International Conference
FROM DELC TO VELSPACE
Plovdiv, 26–28 March 2014

F-REP DESIGNER

Alexander Penev

Abstract. *The paper presents a computer application for 3D geometric modelling based on Function representation scheme (F-Rep). This application called F-Rep Designer uses a platform independent hybrid CPU/GPGPU implementation of Ray tracing algorithm for interactive visualization of the 3D scene during modelling by users. One of the major features of the current implementation of the system is that it is entirely mesh-free (in model, visualization, and user interaction). We also discuss existing and possible applications of the F-Rep Designer in areas such as research, education and creation of e-learning content.*

Keywords: computer graphics, F-Rep, mesh-free, modelling, education
Mathematics Subject Classification 2010: 68U05, 68U07, 97R60

1. INTRODUCTION

Computer graphics has many applications to various fields of modern life. Geometric modelling systems (GMS) have been developing for years and nowadays there are many systems with different capabilities and application areas.

Most modern GMS use combinations of the well-known representation schemes Boundary representation (B-Rep), Constructive Solid Geometry (CSG) and others. They can not do well with some of today's user demands. One of their main advantages – the existence of hardware implementation of the visualization – is becoming less important with the development of modern graphics hardware (mainly GPGPU). This opens up the possibility of using more powerful and informationally complete models such as Function representation (F-Rep) [4] scheme for modelling and interactive work with GMS based mainly on such class of representations. Another approach to work around these problems is to add more information to the basic model of the existing systems. This solves the problem to some degree and extends the applicability of these systems, but in practice adds unnecessary complexity. It can be avoided if instead of improving the legacy systems one creates a new system that is unburdened by much of that legacy.

The project F-Rep Designer was created exactly for this purpose. Unlike most other systems (commercial or experimental) that use B-Rep as the main model of scene representation, F-Rep Designer is based only on F-Rep. It has three main objectives: Applying and testing F-Rep; Training and encouraging students to do scientific research; Creating an entirely mesh-free geometric modelling system, freed from the limitations of most GMS.

2. RELATED WORK

In recent years, due to the increasing computational power of computers, more and more GMS are created based on powerful representation schemes. Most of them are research systems, but in the past few years they have inspired several commercial systems that fill some gaps in the market of 3D modelling software. Visualization of the surface of the models only or the presence of a constructive tree is no longer enough. For example, in order to materialise a solid with a 3D printer one needs more than its 3D triangle mesh. On the other hand, the classical representation schemes for volume description are often difficult to the user, do not provide powerful algorithms for transformation and editing, have low quality, take up too much memory, etc.

There is much theoretical and applied research that uses powerful representation schemes for volume description, 3D textures, etc. Most of it is based on modifications of implicit surfaces, signed distance functions, and other similar approaches that apply mathematical real functions for describing the volume of solids. The goal is to use these functions to perform all the classic set-theoretic operations and to expand the set of operations easily performed over the solids.

One of the most well-known systems is HyperFun [5]. It is a GMS based on F-Rep, having its own scene description language (SDL) and scene visualization algorithms. One of the main ideas is to include in F-Rep other representations (homogeneous hybridity) such as Voxels, Implicit surfaces, CSG (by using the so-called R-functions), Sweeping, etc.

In the Master's thesis of the author [9] a system for description and visualization of 3D (and multidimensional) models based on function representation is given. It defines its own object-oriented SDL and even though in subsequent studies, for many reasons, the idea of creating a specialized language was abandoned for the sake of using widespread general-purpose programming languages (such as C#, for example), this is one of the attempts to use a concept close to F-Rep. As in the current work, we use Ray tracing for model visualization instead of polygonization (in [9] an algorithm to visualize the distance field functions is proposed that is analogous to the now well-known “Sphere tracing” [7]).

Another modern system is Symvol for Rhino [6]. It is a commercial plugin relying on F-Rep as a basic concept for the construction and visualization of scenes based on the description of volumes. A constructive tree whose leaves are F-Rep functions is used, and R-functions are used for the constructive operations. There is also a variety of other operations, bounded blending, nonlinear transformations (twist, taper, bend) and others.

Many other commercial plugins for creating FX effects work with internal geometry representations that are more complex than B-Rep, but in most cases they are converted to B-Rep so that the system fits in the host GMS.

3. F-REP BASIC CONCEPTS

The Function representation schemes (F-Rep) are used for describing geometric objects (solids). F-Rep [4] represents a geometric object by a real continuous function *f* defined in an Euclidean space.

A real continuous function *f*, which describes a solid, is defined by:

(1) $$f : X \to R, X \in E^n .$$

This function induces a point set $S_G = \{ X \in E^n \mid f(X) \geq 0 \}$ in E^n .

A special case of the function *f* is when it gives the signed distance from a point X to the surface of the solid S_G. Modelling by using such a function is more restrictive, but it also has some advantages (mainly for developing fast visualization algorithms like Sphere tracing [7], [9]).

There are many operations on objects of F-Rep [4]: set-theoretic, blending, offsetting, Cartesian product, metamorphosis, bijective and linear mapping, projection, etc. A fundamental advantage of F-Rep is its openness (extensibility) in view of the possibility for adding new primitives, operations, and relations. Among its big advantages is also the easy implementation of nonlinear transformations and other complicated operations. For example, the operation metamorphosis (which means morphing one solid into another one) is hard to be implemented in B-Rep. However, in F-Rep it has a simple solution $f(t) \equiv t \cdot f_1 + (1-t) \cdot f_2, t \in [0,1]$, where f_1 and f_2 are function representations of two solids, *t* is a parameter determining the morphing phase.

All geometric operations in F-Rep can be defined analytically. For example, the set-theoretic operations are implemented by using the so-called R-functions [1], [2], [3], [4] (see (2)-(4), for example). The use of the R-functions makes F-Rep more powerful. R-functions [1], [2] are real functions of real variables which inherit some properties of the logical functions (binary or ternary logic). For example, the conjunction is called the logical friend of the R-function

(2) $$f_1 \wedge_a f_2 \equiv \frac{1}{1+a} \cdot \left(f_1 + f_2 - \sqrt{f_1^2 + f_2^2 - 2a \cdot f_1 \cdot f_2} \right).$$

Analogous functions exist for all other set-theoretic operations, for example:

(3) $$f_1 \vee_a f_2 \equiv \frac{1}{1+a} \cdot \left(f_1 + f_2 + \sqrt{f_1^2 + f_2^2 - 2a \cdot f_1 \cdot f_2} \right),$$

(4) $$\neg f \equiv - f \quad .$$

In practice we use the special cases $a = 1$ ($\min(f_1, f_2)$ for conjunction and $\max(f_1, f_2)$ for disjunction) and $a = 0$ ($f_1 + f_2 - \sqrt{f_1^2 + f_2^2}$, $f_1 + f_2 + \sqrt{f_1^2 + f_2^2}$, respectively).

The contribution of R-functions to computer graphics, and F-Rep in particular, is the possibility for composing practical arbitrary solids (functions) based on other simpler and already constructed functions or primitives as spheres, cylinders, cones, etc. In general, F-Rep provides an easy opportunity to incorporate elements from other representation schemes (not only ones from CSG by R-functions, but models can be parametrized easily, etc.) in itself. This means that F-Rep provides an approach for the realization of homogenised hybrid representations. The inclusion of new operations and transformations is also uniform and smooth.

There are many and various algorithms that explore the model in order to obtain its characteristics, mainly visual, i.e. those which can be used for the purpose of visual representation of the model as an image. Visualization algorithms are divided into two classes: polygonization based (for example marching cubes, marching triangles, adaptive polygonization, particle systems polygonization) and Ray tracing based. The first class converts F-Rep into B-Rep, which is visualized by B-Rep approaches (GPU accelerated Z-Buffer algorithm, for example). Polygonization is used not only for visualization but also for other modelling goals, export, etc. The second class makes visualization directly from observer's viewpoint, without intermediate conversion. A basic operation in these algorithms is the intersection of a ray with a model (nearest model point in some direction).

4. F-REP DESIGNER SYSTEM

F-Rep Designer is a prototype of an experimental geometric modelling system based entirely on the F-Rep representation scheme.

According to the goals described in Section 1, we define some requirements. While some of them may be overly restrictive, they are defined so as to avoid most of the problems of the modern GMS caused mainly by the nature of the used representation scheme and the architecture (although in most cases suitable) of the applications.

Requirements to the F-Rep Designer system. The system must be:

- an interactive system for geometric (3D) modelling;
- F-Rep based – using only F-Rep for the scene model;
- Ray tracing based – using only Ray tracing visualization;
- a hybrid system – using CPU/GPGPU simultaneously;
- mesh-free (in model, visualization, and user interaction);
- SDL-free – using neither its own SDL nor other well-known SDL;
- platform independent;
- plugin based, open, and extensible.

The design and development of the system are scheduled to proceed gradually, the main stages being: 1. Research and design; Creation of an initial prototype; 2. Experiments; 3. Development and applications. We have completed the first stage and to a great extend the second stage.

The system architecture is organized similarly to the one in [10]. It is a system based on plugins and numerous interacting services.

Prototype implementation (see Figure 1) uses the following technologies and programming languages: Mono/MS .Net; C#; OpenCL; GTK#.

To easily achieve some of the goals, the system realization is based on OpenF [10], [11] – a framework for creating Open hybrid geometric modelling systems based on non-homogenized hybrid representation systems. At this stage of development there is no need to use the non-homogenized hybridity provided by the framework.

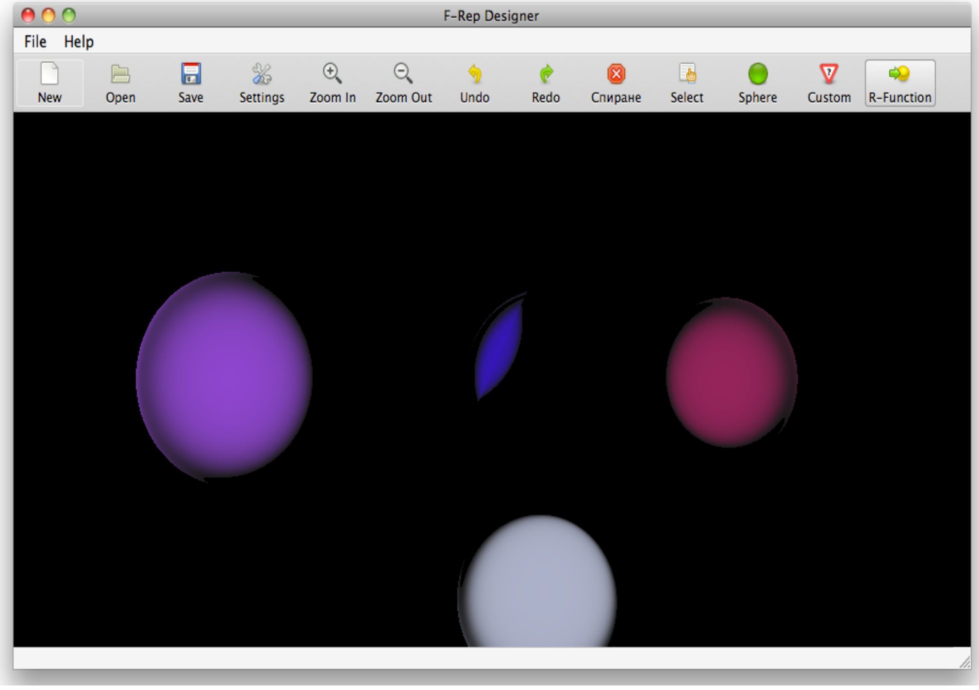

Figure 1. MVC based F-Rep Designer GUI prototype. Model contains three balls and one R-Function based intersection of two balls (in centre of screen)

During the implementation we used object-oriented programming and some classic design patterns, such as Composition, Strategy, and meta architectural templates as MVC, etc.

In F-Rep Designer a dialect of F-Rep is used that we called F-Rep*. The difference is that we use $f(X)\leq 0$ for the interior and the surface of the solids (which imposes a similar change of sign when using R-Function, etc.), and we require that the solids be described by a distance function. For example, to describe a ball we use $x^2+y^2+z^2-r^2\leq 0$. The R-function for intersection is *max*, not *min*.

The model is currently a composition of real functions of real parameters defined analytically. At this stage a constructive tree is not used on purpose, since the composition of functions carries enough information to construct such a tree, if necessary.

We created a simple GUI using GTK# and MVC design pattern. The View part of the system visualizes the model from the user perspective. Visualization is performed by Ray tracing. The calculation of the F-Rep function is performed by compiling functions to IL or by retargeting to OpenCL, compiling and executing on CPU/GPGPU. In the GPGPU implementation the visualization algorithm and the model are compiled together in one. Normal vectors are calculated by automatic differentiating of the code IL (ADIL) of the functions and including the result in the code for compilation. Currently a simple model of illumination is used with multiple point light sources but we are working on the integration of GI rendering system, based on highly modular shaders [13]. An important feature of the implementation is that the visualization of the scene and the user interaction with it occur without polygonization and is fully mesh-free (i.e. B-Rep is not used in any form). This is done in order to simplify the system and to avoid possible problems arising from the B-Rep characteristics. In the future, different techniques and accelerating structures can be used to achieve faster visualisation, user interaction or the ability to maintain very large models.

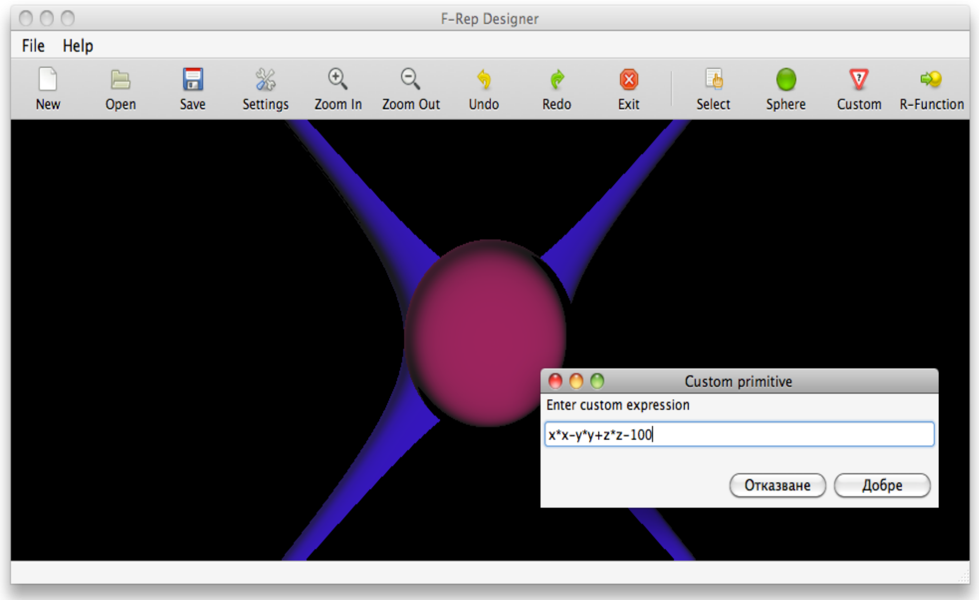

Figure 2. Model with one ball and one custom primitive defined by expression

User interaction is based on the Ray tracing algorithm used for visualization, working in Ray casting mode. This allows selection of elements of the model and their transformation, removal, composition of the model elements by means of R-functions, and so on. The user can perform basic navigation through the scene.

Currently, the prototype has the function to save the models in a file that is performed by binary serialization of the scene object model. However, we expressly put demand on the system not to use any SDL, for the following reasons:

- There is no reason to make users learn one more specialized language;

- Using an existing specialized language (though widespread) can lead to a number of limitations to the system;
- General-purpose programming languages have very large, simple and clear expression means and it is unnecessary to duplicate functionality by repeating them in the specialized language;
- Everything is executable code: the Ray tracer algorithm, the F-Rep model, Normal functions and Illumination model (Shaders).

5. RESULTS AND APPLICATIONS

The main result was the realization of a prototype of the system F-Rep Designer, which has the characteristics and meets the requirements described in the previous section. The main ones are: F-Rep based model; Mesh-free implementation; Hybrid CPU/GPU implementation.

An interesting feature of the implementation is that it uses the following approach: it combines the model (F-Rep, geometry), the functions for calculating the normal vectors, the light model, and the Ray tracer algorithm and compiles them (in C#/IL or OpenCL) to executable CPU/GPU code. This provides some significant advantages. For now, the main difficulty for this approach is that still there are major limitations in the use of OpenCL due to the implementation of large programs, the slow compilation, etc.

We have implemented two Ray tracers available for selection by the user. One is a C# based simple Ray caster using a simple light model. The second is an OpenCL based simple Ray tracer.

The main fields for application of F-Rep Designer are:

- Research – studying the F-Rep and its capabilities when describing spatial scenes, visualization, GPGPU aided modelling and visualization, etc.;
- Education – leading to a more complete implementation of the concepts for university studies in Computer Graphics described in [11] and [12]. The F-Rep Designer was planned as a project in which many students would participate. Until now, its implementation is almost entirely work of students from the Bachelor and Master programs at FMIIT, PU;
- e-Learning content creation – creating of electronic content in mathematics, computer graphics, 3D modelling and animation, parametrized models, etc.

6. CONCLUSION

Our discussion here indicates that the F-Rep Designer is a system with great potential for development and application. Some of the problems and weaknesses in the F-Rep Designer prototype that we became aware of are in the process of removal, but others require the development and improvement of other systems (such as OpenCL) and are not a subject of this study.

Even though we have achieved some important results, for the development and application of F-Rep Designer there is still much to be done. In future we plan to extend and develop F-Rep Designer in several ways:

- To add a third optional Ray tracer by integrating the RayTracer system [13], which has many more options and is written in C#. This C# based Ray tracer will be compiled to .NET MSIL, then it will be decompiled to TAC by one of our other projects (SolidOpt) and will be retargeted to OpenCL, CUDA, and etc. This approach, although much more complex, would provide far more benefits besides the fact that the C# based Ray tracer is more flexible and has more features;
- To use CUDA as an (more mature) alternative to OpenCL;
- To finish the project ADIL and apply it to convert the calculation of the solid's normal vectors into IL, OpenCL, CUDA, etc.

ACKNOWLEDGMENTS

The work is partly funded by Fund Research of Plovdiv University under contract NIS14-FMIIT-002/26.03.2014.

REFERENCES

[1] Rvachev, V., *Geometrical application of logic algebra*, Kiev, Technika, 1967, (In Russian).

[2] Rvachev, V., *R-Function theory and its applications*, Kiev, Naukova Dumka, 1982, (In Russian).

[3] Shapiro, V., *Theory of R-functions and applications: a primer*, Computer Science Technical Reports, TR91-1219, Cornell University, New York, 1991.

[4] Pasko, A., V. Adzhiev, A. Sourin and V. Savchenko, Function representation in geometric modeling: concepts, implementation and applications, *The Visual Computer*, Vol. 11, Issue 8, 1995, 429–446, doi:10.1007/BF02464333.

[5] Adzhiev, V., R. Cartwright, E. Fausett, A. Ossipov, A. Pasko and V. Savchenko, Hyperfun project: a framework for collaborative multidimensional f-rep modeling, *In Implicit Surfaces*, Vol. 99, 1999, 59–69.

[6] Uformia, Symvol for Rhino, http://www.uformiaworld.com/products/symvol-for-rhino, (visited 18 March 2014).

[7] Hart, J., Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces, *Springer-Verlag*, *The Visual Computer*, Vol. 12, Issue 10, (1996-12-01), 527–545, doi:10.1007/s003710050084.

[8] Dimov, D., *Computer graphics*, Plovdiv, PU Publishing, 1999, (In Bulgarian).

[9] Penev, A., *One approach to describe geometric information (MSc thesis)*, Plovdiv, Plovdiv university "Paisii Hilendarski", 1996, (In Bulgarian).

[10] Penev, A., Open hybrid systems for geometric modelling (PhD thesis),

Plovdiv, Plovdiv University “Paisii Hilendarski”, 2013, (In Bulgarian).
[11] Penev, A. and D. Dimov, OpenF applications in Computer graphics education, *Proceedings of the International scientific conference* “*Science, education and time as our concern*”, Smolyan, Bulgaria, 2007, 69–76, (In Bulgarian).
[12] Lesev, H., Photorealistic computer graphics in student training in informatics, *Proceedings of the ARIO-2010*, Plovdiv, 2010, 141–146, (In Bulgarian).
[13] Lesev, H., Optimizing shading process in photorealistic graphic systems, *Proceedings of International scientific conference Informatics in the scientific knowledge*, Varna, 2010, 211–219.

Faculty of Mathematics and Informatics,
University of Plovdiv “Paisii Hilendarski”
236 Bulgaria blvd, Plovdiv, Bulgaria
apenev@uni-plovdiv.bg

F-REP ДИЗАЙНЕР

Александър Пенев

Резюме. *Тази статия представя компютърно приложение за геометрично 3D моделиране, базирано на Функционална представяща схема (F-Rep). Това приложение, наречено F-Rep Дизайнер, използва платформено независима реализация на Ray tracing алгоритъм, базиран на CPU/GPGPU. Той се използва по време на моделиране на 3D сцената за интерактивната й визуализация. Една от основните характеристики на текущата реализация на системата е, че тя е напълно mesh-free (в модела, визуализацията, и взаимодействието с потребителя). Ние също така дискутираме съществуващите и възможните приложения на F-Rep Дизайнер в области като научните изследвания, образованието и създаването на електронно учебно съдържание.*