Space-Time and Higher Dimensional Modeling for Animation

Eric Fausett, Alexander Pasko, Valery Adzhiev[*]

Shape Modeling Lab, University of Aizu, Aizu-Wakamatsu, 965-8580 Japan.
efausett@mail.com, pasko@acm.org

[*] Department of Computer Science, Moscow Engineering Physics Institute,
31, Kashirskoye sh., Moscow, 115409 Russia.
valery@acm.org

Abstract

There are limitations to the current BRep based "model then animate" paradigm when animating time-dependent (dynamic) objects. This paper describes an approach to modeling dynamic objects directly in space-time or even higher dimensional space. The use of the function representation (FRep) is proposed due to the ease with which it handles higher dimensions. After creating a multidimensional FRep object, it then needs to be mapped to "multimedia" space. In the case of animation this means at least one geometric coordinate must be mapped to a dynamic coordinate (time). A case study is presented to explain the technique and demonstrate its use in the animated metamorphosis between several shapes of differing topology.

1 Introduction

There are various kinds of movement considered in animation. The virtual camera can move in relationship to the shapes in the scene, the shapes in the scene can move relative to and interact with one another, or the shapes themselves can change over time or from one image to another. We will call these types of shapes dynamic objects, and it is these types of objects with which we are most concerned in this work.

There is a traditional division in computer animation made between the modeling and the animating processes. When working with dynamic objects however this division is blurred, and at the very least the animation of the object must be taken into consideration when modeling the object itself. The traditional division between the modeling and animation processes leads to limitations in what can be animated.

One important example of such a limitation occurs when attempting to animate the so-called metamorphosis between objects. While some types of metamorphosis are supported, these methods are often very cumbersome. The difficulty arises with the need to manually establish a correspondence between points on the objects. Moreover, a robust method of metamorphosis between objects of arbitrary topology is not known (see [4] for survey). Another problem occurs when applying deformations to boundary representation (BRep) solids or polygonal surfaces. In many cases deformations can lead to self intersections in the object. The application of deformations can also result in unwanted sharp irregularities and/or edges.

To avoid the limitations introduced when the modeling and animation processes of dynamic objects are separated, we must combine them into a single step. The most direct way that we can work with both the shape and it's animation at the same time is to model directly in space-time using the function representation (FRep) [6]. We can take this approach a step further and model in even higher dimensions. After this has been done the objects must be then mapped to geometric coordinates, time, and perhaps other "multimedia coordinates". This approach is not widely supported in spite of the fact that the creation, display, and animation of such multidimensional point sets (shapes) can be very useful in such fields as mathematics, the natural sciences, data mining, and aesthetic and industrial design. A similar approach to the direct modeling of space-time objects in BRep was proposed by Aubert and Bechmann [2]. In this approach however, the complexity of the resulting shapes is limited by the key modeling tools used, namely free-form deformations. In this paper we pursue the same general course to deal with objects of arbitrary complexity and higher dimension in FRep.

Another area that will be considered in this work is the modeling tools with which such dynamic objects are created. Most current modeling and animation systems

provide set theoretic operations for the creation of objects, however these operations do not work between all model types, and many times a conversion to polygons or another representation is necessary before such an operation can be applied. Blending and fillets are also not usually supported for all set theoretic operations. In addition, the ability to easily extend the primitives and operations available for the modeling and animation process is not widely supported. Such extension capabilities would be very useful, if not necessary for the modeling and animation of higher dimensional objects. It would be very difficult to provide all useful higher dimensional primitives and operations in a default modeling system.

In section 2 we will consider the benefits of FRep for solving the problems presented above. Next, we discuss the process of mapping geometric coordinates to "multimedia space" in section 3. This is the last step in creating an animated dynamic object. Finally, in section 4 we will present some simple ways in which modeling directly in space-time or higher dimensions gives us greater freedom of expression in animating dynamic models. The complete animation entitled "Homotopic Fun in 5D Space" is presented.

2 FRep and BRep in Animation

The function representation (FRep) defines a geometric object by a single continuous real function of several variables in the form $F(X) = 0$ [6]. This is a generalization of an implicit surface in which $F(X) = 0$ represents the iso-surface of an object. In these equations, X is a vector of point coordinates in n-dimensional Euclidean space. This Function representation can be used to define objects of arbitrary dimension. We can check any point in the n-dimensional space by taking the coordinates of that point and inserting it into our function. If the value of the function is negative, then the point exists outside the model. If the value of the function is zero, then the point lies on the surface, and if the value of the function is positive, then the point lies within the object.

A set of operations which are closed on the representation are provided. These include set-theoretic operations, blending, offsetting, and many others. Their being closed on the representation means that the result of any operation is expressed itself by a function, and thus can be used as the argument of any other operation. In addition, no special restrictions are imposed on the function definition. In principle, the FRep object is just a black box that takes coordinates as input and returns the function value.

A natural result of the FRep specification is that objects of any dimension can be modeled and manipulated without any special considerations. There are operations that allow for modeling in higher and lower dimensional space. Two examples are the Cartesian product, which increases

an objects dimension, and projection, which decreases an objects dimension.

There are several advantages that FRep holds over BRep. These advantages result in greater flexibility in both modeling and animation. The set of primitives and operations used in defining FRep objects is not fixed. New primitives and operations can be introduced by the user. These extensions can be specified using any combination of analytical, procedural, or tabular definitions. Currently in most modeling systems, the user has no way of adding to the default set of primitives and operations.

Different types of non-linear deformations are generalized by extended space mappings [9]. Many times in BRep, self-intersections can appear when applying certain operations such as deformation. In FRep however, such self-intersection can not happen. Unwanted sharp and irregular edges do not appear when using such deformations in FRep as can happen when using the same deformations in other representations. This is supported by the continuity of the defining functions for both objects and transformations in FRep. In contrast to BRep, metamorphosis between models can be performed automatically, even when models of different genus or models with a different number of disjoint components are used.

Dynamic models can be created directly in space-time or higher dimensions. Each frame of the animation is a cross section of the object taken at a discrete value along the dimension considered as time. Bowyer and Woodwark state, "The things that happen in time – either physically or algebraically – are not equivalent to things happening in an additional spatial dimension. To be more precise, temporal equations that are useful are rarely symmetrical between x, y, z and time; and those that are symmetrical are rarely useful" [3]. While it is true that time and geometric coordinates are rarely symmetrical, there are problems where such uniform treatment of space-time coordinates is useful. We are able to manage the later use of the model by applying our concept of mapping from geometric coordinates to multimedia coordinates. For example, treating space and time together in this way enables us to propose many different models of metamorphosis.

In addition, objects that have been specified using other representations can be readily changed into the functional representation. Conversion algorithms exist for representations that include implicit surfaces, CSG, sweeping, voxel data, and closed parametric surfaces. This allows for access to the large resource of currently existing models that have already been created. Once these objects have been converted, they are then usable with any FRep operation and other FRep models and primitives.

The main disadvantage to FRep is the rendering time. FRep objects can be raytraced, polygonized, or voxelized,
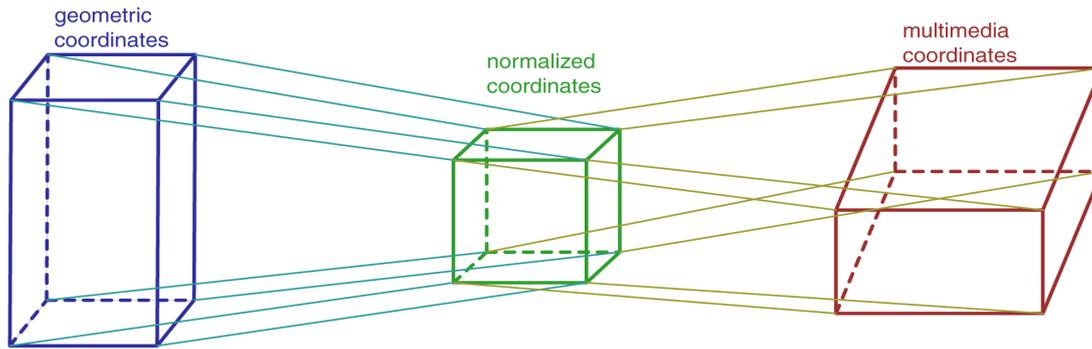
Figure 1: The multidimensional object is first mapped from its geometric coordinates into normalized coordinates. When this is done we have a unit cube of n dimensions with which to map directly to multimedia coordinates.

but this can be time consuming depending on the model's complexity. As computer speeds and memory sizes increase, and with the use of parallel computing, the rendering speed will become less of an issue. There is always the possibility of future advances in rendering algorithms for FRep based objects. Another area that requires further investigation is the direct texturing of FRep objects. This should include the texturing of differing parts of the same object in addition to texturing a dynamic object. Some approaches to this problem have been discussed by Pedersen [5] and Smets-Solanes [10].

3 Multidimensional FRep for Animation

Traditionally in computer graphics and animation, there has been a division between the process of creating an object (modeling) and introducing changes to that object over time (animating). In order to remove the limitations to animating dynamic models imposed by this division, we will consider an animation sequence as a particular type of a multimedia object. A multimedia object can be described using 2D/3D world coordinates, time, color, texture, audio, or other "multimedia coordinates". To generate animation sequences, we propose the mapping of multidimensional FRep objects onto multidimensional spaces of multimedia coordinates.

To operate with multimedia coordinates, one can introduce a system of normalized numerical coordinates (a unit cube) and its one-to-one correspondence to the multimedia space. By selecting a real normalized value, one can use the corresponding value of the multimedia coordinate. The steps in this mapping process are shown in Fig. 1.

The following types of multimedia coordinates with their respective variation intervals can be considered:

- World coordinates of 2D and 3D "real life" geometry. They can be Cartesian, cylindrical, or any other coordinates. The selection of these types and their variation intervals defines the "elementary" image or 2D/3D shape within a bounding box in the selected geometric space. An "elementary" shape (i.e. curve, surface, and isosurface) is a projection of the cross-section of the initial multidimensional object.

- Dynamic coordinates represent continuous values that can be linearly or non-linearly mapped onto physical time. The user can define a path in the space of any dynamic coordinate variables that the animation will trace out in time. This procedure could serve as the basis for implementing a quite complex animation. Each frame of the animation would thus correspond to a cross-section of the multidimensional object.

- Spreadsheet coordinates take discrete values in the given bounding box. This type allows for the spreadsheet-like spatial organization of elementary images or shapes in regularly or irregularly placed 1D, 2D, or 3D nodes.

- Photometric coordinates include color, transparency, texture, and other parameters of visual appearance of the multimedia object. These multiple coordinates can be introduced differently depending on the application. For example, color can be assigned to the shape, background, or light source in the scene.

- Transformation coordinates define transformations (rotation, scaling, filtering, etc.) of the elementary image or object. This allows us to implement "glyphs", which are traditional in visualization.

- Audio/Video coordinates have complex structure and are dynamic by their nature. One needs to synchronize them with other dynamic variables when defining the mapping.

The above list is not complete. Depending on applications and available devices, one could extend it to include other types such as a force feedback coordinate (for haptic interfaces).

Each geometric coordinate variable takes values within a given interval. On the other hand, multimedia coordinates also have their own intervals of variation. To define the mapping, one has to establish correspondence between these intervals. Generally, more than one multimedia coordinate can correspond to one geometric coordinate. For instance, one can map a geometric coordinate simultaneously onto a dynamic coordinate and a color coordinate. This would result in an animation where color changes over time.

To support this approach, we have developed software tools that provide the user a means to:

- Specify a functionally based model in the HyperFun [1] or C languages. HyperFun is a specialized high-level language for FRep object specification with a simplified C type syntax and several additional set-theoretic operators implemented using R-functions [7,8,6]. One can use the library of FRep specific geometric objects and transformations as well as the user's own library written in HyperFun or C.

- Define mappings of object space to multimedia space by assigning "multimedia types" to object coordinates. At least one object space coordinate has to be assigned the dynamic type to generate an animation sequence.

- Generate images of polygonized or ray-traced elementary shapes, animation sequences, or spreadsheets in accordance with assigned multimedia types.

The main two tools developed during this work are for the visualization of models written in the HyperFun language. The tools are the HyperFun Polygonizer for the conversion to polygonal BRep, and HyperFun plug-in to a ray-tracer POVRay. Both of these tools are available on the HyperFun Project site located at the following URL: http://www.hyperfun.org.

4 Case Study: Animation "Homotopic Fun in 5D Space"

As noted above, FRep is by nature multidimensional, and can handle shapes in any dimension that the user sees fit to design. We will present here the animation "Homotopic Fun in 5D Space" which presents a bi-directional metamorphosis (a non-traditional transformation in computer graphics and animation).

This project was started with the selection of key models that would be used in the animation. The four key models are examined below.

Figure 2: *Cat*

- *Cat* (Fig. 2) resembles a cult character of children's animation in Japan. Its complete model can be found on the web at www.hyperfun.org. This model was created using standard primitives and operations available within HyperFun. The *Cat* object is topologically equivalent to a sphere.
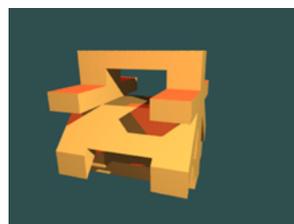

Figure 3: *NiHon*

- *NiHon* (Fig. 3) is a 3D puzzle representing the word for "Japan" in Japanese. First, the two 3D Japanese characters "Ni" and "Hon" are constructed independently as unions of blocks. Then, the solids are oriented along the Z and X-axes respectively. They are then combined as *NiHon* = Ni & Hon, where & represents the R-function based intersection operation. The resulting 3D solid looks like the single character "Ni" or "Hon" when projected onto a plane along the Z and X-axes respectively. The *NiHon* object is topologically equivalent to a sphere with 2 handles.

Figure 4: *Robot*

- *Robot* (Fig. 4) was created using standard FRep primitives and operations available within HyperFun. The *Robot* object is topologically equivalent to a sphere.

Figure 5:   *Rob_let*

- *Rob_let* (Fig. 5) was created using standard FRep primitives and operations available within HyperFun. The *Rob_let* object is a set of five disjoint components. Two of the components are topological spheres, and the other three components are topological spheres with one handle.

At this point, we make a jump up two dimensions in order to create a 5D object that will, in effect, blend these four objects. This blend along with the appropriate mappings will allow us to perform a unique type of metamorphosis in our animation. For this new 5D object we will use a simple bi-linear interpolation between the four 3D key objects listed above. We will define the object *Meta5D* so that:

$$Meta5D(x_1, x_2, x_3, 0, 0) = Cat$$
$$Meta5D(x_1, x_2, x_3, 1, 0) = Robot$$
$$Meta5D(x_1, x_2, x_3, 0, 1) = NiHon$$
$$Meta5D(x_1, x_2, x_3, 1, 1) = Rob\_let$$

Anything between these four values on the (x_4, x_5) plane will be intermediate cross sections consisting of a combination of the four surrounding functions. The object *Meta5D* is defined as:

$$Meta5D(x_1, x_2, x_3, x_4, x_5) =$$
$$\begin{pmatrix} Cat(x_1, x_2, x_3) \cdot (1 - x_4) \\ + Robot(x_1, x_2, x_3) \cdot x_4 \end{pmatrix} \cdot (1 - x_5)$$
$$+ \begin{pmatrix} NiHon(x_1, x_2, x_3) \cdot (1 - x_4) \\ + Rob\_let(x_1, x_2, x_3) \cdot x_4 \end{pmatrix} \cdot x_5$$

This results in a single 5D object described by a single function of five variables. Algebraically, the model of the metamorphosis is the bilinear interpolation between four real-valued functions by coordinates x_4 and x_5. Geometrically, it is a 5D object defined by the real function as:

$$Meta5D(x_1, x_2, x_3, x_4, x_5) \geq 0$$

Another way that the metamorphosis can be viewed is as a homotopy in the functional space, which is reflected in the title of the final animation.

A 2D spreadsheet of 3D cross sections is the most adequate static visual representation of this 5D object. We will define the following mapping for a spreadsheet:

$$\mathbf{x_1} \rightarrow \mathbf{x} \text{ (world x)}$$
$$\mathbf{x_2} \rightarrow \mathbf{y} \text{ (world y)}$$
$$\mathbf{x_3} \rightarrow \mathbf{z} \text{ (world z)}$$
$$\mathbf{x_4} \rightarrow \mathbf{u} \text{ (spreadsheet u)}$$
$$\mathbf{x_5} \rightarrow \mathbf{v} \text{ (spreadsheet v)}$$

The four key objects are placed in the corners of the 5 ~5-spreadsheet pattern with the u and v spreadsheet coordinates varying in the [0,1] interval using a step value of 0.25. For example, the (0.5,0.5) cell contains the image of the equally weighted mixture of all four objects. The spreadsheet can be considered as a set of frames of an animation with two dynamic variables. It can also serve as a reference for constructing an animation sequence containing particularly interesting shape transformations. The spreadsheet is displayed in Fig. 6.
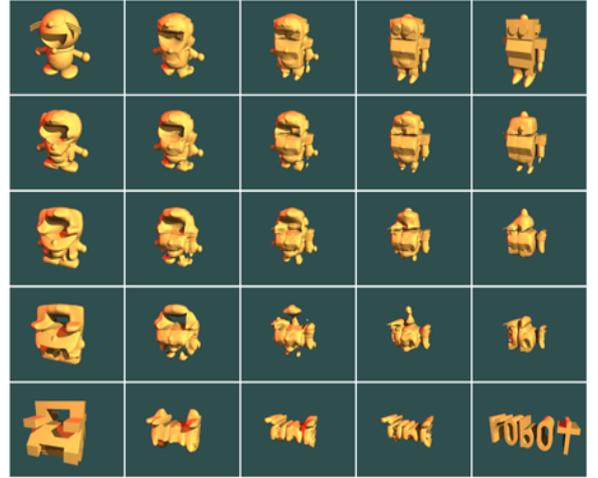

Figure 6:   **A 5 ~5 spreadsheet of *Meta5D* with the u coordinate starting from left to right, and the v coordinate starting from top to bottom.**

To create the final animation we used the following mapping:

$$\mathbf{x_1} \rightarrow \mathbf{x} \text{ (world x)}$$
$$\mathbf{x_2} \rightarrow \mathbf{y} \text{ (world y)}$$
$$\mathbf{x_3} \rightarrow \mathbf{z} \text{ (world z)}$$
$$\mathbf{x_4} \rightarrow \mathbf{t_1} \text{ (dynamic)}$$
$$\mathbf{x_5} \rightarrow \mathbf{t_2} \text{ (dynamic)}$$

This mapping assigns dynamic types to coordinates x_4 and x_5. Each frame of the animation is a rendered 3D object corresponding to a specific point in the (x_4, x_5) plane. The animation sequence corresponds to a curve in (t_1, t_2) space. The spreadsheet helps to introduce such a curve, and then to select points on the curve for the animation frames. The curve used in the animation is approximated in Fig. 7. In the final animation, the object traces out the same path on top of a checkerboard. This serves as an aid to the viewer to understand where the 3D cross-section currently displayed in the (x_4, x_5) plane (see frames in Fig. 8).
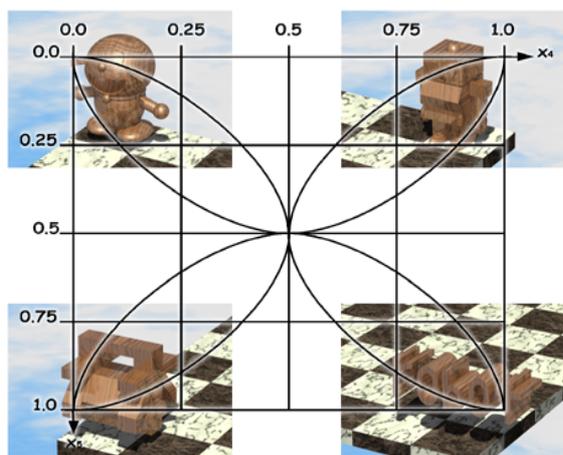

Figure 7: **The approximate curve in the x4-x5 plane that is traced out in the animation.**
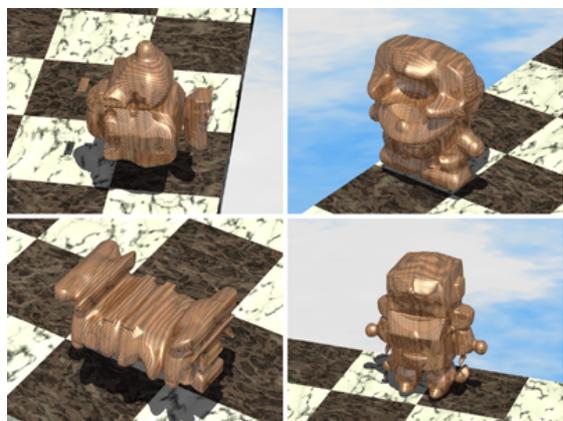

Figure 8: **Intermediate frames of the animation.**

5 Future Work and Conclusions

In this work, we proposed a method of direct space-time modeling to create dynamic objects. We used the function representation in this process due to its multidimensional capabilities. In addition to multidimensionality, FRep has other advantages such as extensibility and operations that are closed on the representation. After creating a higher dimensional object in FRep, we must map its various coordinate variables to multimedia space. In the case of animation, at least one coordinate variable must be mapped to a dynamic coordinate (time).

A final complete example of the animation entitled "Homotopic Fun in 5D Space" was then presented. In this animation, we see the automatic metamorphosis between several 3D objects of arbitrary topology. The presented example is an illustration of the power offered by such an approach. Instead of using simple linear interpolation, splines could be used to define a metamorphosis. A combination of different interpolation schemes and a thoughtful weighting of key objects both locally and globally would allow the user much more control over the aesthetic properties in such an animation. Another approach is to use the so-called combined mappings [9] to define such a transformation. The fleshing-out of these core ideas is left for future work.

References

[1] V. Adzhiev, R. Cartwright, E. Fausett, A. Ossipov, A. Pasko, V. Savchenko, "HyperFun project: A framework for collaborative multidimensional F-rep modeling," *Proc. of the Implicit Surfaces '99 EUROGRAPHICS/ACM SIGGRAPH Workshop*, Bordeaux, France, 1999, pp. 59-69. http://www.hyperfun.org

[2] F. Aubert, D. Bechmann, "Animation by deformation of space-time objects," *Proc. EUROGRAPHICS '97, Computer Graphics Forum*, vol. 16, pp. 57-65, 1997.

[3] A. Bowyer, J. Woodwark, *Introduction to Computing with Geometry*, Information Geometers, 1993.

[4] F. Lazarus, A. Verroust, "Three-dimensional metamorphosis: a survey," *The Visual Computer*, vol. 14, pp. 373-389, 1998.

[5] H. Pedersen, "Decorating implicit surfaces," *Proc. SIGGRAPH '95*, pp. 291-300, 1995.

[6] A. Pasko, V. Adzhiev, A. Sourin, V. Savchenko, "Function representation in geometric modeling: concepts, implementation and applications," *The Visual Computer*, vol. 11, pp. 429-446, 1995.

[7] V. Rvachev, *Methods of Logic Algebra in Mathematical Physics*, Naukova Dumka, 1974 (in Russian).

[8] V. Shapiro, "Real functions for representation of rigid solids," *Computer-Aided Geometric Design*, vol. 11, pp. 153-175, 1994.

[9] V. Savchenko, A. Pasko, "Transformation of functionally defined shapes by extended space mappings," *The Visual Computer*, vol. 14, pp. 257-270, 1998.

[10] J.-P. Smets-Solanes, "Vector field based texture mapping of animated implicit objects," *Proc. EUROGRAPHICS '96, Computer Graphics Forum*, vol. 15, pp. 289-300, 1996.