A hypercube projected on a 2D plane.*

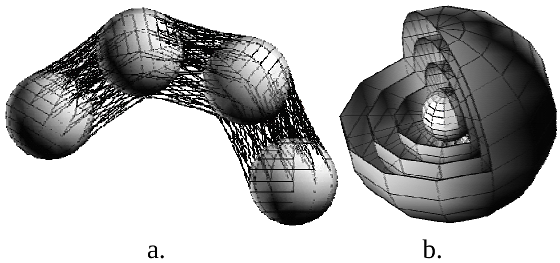


Figure 7. *A hyper revolution of a sphere, fourth dimension represented by translation (a) or scale (b).*

3.3 STIGMA

4D concepts are implemented in STIGMA (Space Time Interpolation for the Geometric Modeling of Animations), our modeler of 4D objects dedicated to animation [BBB98]. This modeler provides several modeling operations: 3D and 4D primitives, operations to increase the dimension of a given object, such as extrusion, Cartesian product and thickening, and an operation to decrease the dimension used to compute the animation. Some view modes are available: the four 3D projections of 4D objects, various representations of whole 4D objects, intersections sequences, and obviously the resulting animation.

4 Virtual Reality for 4D Objects

We present in this section 4 improvements of virtual reality for 4D objects: immersion, visualization, interaction, application control. After we present a viewer of multiple object formats.

4.1 Immersion

We saw on Section 3 the main problem with 4D modeling is the visualization of 4D objects. It's often possible to represent 3D objects and to do analogies, but we have to really represent the four dimensions if we want to directly interact on 4D objects for controlling the resulting animations.

In classical geometric modeling, we decrease the dimension by projecting 3D objects on 2D screens. Thus, by analogy, we project 4D objects on 3D workspaces. Immersive visualization systems provide large workspaces, allowing great perception of huge 4D objects. Big size high resolution screens, multiple screens and stereoscopic vision increase this feeling. 3D tracking allow intuitive global manipulation, internal cells designation and navigation through 4D objects.

4.2 Visualization

On immersive systems, the workspace is a real 3D space, without boundaries. So we implement new concepts of the four projections. These projections can cohabit without material separation, and the active view is highlighted whereas the three others are half lighted. To activate a view, the user just has to walk towards the wished view, or to use his interaction tool [SB05]. Moreover, the user can place them as he wants, classically on square or by using all the dedicated space, comprising the Z depth (Figure 8). For example the active view is near the user, others views are always visible but further.

4.3 Interaction

Interaction with 4D objects is performed by selection, manipulation and navigation concepts, using 3D devices.

Laser metaphor is used to point any view of a 4D object, cells of 4D objects or resulting animation, then the selected element is highlighted by drawing its bounding box (Figure 9).

Global manipulation allows performing translations, rotations or scale on the pointed view, possibly applied to other views (Figure 9.a). If only a part of the 4D object is highlighted, a transformation is applied to the designated cell, with repercussion to the corresponding animation (Figure 9.b).

Navigating through 4D objects is implemented using the flying metaphor [WO91]. This metaphor enables the user to move inside 4D objects, controlling the orientation of the camera with the 3D device.

4.4 Application Control

Working on the VR environment and more precisely modeling, can be performed using two approaches: using application control [WC01, GB05] or not. Since a VR working session is shorter than on a workstation, we optimize the immersion time by drastically reducing the application control time. That is why our implementation proposes a minimal application control engine: data files and initial options are specified on the command line, and during the session, the control is only performed by hitting keys.

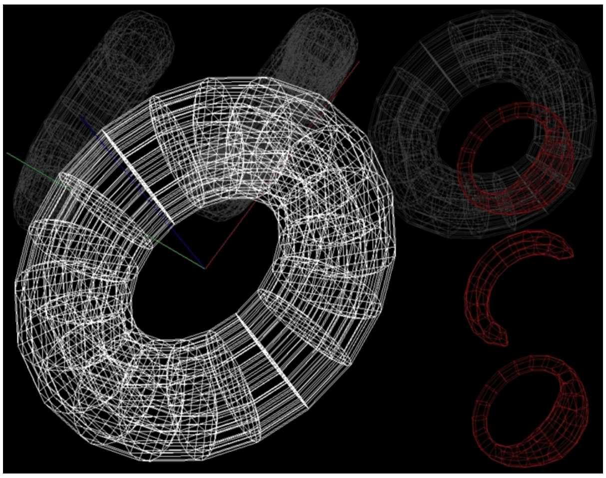
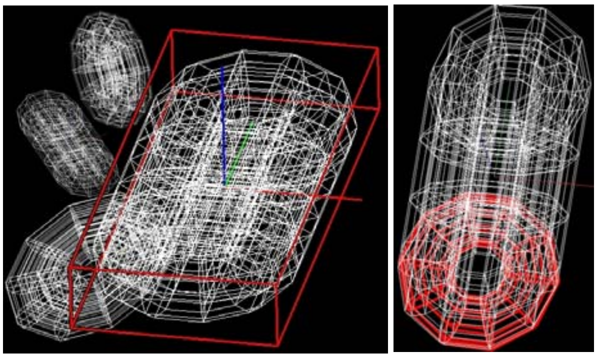


Figure 8. *Workspace filling.*



a. b.

Figure 9. *Global (a) or local (b) selection.*

4.5 STIGMA Viewer

We developed a multi-viewer to visualize and manipulate any objects. Input data can be VRML (using OpenInventor and OpenSG), G-Maps (modeled with STIGMA) or PDB (for Protein Data Base) objects. Objects can be displayed simultaneously, allowing to embed animated objects on a VRML scene for example (Figure 10). Functionality depends on the considered object: interacting with a protein, or providing various displaying modes and interaction methods for 4D objects. This viewer runs on both Linux and Windows platforms, using GLUT or VRJuggler on workstation and virtual reality environment.

5 Conclusion and Future Works

This work shows the contribution of virtual reality to 4D modeling. New concepts of visualization of 4D objects offer a better usage and filling of the 3D workspace, so the user can walk through the space-time scene and control the resulting animation at the same time. Our immersive system, with big size display panels, proposes a better perception of 4D objects than on classical workstations, enhancing thus production of animations performed by graphic artists. Moreover adapted interaction methods help to improve 4D modeling and control.

Following this work, we actually explore the other side of 4D objects and virtual reality connection, that is the use of 4D objects for virtual reality. The goal is to describe application control with 4D objects animations, and use 4D objects to store interface elements.

Figure 10. *4D animation embedded on a VRML scene.*

References

[AB97] F. Aubert and D. Bechmann. *Animation by Deformation of Space-Time Objects*. Eurographics' 97, 16(3): 57-66, 1997.

[BB91] P. Borrel and D. Bechmann. *Deformation of N-dimensional objects*. International journal of compu-tational geometry and applications, 1(4): 427–453, 1991.

[BBB98] S. Brandel, D. Bechmann and Y. Bertrand. *STIGMA: a 4-dimensional modeler for animation*. 9th Eurographics Workshop on animation and simulation, Lisbon, Portugal, 1998.

[BBB00] S. Brandel, D. Bechmann and Y. Bertrand. *Thickening: an operation for animation*. The Journal of Visualization and Computer Animation, 2000, 11: 261–277, N. Magnenat Thalmann & D. Thalmann editor, John Wiley & Sons publisher, 2000.

[BD93] D. Bechmann and N. Dubreuil. *Animation through space and time based on a space deformation model*. The journal of visualisation and computer animation, 4(3): 165–184, 1993.

[BE01] D. Bechmann and M. Elkouhen. *Animating with the multidimensional deformation tool*. Workshop on Computer animation and simulation 2001.

[Blo97] J. Bloomenthal. *An introduction to Implicit Surfaces*. Morgan Kaufmann ed., 1997.

[DG96] D. DeCarlo and J. Gallier. *Topological evolution of surfaces*. Graphics Interface'96, pp. 194–203, 1996.

[FPA00] E. Fausett, A. Pasko and V. Adzhiev. *Space-Time and Higher Dimensional Modeling for Animation.* Computer Animation 2000, pp. 140–145, 2000.

[GA98] E. Galin and S. Akkouche. *Fast Surface Reconstruction from Contours using Implicit Surfaces*. Implicit Surfaces'98 Conference, 3: 139–144, 1998.

[GB05] D. Gerber and D. Bechmann. *The spin menu: A menu system for virtual environments*. IEEE Conference on Virtual Reality, pp 271–272, 2005.

[KCP92] J.-R. Kent, W.-E. Carlson and R.-E. Parent. *Shape transformation for polyhedral objects*. SIGGRAPH'92, 26: 47–54, 1992.

[Lie89] P. Lienhardt. *Subdivision of N-Dimensional Spaces and N-Dimensional Generalized Maps*. 5th ACM Symposium on Computational Geometry, Saarbrücken, Germany, pp. 228–236, 1989.

[LV98] F. Lazarus and A. Verroust. *Three-dimensional metamorphosis: a survey*. The Visual Computer, 4(3): 275–324, 1998.

[NB93] P. Ning and J. Bloomenthal. A*n evaluation of implicit surface tillers*. IEEE Computer Graphics & Applications 13(3): 33–41, 1993.

[Par92] R. E. Parent. *Shape transformation by boundary representation: a recursive approach to establishing face correspondences*. The Journal of Visualization and Computer Animation, 3: 219–239, 1992.

[PAS01] A. Pasko, V. Adzhiev and B. Schmitt. *Constructive Hypervolume Modelling.* Graphical Models, special issue on Volume Modeling, 63(6): 413–442, 2001.

[PPI+02] G. Pasko, A. Pasko, M. Ikeda and T. Kunii. *Bounded blending operations*. Shape Modeling and Applications, IEEE Computer Society, pp 95–103, 2002.

[SB05] L. Sternberger and D. Bechmann. *Deformable raycasting interaction technique*. In YoungVR February, 2005.

[SL01] X. Skapin and P. Lienhardt. *Using Cartesian product for animation*. The Journal of Visualization and Computer Animation. 12(3): 131–144, 2001.

[SP01] K. Singh and R. Parent. *Joining polyhedral objects using implicitly defined surfaces.* The Visual Computer, 17: 415–428, 2001.

[TPG01] G. Treecd, R. Prager and A. Gee. *Volume-based three-dimensional metamorphosis using sphere-guided region correspondence*. The Visual Computer, 17: 397–414, 2001.

[WC01] P. Wellard and S. C. Chapman. *An investigation into the design of an interface for interaction with a virtual environment representing a four-dimensional object*. EGVE01 (7th EG Workshop on Virtual Environments & 5th Immersive Projection Technology Workshop), Stuttgart, 16–18 May, 2001.

[WMW86] B. Wyvill, C. MacPheeters and G. Wyvill. *Animating soft objects*. The Visual Computer, 2(4): 235–242, 1996.

[WO91] C. Ware and S. Osborne. *Exploration and Virtual Camera control in the Virtual Three Dimensional Environments*. ACM Symposium on Interactive 3D Graphics, in Computer Graphics, 24(2):175-183.

[ZSH00] M. Zöckler, D. Stalling and H.-Ch. Hege. *Fast and intuitive generation of geometric shape transitions.* The Visual Computer, pp 241–253, 2000.