Technical Report TR-NCCA-2008-02

A WEB ORIENTED FUNCTION-BASED VOLUME MODELING FRAMEWORK

Turlif Vilbrandt, Oleg Fryazinov, Cherie Stamm, Alexander Pasko

The National Centre for Computer Animation
Bournemouth Media School
Bournemouth University
Talbot Campus,
Poole, Dorset BH12 5BB
United Kingdom

2008

Technical Report TR-NCCA-2008-02

ISBN: 1-85899-123-4

Title: A web oriented function-based volume modeling framework

Authors: Turlif Vilbrandt, Oleg Fryazinov, Cherie Stamm, Alexander Pasko

Key words and Phrases: function representation, volumetric modeling, HyperFun, Firefox, XML, HTML

Abstract:

The paper presents a web-oriented function-based modeling framework utilizing a high-level volume modeling language for defining 3D point sets and their volumetric attributes. We illustrate how function-based modeling allows for advanced control and manipulation of 3D models and associated attributes while at the same time allowing for simplified interfaces and parametrization that is difficult or impossible to achieve in other systems. This and other properties of function-based modeling allows for the creation of modeling interfaces suitable for the casual and novice user on the web.

Report date: 13 October, 2008

Web site to download from: URL: http://eprints.ncca.bournemouth.ac.uk

The authors' e-mail addresses: turlif@turlif.org, ofryazinov@bournemouth.ac.uk, pasko@acm.org

Supplementary Notes:

The National Centre for Computer Animation
Bournemouth Media School
Bournemouth University
Talbot Campus,
Poole, Dorset BH12 5BB
United Kingdom

Introduction

Current 3D tools on the web are typically targeted for various kinds of interactive visualization. Few are designed as modeling platforms and even fewer are designed as easy to use web friendly modeling platforms that interact with interfaces constructed in HTML and Javascript. This is in large part due to the difficulty that typical frameworks and representations present for the construction of lightweight, accurate and easy to use interfaces. Function-based modeling offers a useful alternative as a low bandwidth, streamlined and uniform approach.

Function-based modeling or geometric modeling using Function Representation (FRep) [1] defines a geometric object by a single, continuous, real function of point coordinates as $F(X) \geq 0$. Simple yet powerful, FRep can represent objects that are hard or impossible to currently model while also able to represent or incorporate legacy modeling data and approaches. Thus, detailed procedural models, free-form organic shapes, and mechanical parts with sharp edges can be described with relative ease by real functions with underlying constructive tree data structures. An extension of FRep to a volumetric representation was proposed in [2]. A volumetric object is represented by a vector-function, with the first component describing geometry and other components being numerical attributes representing point-wise object attributes or properties such as material, color, transparency, and any others. This approach provides a uniform framework where shape models and volumetric attributes of a model are treated in a similar manner.

Function-based shape modeling can help to not only streamline the representational framework but the user experience and interface as well. Various underlying shape modeling representations and paradigms have a direct effect on the modeling process, including what kind of models can be generated and even the understanding of the objects being modeled. Underlying representations also effect or set limits on the modeling user interface and thus how, why and even what kind of users model. FRep's novel approach presents an opportunity for very simple and powerful interfaces to be created for a much broader range of applications and end users than are typically considered, including those that would normally have no knowledge, skill or interest in shape modeling.

In this article we introduce a web-based FRep modeling framework, Hyperfox, utilizing the popular free and open source web browser, Firefox and HyperFun, which is a high-level programming language for defining parameterized function-based volumetric objects. We describe and illustrate our current results and software, and relate future development directions.

Existing work

Model parameterization

Defining a shape model with variable parameters (parametric family of shapes), such that each new set of parameter values corresponds to another valid instance of a shape, is one of the ultimate goals of the shape modelling research and development. The model parameterization is essential in interactive modelling based on user-modifiable definitions [3]; in the automatic objects reconstruction from scanned point samples using laser or tomography scanning techniques [4]; and in the shape optimization satisfying some design criteria [5]. There are different approaches to the shape parameterization problem strongly connected with specific types of involved shape models.

In the boundary representation (BRep) solid objects are modeled by defining their piecewise boundaries using polygonal meshes or free-form parametric surfaces. For BReps, parameterization is not well defined and limited [3]. For example, objects with different topology cannot belong to a parametric family of BRep solids. In spite of the limitations [6], modern commercial systems support parametric solid modelling based on BRep.

Constructive models such as obtained using Constructive Solid Geometry (CSG) allow for more flexible parameterization supporting to some extent changing topology of the object [3][7]. However, parameterization with smooth transition between drastically different shapes is problematic. The Function Representation (FRep) [1] offers a higher level of parameterization of shape models. Depending on the parameter variations, objects can completely change their geometry and topology.

Web-based shape modeling

Increasing development and use of the Internet and web is accompanied by a many applications for visualization and interaction and some applications for web-based shape modeling. However, most of the formats for these applications are almost exclusively based on polygonal and parametric surface representations. The data size for accurate or detailed models is often very large. This can make the transmission of data between various systems very complicated or sometimes impossible without some loss to the model. In addition most of these tools do not target modeling but interactive visualization only and the web tools that do provide modeling are typically some combination of proprietary, expensive, require special expertise to use and consume large amount of bandwidth. Furthermore, very few or none of these modeling systems are designed to drive interfaces utilizing standard web technologies like HTML and Javascript.

The Virtual Reality Modeling Language (VRML)[8] is an open standard for representing 3D models on the web. In this format geometric information is stored with scene information, such as light sources, textures, colours etc. The XML-based open standard X3D [9], the successor of VRML, allows extensions for various purposes. One of the X3D profiles fully supports the VRML97 specification. However, VRML as well as X3D supports only polygonal representation of the shapes and offer only a restricted set of pre-defined primitives. Additionally, VRML and X3D representations of shapes are usually very large because of the nature of polygonal objects.

Recently, function-based modeling was proposed for these formats in the form of extensions. Thus, in [10] was the first attempt at extending VRML with implicit models. However, only skeletal implicit surfaces were used in the extension. Soon after, VRML nodes with support for HyperFun [11] was created [12] and later an extension of the VRML language with support for FRep was built [13]. Recently an extension for X3D language, FX3D, was proposed in [14] where functional-based nodes were added to the language specification. In these nodes only analytical functions are used, as well as a restricted set of pre-defined functionally-based primitives, which does not allow working with procedural models, such as fractals. Moreover, both the VRML and X3D function-based extensions require a specific non-standard 3D web viewer designed for polygon meshes that makes using these extensions difficult and modification of the framework and/or extensions is nearly impossible as both are strictly proprietary, closed systems. Volume modeling for web-based systems was also proposed mostly for VRML and X3D formats. Thus, in [15] the polygonization are used to represent volume model in VRML, and in [16] an extension for VRML was proposed. Both methods inherit disadvantages of VRML's extensions already discussed.

A web "applet" built using Java and Java 3D has been developed to visualize functionally represented models. In [17] this framework was introduced as a collaborative web-based shape modeling system. The main disadvantage of this approach was the application's efficiency due to the speed of the Java language interpreter when compared with speed of native C/C++ applications. In addition the Java language and particularly Java extensions like Java 3D has proven to be very difficult for users to install and even to have serious compatibility issues across systems and versions of Java.

System overview

HyperFun

In this work we use the HyperFun programming language [11] that supports all the notions of FRep. This language is a simple and yet powerful tool that allows non-specialists to create complex parameterized shape models and their volumetric attributes. The language includes assignment, conditional and iterative statements, standard mathematical functions as well as special build-in operators and standardized library functions. The "FRep library" contains the most common primitives, geometric and attribute operations. The object can be represented by a real-valued function with real-valued arguments. The example below describes a union of a block and a sphere with a pseudo-random distribution of three atrributes throughout the volume:

```
my_model( x[3], a[1], s[3])
{
array center[3];
center = [-1,1,-1];
block = hfBlock(x,center,2,2,2);
noise = hfA_NoiseP(x);
s[1] = 0.5*(1+noise);
s[2] = 0.5*(1+noise);
s[3] = 0.5*(1+noise);
my_model = (9 - x[1]* x[1] - x[2]* x[2] - x[3]* x[3]) | block;
}
```

The HyperFun programming language is a part of an online international free and open source software (FOSS) project (`http://www.HyperFun.org`) utilizing FRep for modeling, visualization, animation and fabrication.

Web based FRep modeling architecture

Unlike most 3D web technologies our architecture's primary focus is not towards sophisticated visualization and virtual online environments but simple, robust and powerful modeling tools. We designed our architecture to operate both as in client-server and standalone environments. The architecture can not only access content on a server but utilize local content as well. In this way the architecture can be seen as more than a web based system as no server is necessarily required.

By definition of the HyperFun format, the model is defined by a real continuous vector-function in the HyperFun language as plain text and can be stored in a plain text file with the .HF file name extension. These files are very compact normally less than 100 kilobytes even for complex models. At present there are several tools that are freely available for download from the HyperFun Project Web site that allow a user to work with HyperFun files. A model can be download from a site, authored in another tool and loaded into the web browser or authored directly in the web browser.

Depending on the application at diffent stages of the modeling process it is nessary to visulaize the obtained volumetric model. This is done by parsing the model, calculating the function, and finally interactive rendering. At present, for visualization our architecture utilizes approximation of the surface by a set of polygons (or polygonization) with color attributes assigned to their vertices. This method is generally not robust, requiring the user to wait while generating the mesh and utilizing large amounts of system memory. To speed up the calulation of the function, a complied libary of common objects has been created.

Implementation and examples

Hyperfox

To demonstrate various possibilities of FRep modeling for the web a plugin has been created for the popular FOSS web browser, Firefox based on the HyperFun language and tools. The plugin is built using ANSI C and C++ and currently is compiled in various forms for both Windows and Linux OS. It uses Gecko SDK to provide an OpenGL window for the visualization of HyperFun models on a web page inside Firefox (see Figure 1a). It is a fully registered plugin with Firefox and assigned to handle HyperFun mime types.

In addition and most importantly, it has a variety of internal methods exposed to the web browser via the Javascript language allowing for (at the moment, limited) interaction between the browser and Hyperfox. The example of scripting the HyperFun plugin is below:

```
embed.setBoundingBox('-10', '10', '-20', '0.0', '-10.0', '10.0');
var oXgrid = document.getElementById("gridx");
if (!embed.isInt(oXgrid.value)) {
  alert("Please correct value for GridX"); return; }
embed.setGrid(oXgrid.value, '30', '30');
embed.setLineColor('#00ff00');
embed.setFaceColor('#FFFFCC');
embed.setBackground('#ffffff');
embed.setSource(oTextbox1.value);
```

Being scriptable, the plugin allows user to change parameters for the model and for the visualization window (see Figure 1b). Rendering in the browser window takes place by using the HTML Embed node. Inside a webpage there is no default interface displayed by the HyperFun plugin, instead allowing anyone to author their own web based interfaces and modeling applications that communicate with the HyperFun plugin through defined interfaces.

In addition to modeling in a webpage, a default user interface can be found in the tools menu of Firefox. This interface provides a local text only modeler built with the Firefox's native interface language, XUL, that allows for the ability to create custom interface inside the Firefox browser. The user interface for the modeler is shown in Figure 2. It provides the user with all the standard tools found in other HyperFun based modeling tools. The interface allows users to define a model in the HyperFun language, set various parameters for the generation of polygonal meshes, render the model for visualization and to execute operations such as loading and saving data. Inside the XUL interface, Hyperfox plugin's visualization window is embedded and interactively renders HyperFun models using OpenGL.

The HyperFun plugin can be installed very simply by visiting `http://HyperFun.org/HF_tools.html` with the latest version of the Firefox browser which can also be found on the web at `http://www.mozilla.com/firefox/`.

Rapidly building interfaces for customized 3D modeling applications

In addition to the default interface, the Hyperfox plugin provides a web author the ability to embed interactive HyperFun visualizations in a web page and rapidly produce simple interfaces for advanced parametrized object modeling and customized web applications. Interfaces utilize the basic functionality offered by HTML/XML and Javascript technologies which are well known and easy to work with.

We have created a simple web application as an example to demonstrate some of the possible ways in which Hyperfox can be used on the web. This demonstration is available to use and can be found on the web at `http://www.makething.org`. In our example the Hyperfox plugin is embedded directly on the web page, and is passed the HyperFun file to process and display. This HyperFun file is wrapped inside an XML file that contains added information specific for each model's parameters, bounding box and required rendering resolution. It defines which parameters have been made available by the model for display within the web interface, the default values of these parameters and minimum, maximum limits of these parameters and the desired increments to be displayed (see Figure 3).

Client side Javascript functions parse through these parameters and dynamically build a set of web forms for user interaction. When the parameters on the web page are changed (and a button to view the cahnges is selected), Javascript grabs the changes, replaces the appropriate values in the HyperFun model, and pass this file back to Hyperfox, which then renders the modified model to the screen.

Due to the use of FRep and the HyperFun language, very complex modifications can be made available to a user even in such a simple interface. In a very simple example with a model of a coffee cup, two completely different function-based models - one a cup and the other a mug - can undergo metamorphoses between one another to design new cups of users choosing. There is no need to have a model with an identical number of mapped vectors, nor does a user need to understand or manipulate complex transformations of vectors. A user can simply pick a value or even choose a position on a slider bar that indicates a cup more like one or the other (see Figure 4). The cup model also has a constructively modeled, volumetric texture. Space partitions for the texture are created using two nested loops for a union of spherical structures with added solid noise. It is important to note that these volumetric attributes or colors are a uniform part of the same HyperFun model that defines the shape of the cup and can be controlled and parametrized in a similar fashion as the shape of the object. In addition the small size and exact accuracy of HyperFun models makes it an ideal platform for simple web based modeling applications. With this interface, it is now possible to rapidly modify parameters of models that would be impossible or very difficult to model with X3D or other traditional 3D web technologies.

Conclusion and future work

In this paper we presented a framework for web-based modeling of functionally defined volumetric objects. By using the HyperFun modeling language and the extensions mechanism of the Firefox browser we introduced the power of FRep modeling systems inside a web browser. However, our modeling system has some limitations:

- The polygonization of FRep models can take long time, forcing the user to wait. The weakest part for all modeling systems based on FRep is the speed to quality ratio of the polygonization. The more precise visualization requested, the more time it will take to generate. This problem could be solved by using several techniques, for example by utilizing GPUs for computation or by ray-tracing [18], which is a precise method to visualize functionally defined models, but it is more computationally expensive to perform in real time.

- At the moment, the Hyperfox framework only works with the free and open-source browser, Firefox. While this frame work is free for anyone to download and use (and even modify), it can create limitations to the number possible users for online applications.

- The simple interactive 3D interface (visulation only) for the creation of new HyperFun models is not interactive enough at this moment.

Even with these limitations FRep is a useful framework as the bases for dynamic, simplified and web-based modeling platforms. Our future work is aimed at removing the existing limitations and further developing function-based volume modeling frameworks.

References

[1] A. Pasko, V. Adzhiev, A. Sourin, and V. Savchenko. Function representation in geometric modeling: concepts, implementation and applications. *The Visual Computer*, 11(8):429–446, 1995.

[2] A. Pasko, V. Adzhiev, B. Schmitt, and C. Schlick. Constructive hypervolume modeling. *Graph. Models*, 63(6):413–442, 2001.

[3] V. Shapiro and D. L. Vossler. What is a parametric family of solids? In *SMA '95: Proceedings of the third ACM symposium on Solid modeling and applications*, pages 43–54, 1995.

[4] P.-A. Fayolle, A. Pasko, E. Kartasheva, C. Rosenberger, and C. Toinard. Automation of the volumetric models construction. In *Heterogeneous Objects Modelling and Applications, Lecture Notes in Computer Science, vol. 4889*, pages 214–238. Springer Verlag, April 2008.

[5] J. Chen, V. Shapiro, K. Suresh, and I. Tsukanov. Shape optimization with topological changes and parametric control. *International Journal for Numerical Methods in Engineering*, 71(3):313–346, December 2006.

[6] S. Raghothama and V. Shapiro. Boundary representation deformation in parametric solid modeling. *ACM Trans. Graph.*, 17(4):259–286, 1998.

[7] A. Bowyer, D. Eisenthal, D. Pidcock, and K. Wise. Configurations, constraints and CSG. *International Journal of Shape Modelling*, 6(2):131–143, 2000.

[8] R. Carey and G. Bell. *The annotated VRML 2.0 reference manual*. Addison-Wesley Longman Ltd., 1997.

[9] X3D - extensible 3D, New-Generation Open Web3D Standard. `http://www.web3d.org/x3d/`.

[10] B. Wyvill and A. Guy. Implicit modelling and VRML. In *Virtual Environments on Internet, WWW, and Networks*. British Computer Society, April 1997.

[11] V. Adzhiev, R. Catwright, E. Fausett, A. Ossipov, A. Pasko, and V. Savchenko. HyperFun project: Language and software tools for F-rep shape modeling. *Computer Graphics and Geometry*, 1(2):75–100, 1999.

[12] J. Zara. HyperFun project and VRML language. `http://www.cgg.cvut.cz/~zara/HyperFun/`, 2001.

[13] L. F. Min, A. Sourin, and K. Levinski. Function-based 3D web visualization. In *CW '02: Proceedings of the First International Symposium on Cyber Worlds (CW'02)*, page 0428. IEEE Computer Society, 2002.

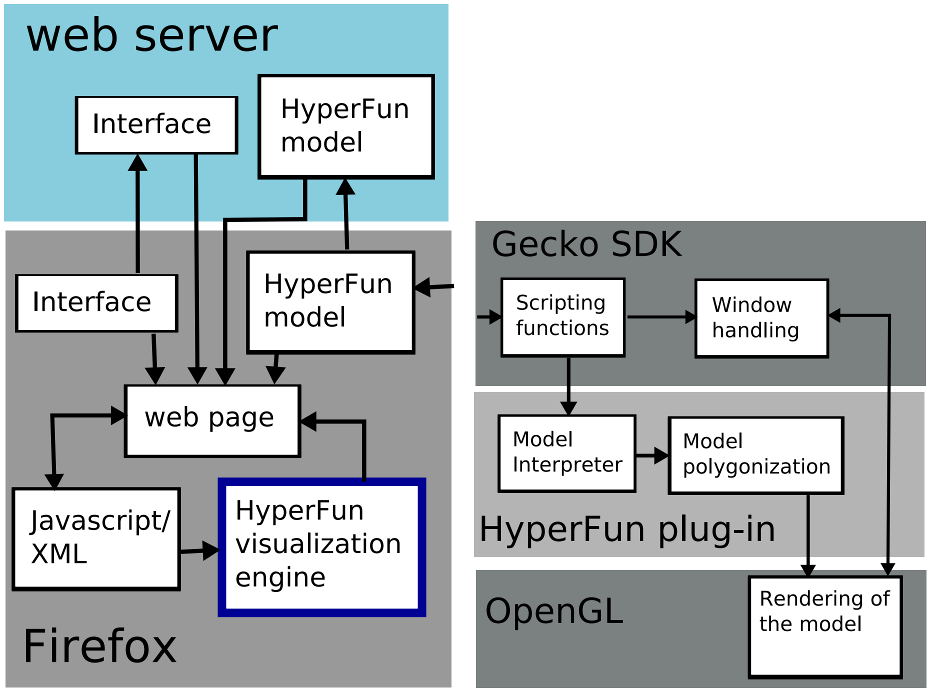
[14] Qi Liu and Alexei Sourin. Function-based shape modeling and visualization in X3D. In *Web3D '06: Proceedings of the eleventh international conference on 3D web technology*, pages 131–141. ACM, 2006.

[15] K. D. Engel, R. Westermann, and T. Ertl. Isosurface extraction techniques for web-based volume visualization. In *IEEE Visualization '99*, pages 139–146, 1999.

[16] J. Behr and M. Alexa. Volume visualization in VRML. In *Web3D*, pages 23–27, 2001.

[17] R. Cartwright, V. Adzhiev, A. Pasko, Y. Goto, and T. L. Kunii. Web-based shape modeling with Hyperfun. *IEEE Comput. Graph. Appl.*, 25(2):60–69, 2005.

[18] O. Fryazinov and A. Pasko. Interactive ray shading of frep objects. In *WSCG'2008, Communications Papers proceedings*, pages 145–152, 2008.



(a) (b)

Figure 1: a) Server-browser interface for Hyperfox architecture, b) Visualization engine in Hyperfox

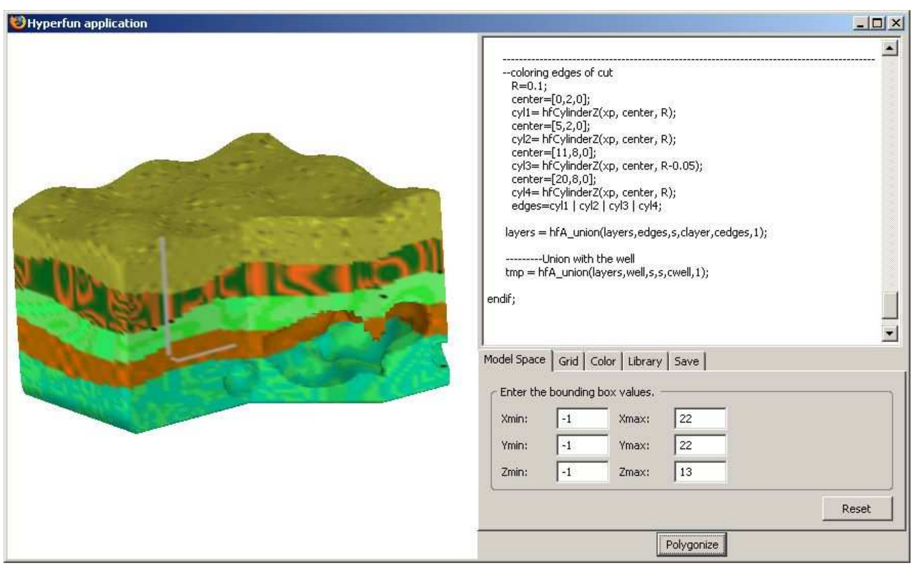


Figure 2: Hyperfox extension's visualization window with volumetric geological model with irregular material distribution, internal cavities, and oil well.

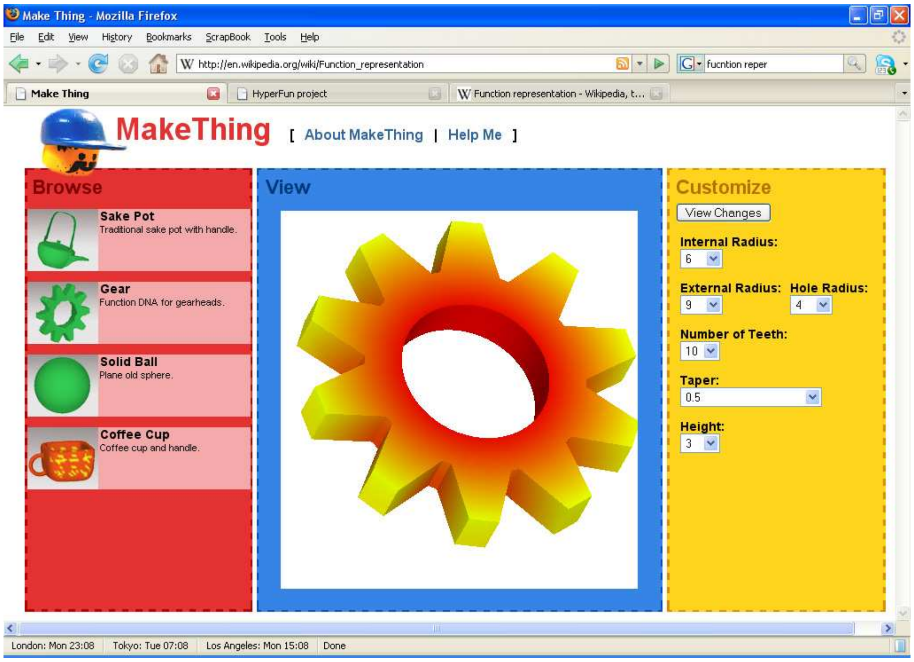


Figure 3: Web interface for Hyperfox extension

Figure 4: Coffee cup at different stages of metamorphosis