

Интерактивная система для создания форм на базе функций возмущения

С.И. Вяткин, Б.С. Долговесов
Институт автоматики и электротехники СО РАН,
Новосибирск, Россия
sivser@mail.ru, bsd@iae.nsk.su

Аннотация

Данная статья посвящена интерактивному моделированию функционально заданных форм без предварительной триангуляции. Для этого была разработана инструментальная среда, в которой интерактивно можно создавать или редактировать функционально заданные объекты с применением функций возмущения. Созданное программное обеспечение значительно упрощает создание функционально заданных объектов.

Ключевые слова: Интерактивное моделирование функциональных форм, свободные формы, функции возмущения.

1. ВВЕДЕНИЕ

В последнее время компьютерная графика, связанная с интерактивным моделированием и редактированием трехмерных объектов развивается быстрыми темпами. Известны такие коммерческие системы как SoftImage, MAYA, 3DStudioMax и др. для интерактивного редактирования полигональных моделей. Существуют интерактивные системы на базе polynomials [1], implicits [2], triangle meshes [3, 4], images [5], volumes [6-8], function-based models [9-15]. Одним из перспективных способов задания объектов является функциональное задание. Функциональное представление поверхностей относится к одному из нескольких быстро развивающихся направлений в геометрическом моделировании и компьютерной графике. В ряде работ [16-25] показаны функциональные способы описания визуализируемых объектов, позволяющие существенно, в сравнении с полигональным заданием, сократить объем баз данных для определенного класса объектов. Для этого предлагается определить объект с помощью вещественной непрерывной описывающей функции трех переменных (x_1, x_2, x_3) в виде $F(X) \geq 0$, и рассматривать объекты как замкнутые подмножества Эвклидова пространства E_n , определяемые описывающей функцией $F(X) \geq 0$, где F - непрерывная вещественная функция, и $X = (x_1, x_2, x_3)$ - задаваемая координатными переменными точка в E_n . Здесь $F(X) > 0$ задает точки внутри объекта, $F(X) = 0$ - точки на границе и $F(X) < 0$ - точки, лежащие снаружи и не принадлежащие объекту. Лабораторией синтезирующих систем визуализации было предложено задание объектов с применением функций возмущения [26-29]. До недавнего времени удовлетворительные способы создания функциональных объектов отсутствовали. Объекты описывались вручную, в текстовом виде. Эта операция чрезвычайно трудоемка. Поэтому задача создания инструментальной среды для редактирования функционально заданных объектов с

применением функций возмущения стала очень актуальной. Для достижения поставленной задачи было сделано следующее: сформулированы требования к разрабатываемому программному обеспечению; разработан формат хранения данных в памяти; разработаны и реализованы алгоритмы работы с файлами и алгоритмы построения сцены из памяти; разработан и реализован алгоритм добавления объекта в сцену, в место указанное мышкой; разработан и реализован алгоритм нахождения объектов в сцене, в месте выделенном мышкой; создан интерфейс к программе.

В результате была разработана интерактивная инструментальная среда, которая значительно упрощает процесс создания и редактирования функционально заданных объектов. Для визуализации был использован алгоритм многоуровневого отслеживания лучей [26, 27, 28], который осуществляет эффективный поиск точек поверхностей, участвующих в формировании изображения.

2. ФУНКЦИОНАЛЬНОЕ ЗАДАНИЕ ОБЪЕКТОВ

Дадим краткое определение функций возмущения, описанное в работе [26]. Предложено описание сложных геометрических объектов с помощью задания функций возмущения от базовых поверхностей (рис. 1).

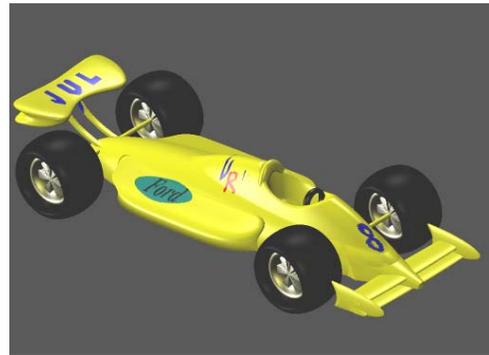


Рис. 1. Функционально заданный объект, требующий в 500 раз меньше информации по сравнению с заданием его полигонами.

Свободная форма есть композиция базовой поверхности и возмущения $F^2(x, y, z) = F(x, y, z) + R(x, y, z)$, где функция возмущения $R(x, y, z)$ находится следующим образом:

$$R(x, y, z) = \begin{cases} Q^2(x, y, z), & \text{при } Q(x, y, z) > 0 \\ 0, & \text{при } Q(x, y, z) \leq 0, \end{cases} \quad (1)$$

где $Q(x, y, z)$ - например, возмущающая квадрака.

Таким образом, задача конструирования объекта сводится к задаче деформации базовой поверхности нужным образом, а не к аппроксимации ее примитивами. Предложенный способ описания объектов трехмерных сцен базовыми поверхностями и функциями возмущения имеет компактное описание, что позволяет уменьшить в 500 и более раз объем передаваемых данных.

3. ВИЗУАЛИЗАЦИЯ

Важной частью интерактивной системы является программа VxDemo, предназначенная для визуализации функционально заданных объектов. На рисунке 2 показано главное окно программы. Входными данными для VxDemo является текстовый файл (с расширением SCN) созданный в каком либо текстовом редакторе. Файл содержит описание сцены. Текст файла сцены представляет собой набор лексем, которые можно условно разделить по следующим категориям: графические примитивы, свойства примитивов, операторы, текстура, параметры освещения.

Выходными данными являются результирующие изображения сцен в графическом формате .

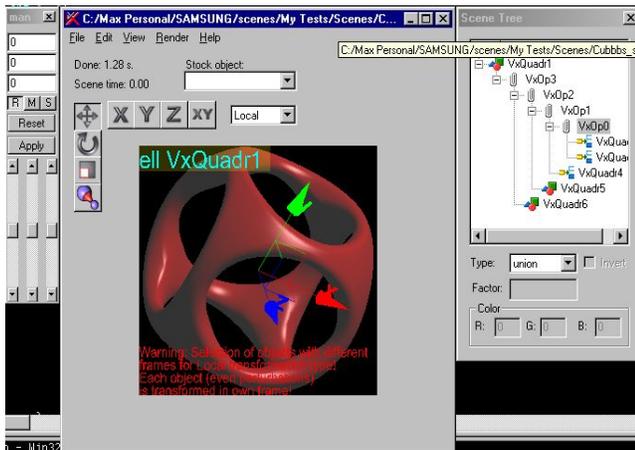


Рис. 2. Главное окно программы VxDemo

Для изменения параметров отображения сцены служит окно Options, которая вызывается через одноименный пункт меню View (рис. 3). В окне Options есть следующие основные редактируемые поля:

Quatro Levels - означает число уровней четверичного деления, т.е. если Quatro Levels = 8, то это означает, что изображение будет иметь разрешение 256 x 256.

Binary Levels – число уровней двоичного деления, т.е. 2^{Binary Levels} – это число разбиений по Z координате.

Filter Level – уровень фильтрации изображения, например, Filter Level = 1 это означает, что цвет в каждом пикселе вычислен путем замешивания цветов четырех субпикселей. Filter Level = 2 – шестнадцати и т. д. (2^{ 2 * Filter Level }).

Perspective Factor – перспективный фактор, если = 1, изображение генерируется без учета перспективного преобразования, значения > 1, - перспективное преобразование учитывается.

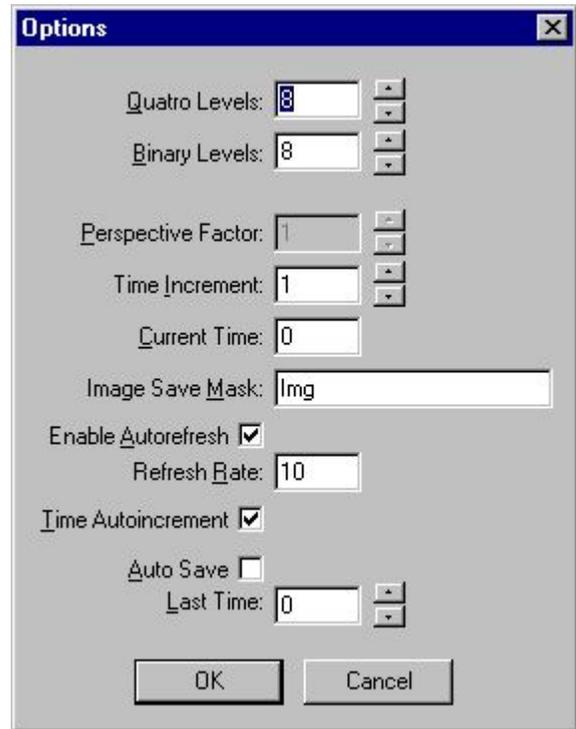


Рис. 3. Окно изменения параметров отображения сцены

Как уже упоминалось выше, для визуализации был использован алгоритм многоуровневого отслеживания лучей [26-28].

4. ИНТЕРАКТИВНОЕ СОЗДАНИЕ ФОРМ

Поскольку конструирование объекта сводится к задаче деформации базовой поверхности нужным образом, а не к аппроксимации ее примитивами, то данный процесс напоминает лепку модели из пластилина с применением геометрических операций, представленных в работе [30]. Для этого необходимо было решить следующие задачи.

- 1) Реакцию программы на события от мыши (и, соответственно, выбор и модифицирование объектов). Необходимо уметь выделять объект (или несколько объектов) в сцене и предоставлять пользователю возможность проделать с ним (с ними) некоторые операции (рис. 4 - рис. 9).
- 2) Выбор операции – аффинные преобразования MRS (move, rotate, scale), геометрические операции [30] или деформация (рис. 10, 11). Деформация состоит в возможности добавлять в любую точку на поверхности возмущение с параметрами задаваемыми *инструментом*. *Инструмент* задает область действия и вид возмущения. Для этого необходимо графически предоставить пользователю некоторую информацию (в том числе выделять объект, например, цветом, или с помощью bounding box, рисовать оси и т.д.). Всю отрисовку «поддержки» работы с операциями лучше производить с помощью OpenGL – для скорости и функциональности.
- 3) Возможность работы со списком инструментов.
- 4) Запись в файл и загрузка дерева сцены.
- 5) Возможность создания новых этегов и их распознавания.

Для этого было сделано следующее. Рендеринг был выделен в отдельный поток. Отдельный поток необходим для того, чтобы при любом изменении сцены пользователем, рендеринг можно было оперативно прервать и перезапустить. Была сделана поддержка рендеринга OpenGL (совместимость буферов, depth-test и т.д.), с формированием bounding box для объектов, с автоматическим переходом на пониженную детальность при редактировании (если это необходимо), с нахождением выбранных объектов сцены.



Рис. 4. Клипирующие плоскости

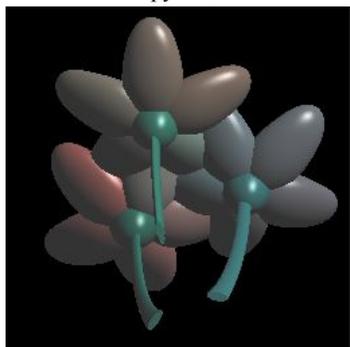


Рис. 5. Начальная сцена

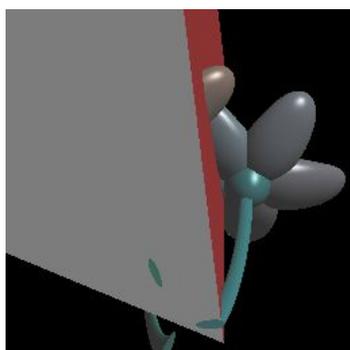


Рис. 6. Объединение начальной сцены с клипирующими плоскостями (клипирующие плоскости предварительно были повернуты).

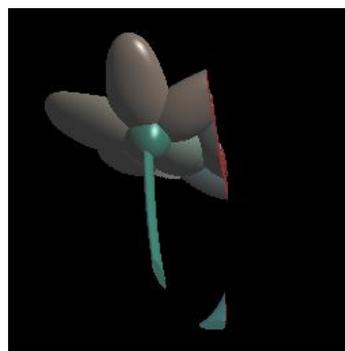


Рис. 7. Клипированная сцена (с помощью операции пересечения)

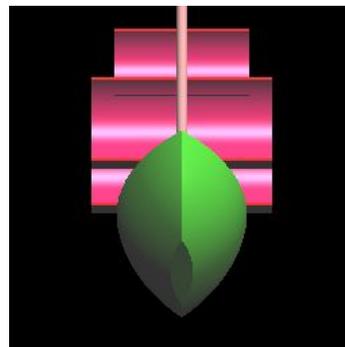


Рис. 8. Начальная сцена перед вставкой

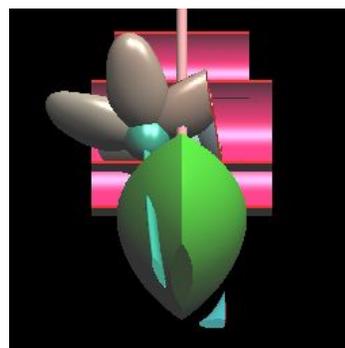


Рис. 9. Вставка в начальную сцену клипированной части

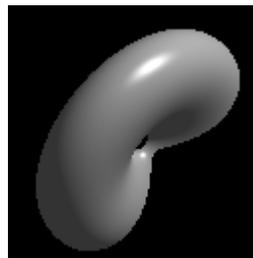


Рис. 10. Деформированный тор

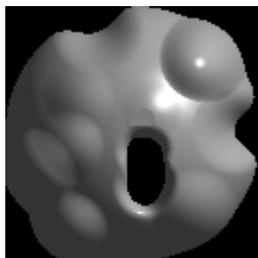


Рис. 11. Объект с положительными (выпуклые части) и отрицательными (вогнутые части) возмущениями

Был создан интерфейс с отрисовкой осей объекта (рис. 2), кнопками выбора операции (рис. 12, 13), с обработкой соответствующих событий от мыши и т.д., с возможностью выделения и работы только с возмущениями (чтобы при деформации менялись существующие возмущения, а не только добавлялись новые). Сцена была представлена в виде дерева с соответствующим контролем в отдельном диалоге, с возможностью копирования, удаления, вырезания, вставки и перетаскивания элементов.

Программная модель алгоритма растривания и структуры данных объектов реализована на языке высокого уровня C++. Этот язык хорошо подходит для реализации объектной модели описываемой системы, так как является объектно-ориентированным.

Данные в памяти хранятся в виде дерева, в котором каждый объект сцены представлен в виде описания функции со своими свойствами и поддеревьями. Дерево отображается на экране (рис. 14), и мы можем скопировать, вырезать и вставить отдельные ветки или дерево целиком. Также мы можем вызывать окно свойств для каждого элемента дерева и уже в этом окне менять свойства элемента (рис. 15). Действия в основном окне программы производятся над выделенным в дереве элементом.

Для ранее созданного языка описания сцены построен синтаксический анализатор, который заносит данные из файла в дерево объектов и проверяет файл на наличие ошибок. Если существует ошибка, появляется сообщение, которое записывается в лог-файл, и построение продолжается дальше. Алгоритм сохранения сцены обходит дерево объектов и сохраняет его в файл в соответствии с языком описания сцен.

Алгоритм построения сцены из памяти делает обход дерева и построение геометрической модели, в формате построителя изображения.

4.1 Алгоритм добавления объекта в сцену, в место указанное мышкой

Находим X, Y пересечением в координаты сцены координат vrp , на который выводим изображение. Умножаем всю матрицу объекта на матрицу камеры. Запускаем алгоритм построения сцены и на последнем уровне бинарного деления проверяем, не совпали ли X, Y координаты с координатами точки, которая в данный момент просчитывается. Если совпали, берем текущую Z координату. В полученные координаты мы передвигаем объект, который хотим добавить. Таким образом, мы добавляем объект на поверхность другого объекта в место, которое указываем мышкой.

4.2 Алгоритм нахождения объектов в сцене, в месте, выделенном мышкой

Находим X, Y, Z координаты сцены, куда попали мышкой. Запускаем алгоритм построения сцены и на последнем уровне бинарного деления смотрим, не совпали ли X, Y координаты с координатами точки, которая в данный момент просчитывается. Если совпали, начинаем заносить все объекты сцены, которые поучаствовали в просчете этой точки в список. Построение картинки для скорости делаем с небольшим уровнем рекурсии и без вывода на экран.

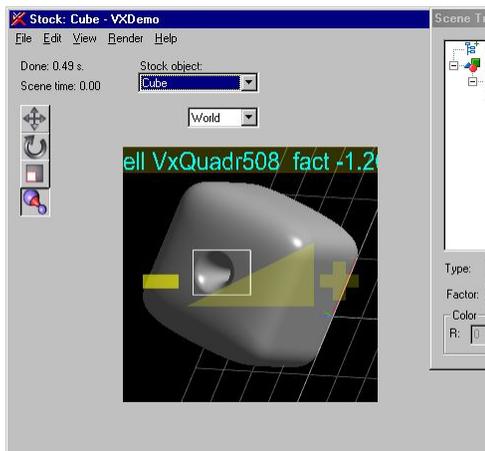


Рис. 12. Главное окно программы. Локальная деформация объекта

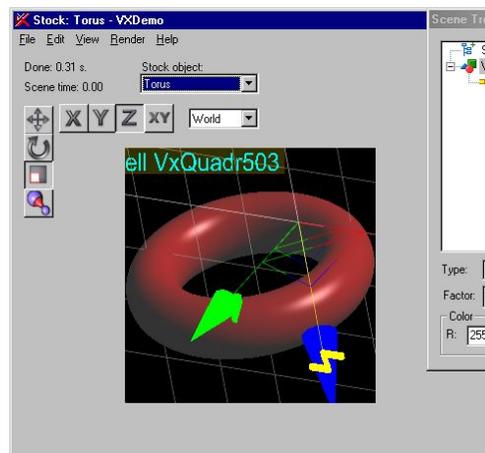


Рис. 13. Главное окно программы. Масштабирование

Таким образом, можно интерактивно добавлять, удалять, копировать, перемещать в пространстве объекты, а также добавлять или удалять какие-либо свойства. Объекты можно перемещать как проведением мышки по экрану, так и перетаскиванием слайдеров, а также вручную подставлять значения координат.

5. ЗАКЛЮЧЕНИЕ

В результате данной работы была разработана инструментальная среда, в которой интерактивно можно создавать или редактировать функционально заданные объекты с применением функций возмущения. Созданное программное обеспечение значительно упрощает создание

функционально заданных поверхностей с применением функций возмущения (рис. 16).

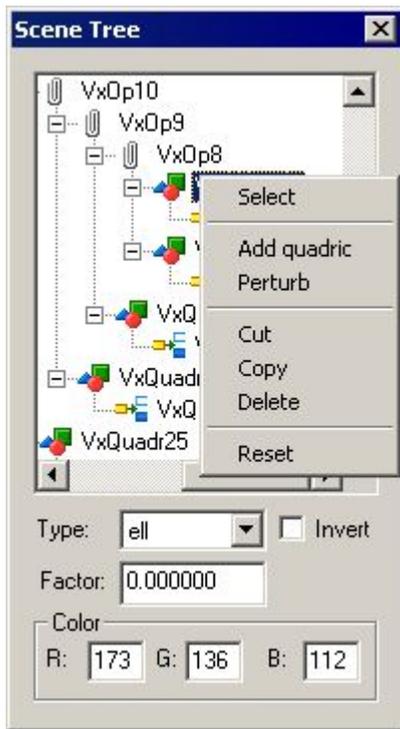


Рис. 14. Отображение дерева на экране

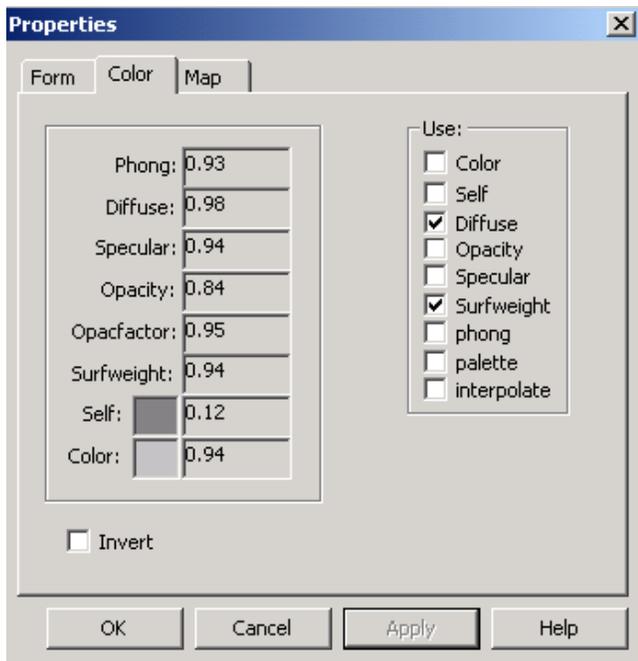


Рис. 15. Окно свойств для одного элемента дерева



Рис. 16. Пример изображения построенного с помощью данной системы (на создание объекта потребовалось примерно 15 минут времени)

6. БИБЛИОГРАФИЯ

- [1] Alias Wavefront 2001. Maya. <http://www.aliaswavefront.com>.
- [2] R.N. Perry and S.F. Frisken. Kizami: A System for Sculpting Digital Characters, in *SIGGRAPH'01*, 2001, pp. 47-56.
- [3] M. Agrawala, A.C. Beers and M. Levoy, M. 1995. 3D Painting on Scanned Surfaces, pp. 145-150. ISBN 0-89791-736-7.
- [4] Right Hemisphere 2001. DeepPaint3D. <http://www.us.deeppaint3d.com>.
- [5] B.M. OH, M. Chen, J. Dorsey and F. Durand. Image-based modeling and photoediting, in *SIGGRAPH'01*, 2001, pp. 433-442.
- [6] E. Ferley, M.-P. Cani and J.-D. Gascuel. Virtual Sculpture, *Visual Comput* 16:8, 2000, pp. 469-480.
- [7] S. Wang and A. Kaufman. Volume Sculpting, in *SIGGRAPH'95*, 1995, pp. 151-156.
- [8] J. Baerentzen. Octree-based Volume Sculpting, Proc. Late Breaking Hot Topics, *IEEE Visualization'98*, 1998, pp. 9-12.
- [9] A.Sourin, "[Functionally based virtual embossing](#)". *The Visual Computer*, 17 (2001), 4, pp.258-271.
- [10] A.Sourin, "[Functionally based virtual computer art](#)", *The 2001 ACM Symposium on Interactive Computer Graphics, I3D2001*, Research Triangle Park, NC, USA, 19-21, March, 2001, ACM ISBN: 1-58113-292-1, pp.77-84.
- [11] K.Levinski and A.Sourin, "[Interactive function-based artistic shape modeling](#)", *2002 International Symposium Cyber Worlds: Theory and Practice 2002*, Tokyo, Japan 6-8 November, 2002 pp.521-528.
- [12] K.Levinski and A.Sourin, Interactive Function-Based Shape Modeling for Cyberworlds, *2004 International Conference on Cyberworlds*, Tokyo, 18-20 November, 2004, pp.54-61.
- [13] Marc Alexa, Alexis Angelidis, Marie-Paule Cani, [Karan Singh](#), Denis Zorin. Interactive Shape Modelling. *The Eurographics Association, Volume Tutorial 5 - September 2005*.

[14] <http://viola.usc.edu/paper/mmsp97/CONTENTS/PAPERS/104/INDEX.HTM>

[15] Savchenko V., Pasko A., Kunii T., et al. Feature based sculpting of functionally defined 3D geometric objects" // Multimedia Modeling. Towards Information Superhighway, T.S. Chua, H.K. Pung and T.L. Kunii (Eds.), *World Scientific*, Singapore, 1995, P. 341.

[16] Beier T., Practical uses for implicit surfaces in animation // In Modeling and Animating with Implicit Surfaces, P 20.1-20.11, 1990. *SIGGRAPH Course Notes 23*.

[17] Blinn J., A generation of algebraic surface drawing // *ACM Transactions on Graphics*, 1(3): July 1982, P. 235-256.

[18] Bloomenthal J., Modeling the mighty maple // *Computer Graphics*, 19(3): July 1985, P. 305-311.

[19] Bloomenthal J., Shoemake K., Convolution surfaces // *SIGGRAPH'91, Computer Graphics*, vol.25, No.4, 1991, P. 251-256.

[20] Bloomenthal J. Sceletal Design of Natural Forms // *Doctoral dissertation, University of Calgary, Department of Computer Science*, 1995.

[21] McCormack J., Sherstyuk A. Creating and rendering convolution surfaces // *Computing Graphics Forum*, vol. 17, No.2, 1998, P. 113-120.

[22] Muraki S., Volumetric shape description of range data using "blobby model" // *Computer Graphics*, 25(4): July 1991, P. 227-235.

[23] Nishimura H., Hirai M., Kawai T., et al., Object modeling by distribution function and a method of image generation // *The Transactions of the Institute of Electronics and Communication Engineers of Japan*, J68-D(4): 1985, P. 718-725.

[24] Pasko A., Adzhiev V., Sourin, A., et al. Function representation in geometric modeling: concepts, implementation and applications // *The Visual Computer*, 11, 6, 1995, P. 429.

[25] Sealy G., Wyvill G., Smoothing of three dimensional models by convolution // *In Computer Graphics International'96*, June 1996, P. 184-190.

[26] Вяткин С.И., Долговесов Б.С., Есин А.В. и др. Геометрическое моделирование и визуализация функционально-заданных объектов // *Автометрия*. 1999. №6. стр. 65-73.

[27] S.I. Vyatkin, B.S. Dolgovesov, A.V. Yesin et al. "Geometric Modeling and Visualization of Functionally Defined Objects" // *Optoelectronics, Instrumentation and Data Processing*. Allerton Press, Inc. USA. - 1999.-N6.

[28] S.I.Vyatkin, B.S.Dolgovesov, A.V.Yesin, A.A.Zhigach, S.E.Chizhik, and R.A.Sherbakov. "Geometric Modeling and Visualization of Functionally Defined Objects" *Computer Graphics and Geometry Internet-Journal* Vol. 3., N. 2. <http://www.cgg.ru/>

[29] Vyatkin S., Dolgovesov B., Guimaoutdinov O., Synthesis of Virtual Environment Using Perturbation Functions, volume III (Emergent Computing and Virtual Engineering), *World Multiconference on Systemics, Cybernetics and Informatics Proceedings*, Orlando, FL, USA, July 22-25, 2001, 350-355.

[30] Вяткин С.И., Долговесов Б.С. Функции возмущения в геометрическом моделировании // *Труды 14-й Междунар.*

конф. "Графикон-2004". Москва, 2004. http://www.graphicon.ru/2004/Proceedings/technical_ru.html

Об авторах

Сергей Иванович Вяткин – к.т.н, ст.н.с. Лаборатории синтезирующих систем визуализации Института автоматизации и электрометрии СО РАН.

Адрес: Новосибирск, 630090, пр-т Коптюга, 1, ИАиЭ.

Телефон: (383) 333-36-30

E-mail: sivser@mail.ru

Борис Степанович Долговесов – к.т.н, зав. Лабораторией синтезирующих систем визуализации Института автоматизации и электрометрии СО РАН.

Адрес: Новосибирск, 630090, пр-т Коптюга, 1, ИАиЭ.

Телефон: (383) 333-36-30

E-mail: bsd@iae.nsk.su

An Interactive System for Shape Creating Based on Perturbation Functions

Abstract

In this paper we present an interactive system for function-based shape creating where relatively small perturbation functions are used rather than thousands of polygons. Interactive modification of the function model with concurrent visualization of the respective part of it lets us provide both the interactivity and any required level of detail leading to photo-realistic appearance of the resulting shapes. For shape creating we propose a set of algorithms and software based on function-based surfaces that perform an interactive rate and enable intuitive operations. We developed algorithms and software implementations for the following parts of interactive system for function-based shape creating that are most important: sculpting, filtering, painting, cutting, pasting.

Keywords: *Interactive function-based shape modeling, freeforms, perturbation functions.*

About the author(s)

Sergei I. Vyatkin (Ph.D.) is a senior scientific researcher of Synthesizing Visualization Systems Laboratory at Institute of Automation and Electrometry SB RAS.

His contact email is sivser@mail.ru.

Boris S. Dolgovesov (Ph.D.) is a head of Synthesizing Visualization Systems Laboratory at Institute of Automation and Electrometry SB RAS.

His contact email is bsd@iae.nsk.su