

УДК 681.5.15

## МЕТОД ИНТЕРАКТИВНОГО МОДЕЛИРОВАНИЯ ФУНКЦИОНАЛЬНО ЗАДАННЫХ ОБЪЕКТОВ БЕЗ ПРЕДВАРИТЕЛЬНОЙ ТРИАНГУЛЯЦИИ ПОВЕРХНОСТИ

С. И. Вяткин

*Институт автоматизации и электрометрии СО РАН,  
630090, г. Новосибирск, просп. Академика Коптюга, 1  
E-mail: sivser@mail.ru*

Проведено исследование интерактивного моделирования геометрических форм, заданных математическими функциями. Предложен метод интерактивного моделирования функционально заданных объектов без предварительной триангуляции. Это позволяет точнее создавать трёхмерные формы и упрощает систему моделирования. Для быстрой визуализации объектов используется метод, адаптированный для графических ускорителей.

*Ключевые слова:* интерактивное моделирование, геометрические объекты, функции возмущения.

**Введение.** Возможность интерактивного создания форм трёхмерных объектов рассматривается в [1–3]. До этого удовлетворительных методов моделирования функциональных объектов не было разработано. Объекты описывались вручную в текстовом виде. Данная операция чрезвычайно трудоёмка. Поэтому задача формирования инструментальной среды для редактирования трёхмерных объектов стала актуальной. Первоначально были созданы системы моделирования полигональных объектов. В системе автоматизированного проектирования AutoCAD (в версии R11) появляется возможность трёхмерного моделирования. В качестве недорогой альтернативы САПР была разработана система bCAD, в которой используется как двухмерная, так и трёхмерная технология проектирования и имеются встроенные средства для визуализации.

К более поздним разработкам относятся такие коммерческие системы, как SoftImage, Alias Wavefront (Maya), 3D Studio Max, LightWave 3D, программа трёхмерного моделирования и рендеринга Modo 3D и др. Существуют интерактивные системы моделирования на базе полиномов, неявных поверхностей [4], треугольных сеток [5], изображений [6], объёмов [7], функционально заданных моделей [8, 9].

Для полигональных моделей трудна реализация преобразований геометрических объектов, обусловленная введением операций над функциями, что необходимо при моделировании движений тел. Сложно выполнять унарные и бинарные геометрические операции, а также определять столкновения объектов.

При полигональном задании очень трудоёмка деформация поверхностей, поскольку каждая вершина треугольной сетки должна подвергаться геометрическим вычислениям.

Сегодня для реалистической графики необходимо компактное задание сцены с точным описанием геометрии трёхмерных объектов, обеспечивающее эффективную реализацию различных геометрических операций с моделями объектов при моделировании поведения взаимодействующих тел.

Таким образом, низкая реалистичность полигональных моделей и ограниченные функциональные возможности по формированию графических сцен обуславливают научно-прикладную проблему, для решения которой необходима разработка теоретических основ

моделирования и визуализации трёхмерных объектов виртуальной среды, позволяющих устранить недостатки, характерные для полигонального задания моделей.

Для моделирования форм объектов используются три основных типа функций: параметрические, явные и неявные. Параметрические функции, в частности NURBS — математическая форма (частный случай  $B$ -сплайнов), применяемая в компьютерной графике для генерации и представления кривых и поверхностей, используются во многих системах моделирования. Примерами явных и неявных функций могут служить мягкие объекты [10], капельные модели [11] и функции возмущения [12]. Однако в функциональных заданиях поверхностей вычисление определяющей функции в указанной точке часто оказывается весьма трудоёмкой задачей, поэтому для интерактивного режима [9] выполняют предварительную триангуляцию поверхности [13].

Применение известных методов задания объектов на основе функций ограничено довольно узким классом моделируемых поверхностей и медленной визуализацией. Один из главных недостатков известных методов визуализации — сложность вычисления точек поверхности. Некоторые методы не гарантируют нахождения точек поверхности. Более подробно недостатки известных методов задания функциональных объектов и визуализации описаны в [14].

В данной работе рассматривается метод интерактивного моделирования, который является средством создания трёхмерных объектов без предварительной триангуляции. Для быстрой визуализации используется метод с применением графических акселераторов [14]. Разработана инструментальная среда, в которой интерактивно можно создавать или редактировать трёхмерные объекты с помощью функций возмущения. Предлагаемое программное обеспечение значительно упрощает создание функционально заданных моделей.

**Объекты на основе функций возмущения.** Функционально заданные объекты строятся из поверхностей второго порядка с аналитическими функциями возмущения, благодаря чему достигается высокий коэффициент геометрического сжатия высокореалистичных трёхмерных объектов. Предлагается описывать геометрические объекты (свободные формы), задавая функцию отклонения (второго порядка) от базовой поверхности второго порядка (квадрики) [12].

Функция задаётся алгебраическим неравенством второй степени с тремя неизвестными  $x, y, z$  в виде  $F(X) \geq 0$ . Рассматриваются поверхности как замкнутые подмножества евклидова пространства  $E^3$ , определяемые описывающей функцией  $F(X) \geq 0$ , где  $F$  — непрерывная вещественная функция,  $X = (x_1, x_2, x_3)$  — задаваемая координатными переменными точка в пространстве  $E^3$ . Выражение  $F(X) > 0$  определяет точки внутри поверхности,  $F(X) = 0$  — точки на границе,  $F(X) < 0$  — точки, лежащие снаружи и не принадлежащие поверхности.

На базе квадрик строятся свободные формы. Свободная форма представляется композицией базовой квадрики и возмущений:

$$F'(x, y, z) = F(x, y, z) + \sum_{i=1}^N f_i R_i(x, y, z), \quad (1)$$

где  $f_i$  — формфактор,  $R(x, y, z)$  — возмущение:

$$R_i(x, y, z) = \begin{cases} Q_i^3(x, y, z), & \text{если } Q_i(x, y, z) \geq 0, \\ 0, & \text{если } Q_i(x, y, z) < 0. \end{cases} \quad (2)$$

где  $Q(x, y, z)$  — возмущающая квадрика. Поверхность является гладкой, когда степень возмущающей квадрики (2) больше двух. Это условие гарантирует непрерывность функции и её производной.

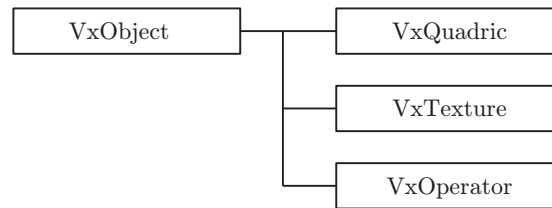


Рис. 1

Есть базовые типы квадрик и функции возмущения [12].

На рис. 1 показана иерархия объектов в системе интерактивного моделирования.

Предложенный способ описания объектов трёхмерных сцен базовыми поверхностями и функциями возмущения имеет компактное представление, что позволяет уменьшить от 10 до 1000 раз объём передаваемых данных в зависимости от конкретных трёхмерных сцен и моделей. Объекты с плоскими гранями также легко задаются, например куб может быть задан тремя квадрами. Кроме того, при решении описывающей функции в виде неравенства  $F(X) \geq 0$  можно визуализировать не только поверхность, но и внутреннюю структуру объекта.

**Метод интерактивного моделирования.** Для того чтобы создать сложную сцену, требуется в ней описать какое-то определённое количество примитивов, необходимых для конкретной задачи. Отображаемый объект, с которым алгоритм растеризации осуществляет взаимодействие посредством запросов, представляет собой всю трёхмерную сцену. Поэтому геометрическая модель должна позволять конструировать объекты и их композиции неограниченной сложности. Достигается это в первую очередь применением булевых операций объединения и пересечения. Вся сцена имеет вид дерева, каждый узел которого является объектом-конструктором, осуществляющим логические операции над своими потомками, а вершины дерева — это примитивы, используемые системой. В момент, когда растрирующий алгоритм обращается с каким-либо запросом к объекту-конструктору, этот объект обращается к своим потомкам, преобразует полученный результат и выдаёт соответствующий ответ на запрос. Потомком может быть как примитив, так и другой объект-конструктор. При применении геометрических операций, поворотов, перемещений, масштабирования к объекту-конструктору он повторяет все эти операции со своими потомками.

Таким образом, сцена описывается двоичным деревом, корнем которого является объект, представляющий всю трёхмерную сцену и обладающий свойством отвечать на запрос о пересечении с объёмом. В узлах находятся операторы объединения или пересечения. Оператор также является объектом, а листья — объектами, выраженными квадрами.

Библиотека классов функционально заданных объектов имеет статическую карту для имён зарегистрированных в библиотеке объектов и для создания функций. Имена зарегистрированных объектов автоматически определяются как лексемы при синтаксическом анализе сцены. Задачей подготовительной фазы является построение геометрической модели. На входе программы имеется несколько текстовых файлов, каждый из которых описывает логически законченный элемент сцены и все его необходимые параметры. С помощью препроцессора эти файлы транслируются в один файл, содержащий полное описание сцены. Полученный файл подаётся на вход синтаксического анализатора — парсера, который разбирает конструкции текста и осуществляет построение модели. При успешном завершении построения модель подвергается проективному преобразованию, после чего фаза предобработки заканчивается. Иерархия классов функционально заданных объектов показана на рис. 2. Структура базовых классов приведена в таблице.

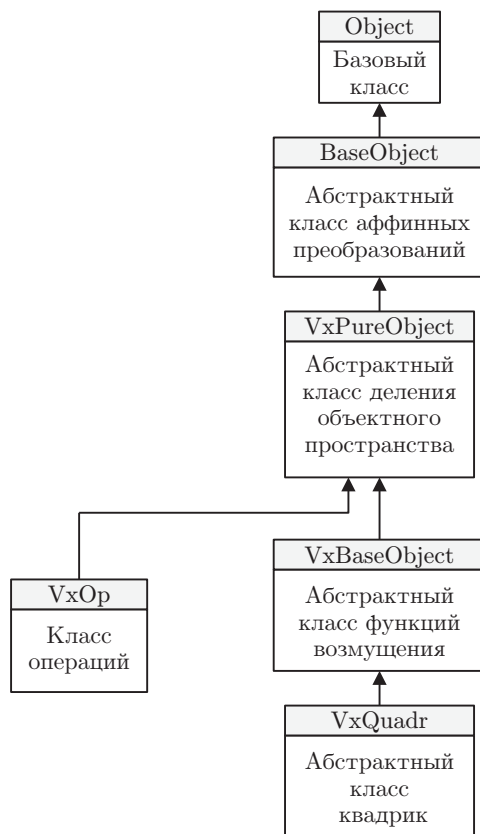


Рис. 2

Структура и имя класса	Функция
SDynCreate	Хранение имён базовых классов
string m_strClassName;	—
bool IsDerivedFrom (const SDynCreate pBaseClass) const;	Задание типа проверки
Object CreateObject();	Создание объекта m_strClassName class
static Object CreateObject (LPCSTR lpszClassName);	Поиск структуры SDynCreate struct и создание объекта, указанного в lpszClassName
mutable int m_nRefCount; virtual int Release() const;	Уменьшение указателя счётчика
virtual bool IsKindOf (const SDynCreate pClass) const;	Проверка типа
int AddRef() const;	Увеличение указателя счётчика
static const SDynCreate & GetDynCreateInfo();	Возвращение информационной структуры, ассоциированной с этим классом
int GetRefCount() const;	Возвращение текущих значений указателя счётчика

Задача конструирования объекта сводится к задаче деформации базовой поверхности нужным образом, а не к аппроксимации её примитивами. Данный процесс напоминает лепку модели из пластилина с применением геометрических операций, представленных в работе [15]. Для вычисления координат точек деформируемой поверхности используется метод определения столкновений функционально заданных объектов [16]. Данный метод обнаружения столкновений основан на отношении пересечения объектов и рекурсивном делении объектного пространства для поиска первой точки соприкосновения объектов.

Пусть объекты  $G_1$  и  $G_2$  определены как  $f_1(X) \geq 0$  и  $f_2(X) \geq 0$ . Бинарная операция пересечения объектов  $G_1$  и  $G_2$  запишется следующим образом:

$$S_c(G_1, G_2) = \begin{cases} 0, & \text{если } G_1 \cap G_2 = \emptyset, \\ 1, & \text{если } G_1 \cap G_2 \neq \emptyset. \end{cases} \quad (3)$$

Функция  $f_3(X) = f_1(X) \& f_2(X)$  используется для вычисления  $S_c$  (3). Можно утверждать, что  $S_c = 0$ , если  $f_3(X) < 0$  для любой точки пространства  $E^3$ .

Деформация состоит в возможности добавления в любую точку на поверхности возмущения с параметрами, определяемыми инструментом. Инструмент задаёт область действия, вид возмущения и выбирается пользователем (например, тонкий цилиндр). Для этого создана библиотека объектов (инструментов), т. е. можно добавлять в любую точку на поверхности возмущение с заранее заданными параметрами с помощью выбранного инструмента. Есть также функция выделения объекта в сцене, чтобы производить с ним операции: вращение, сжатие—растяжение, движение и пр. Таким образом, добавляется возмущение на поверхность объекта в место, указанное с помощью мыши. Как уже упоминалось, данные в памяти хранятся в виде дерева, в котором каждый объект сцены представлен в виде описания функции со своими свойствами. Дерево отображается на экране, можно копировать, вырезать и вставлять отдельные ветки или дерево целиком. Также имеется возможность вызывать окно свойств для каждого элемента дерева и уже в нём менять свойства элемента. Действия в основном окне программы производятся над выделенным в дереве элементом. Функция копирования геометрии сцены необходима для получения списка объектов, так как при изменении сцены надо уметь восстанавливать её в начальное состояние.

На рис. 3 приведён алгоритм нахождения наименьшего общего предка для объектов, а на рис. 4 — алгоритм добавления объекта (возмущения) в сцену. С помощью операции пересечения объектов сцены с инструментом, помещённым в координаты  $x, y, z$ , можно рассматривать объекты, на которые действуют инструментом. Для функции определения принадлежности точки объекту и для добавления уровней глубины в объект и указателей на предка применяется флаг, который будет сигнализировать об участии поддерева в рендеринге. Далее запоминается указатель на объект, если у него нет потомков. Таким образом находят корни дерева. Создаётся поле в классе `VxObject`, которое будет указывать глубину дерева (см. рис. 4). Глубина поддерева рассчитывается непосредственно перед нахождением самого младшего предка для всех объектов списка. Если глубину рассчитывать во время построения сцены, то может понадобиться пересчёт глубин всех узлов после операций объединения или пересечения, что потребует много времени, поэтому вычисляется глубина только корней. На рис. 5 показан алгоритм выбора объектов в сцене.

Важным моментом является возможность деформации исходной формы, а не только добавление новых возмущений.

Часть интерактивной системы — программа `VxDemo`, предназначенная для визуализации функционально заданных объектов. На рис. 6 показано главное окно программы. Входными данными для `VxDemo` является текстовый файл (с расширением `SCN`), созданный в каком-либо текстовом редакторе или интерактивно. Файл содержит описание сцены.

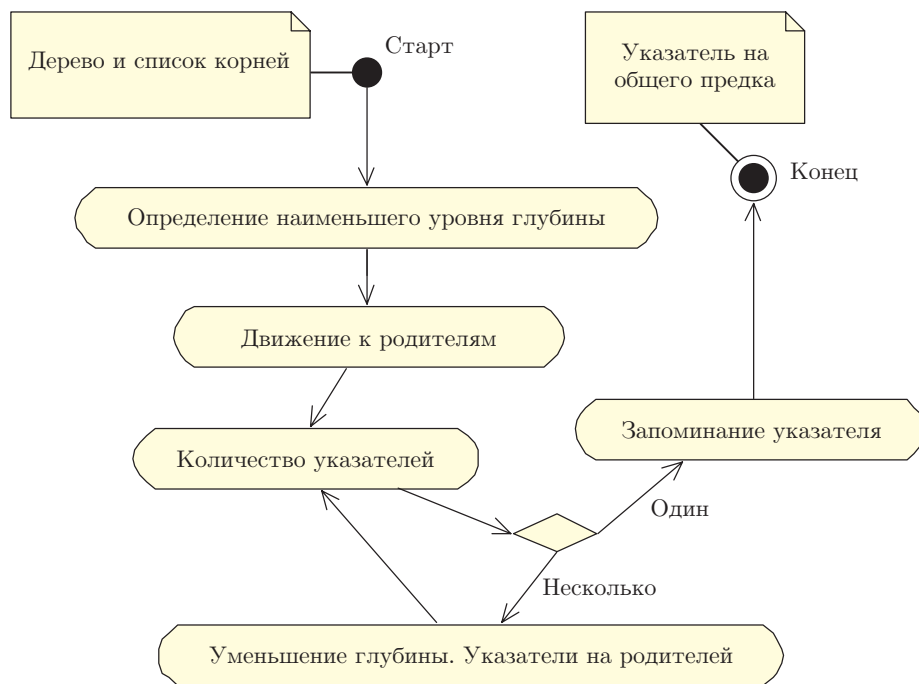


Рис. 3

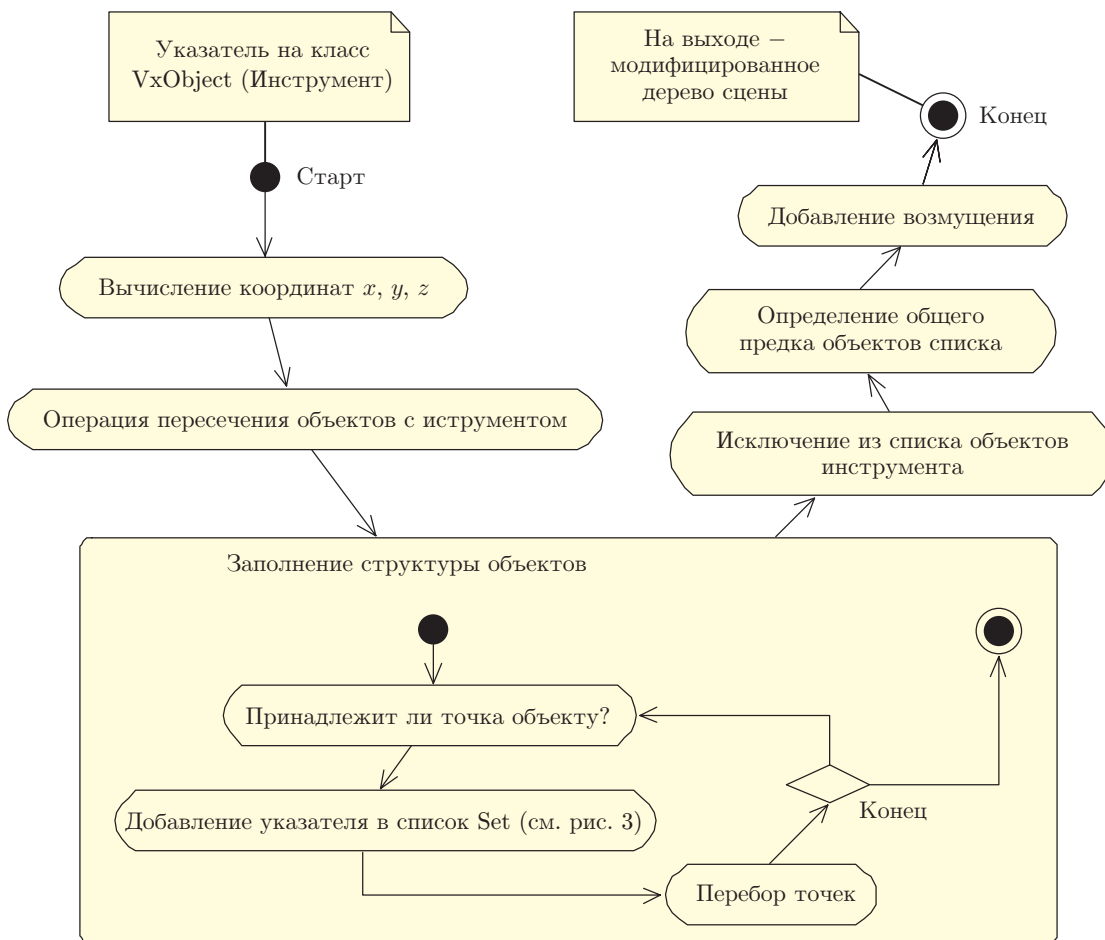


Рис. 4

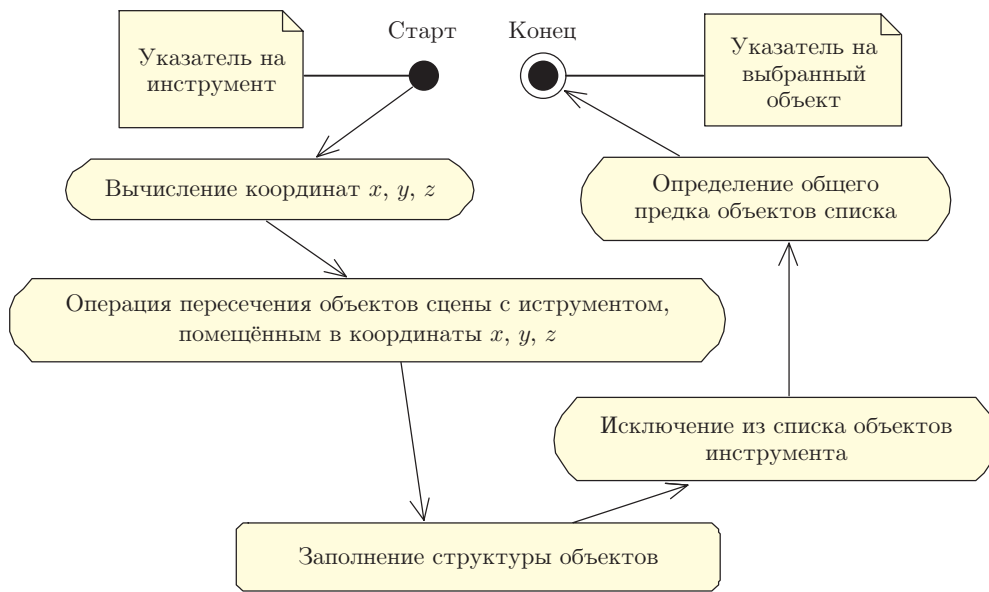


Рис. 5

Текст файла сцены представляет собой набор лексем, которые можно условно разделить на следующие категории: графические примитивы, свойства примитивов, операторы, текстура, параметры освещения.

Выходными данными являются результирующие изображения сцен в графическом формате. Для изменения параметров отображения сцены служит окно Options, которое вызывается через одноимённый пункт меню View.

Инструмент задаёт область действия и вид возмущения. Для этого пользователю графически предоставлена некоторая информация (в том числе выделение объекта, например, цветом или с помощью контейнера (bounding box), рисование осей и т. д.). Вся отрисовка «поддержки» работы с операциями производится с помощью OpenGL для скорости и функциональности.

Таким образом, к основным особенностям класса рендеринг относятся: выделение в отдельный поток, поддержка OpenGL (совместимость буферов, тест глубины и т. д.), а также формирование контейнера для объектов.

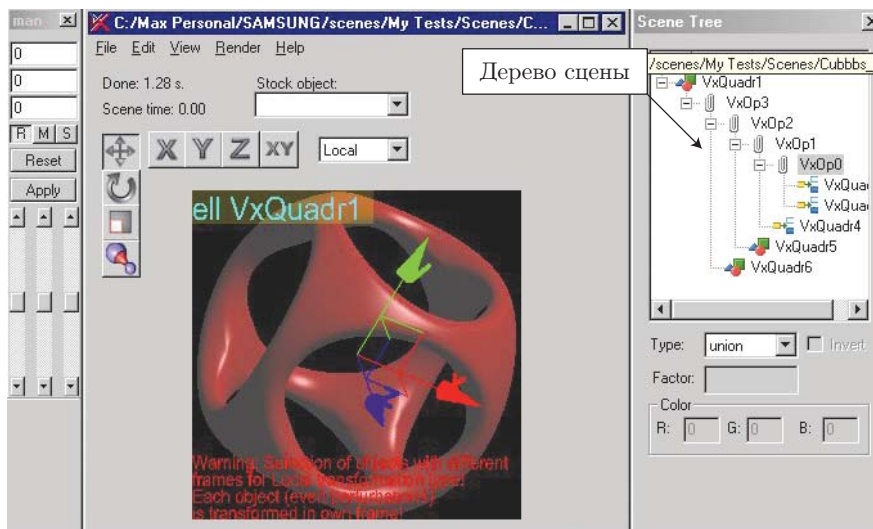


Рис. 6

Отдельный поток необходим для того, чтобы при любом изменении сцены пользователем рендеринг можно было оперативно прервать и перезапустить.

Были решены следующие задачи: реакция программы на события от мыши (выбор и модифицирование объектов); выделение объекта (или нескольких объектов) в сцене и предоставление пользователю возможности проделать с ним (с ними) некоторые операции; выбор операции — аффинные преобразования MRS (move, rotate, scale), геометрические операции и деформация; возможность работы со списком инструментов; запись в файл и загрузка дерева сцены; возможность создания новых тэгов и их распознавание.

Программная модель метода интерактивного моделирования функционально заданных объектов исполнена на языке высокого уровня C++. Этот язык хорошо подходит для объектной модели описываемой системы, так как является объектно-ориентированным.

Были разработаны и реализованы необходимые методы и классы C++: метод бинарного поиска элементов изображения; классы функционально заданных объектов; классы рендеринга функционально заданных объектов; классы интерфейса системы интерактивного объёмно-ориентированного геометрического моделирования с возможностью простого механизма расширения новых алгоритмов и характеристик.

Эти иерархические классы составляют ядро пакета, который может быть расширен для добавления функциональности или изменения характеристик.

**Заключение.** Для интерактивного моделирования и редактирования функционально заданных форм в целях уменьшения времени вычислений во многих системах применяется промежуточная триангуляция редактируемой поверхности, что приводит к усложнению как самого метода моделирования, так и всей системы в целом. Кроме этого, при таких промежуточных манипуляциях теряется точность вычислений. Следует также отметить, что не всякие геометрические формы можно удовлетворительно триангулировать, а значит, существуют ограничения на их сложность. В предлагаемой работе удалось избежать дополнительной операции триангуляции, сохранив режим интерактивности. Это связано с методами задания трёхмерных объектов, визуализации, определения столкновения объектов, геометрическими операциями и возможностями локальной и глобальной деформаций.

В результате были созданы инструментальная среда, в которой интерактивно можно формировать и/или редактировать функционально заданные объекты, и программное обеспечение, значительно упрощающее построение функционально заданных объектов с применением функций возмущения.

## СПИСОК ЛИТЕРАТУРЫ

1. **Szeliski R., Tonnesen D.** Surface modeling with oriented particle systems // Proc. of the 19th Ann. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH'92). N. Y., USA: ACM, 1992. P. 185–194.
2. **Welch W., Witkin A.** Free-form shape design using triangulated surfaces // Proc. of the 21st Ann. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH'94). N. Y., USA: ACM, 1994. P. 247–256.
3. **Curless B., Levoy M.** A volumetric method for building complex models from range images // Proc. of the 23rd Ann. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH'96). N. Y., USA: ACM, 1996. P. 303–312.
4. **Perry R. N., Frisken S. F.** Kizamu: a system for sculpting digital characters // Proc. of the 28th Ann. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH'01). N. Y., USA: ACM, 2001. P. 47–56.
5. **Agrawala M., Beers A. C., Levoy M.** 3D painting on scanned surfaces // Proc. Symp. on Interactive 3D Graphics (I3D'95). N. Y., USA: ACM, 1995. P. 145–150.



6. **Oh B. M., Chen M., Dorsey J., Durand F.** Image-based modeling and photo editing // Proc. of the 28th Ann. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH'01). N. Y., USA: ACM, 2001. P. 433–442.
7. **Ferley E., Cani M.-P., Gascuel J.-D.** Virtual sculpture // The Visual Computer. 2000. **16**, N 8. P. 469–480.
8. **Sourin A.** Functionally based virtual embossing // The Visual Computer. 2001. **17**, N 7. P. 258–271.
9. **Levinski K., Sourin A.** Interactive function-based shape modeling for cyberworlds // Proc. of the Intern. Conf. on Cyberworlds. Tokyo, Japan, 18–20 November, 2004. P. 54–61.
10. **Wyvill G., McPheeters C., Wyvill B.** Data structure for soft objects // The Visual Computer. 1988. **2**, N 4. P. 227–234.
11. **Wyvill B., Guy A., Galin E.** Extending the CSG tree. Warping, blending and Boolean operations in an implicit surface modeling system // Computer Graphics Forum. 1999. **18**, N 2. P. 8–24.
12. **Вяткин С. И.** Моделирование сложных поверхностей с применением функций возмущения // Автометрия. 2007. **43**, № 3. С. 40–47.
13. **Levinski K., Sourin A.** Interactive polygonisation for function-based shape modelling // Eurographics 2002. Saarbrücken, Germany, Sept. 2–6, 2002. P. 71–79.
14. **Вяткин С. И.** Метод бинарного поиска элементов изображения функционально заданных объектов с применением графических акселераторов // Автометрия. 2014. **50**, № 6. С. 89–96.
15. **Вяткин С. И., Долговесов Б. С., Валетов А. Т.** Геометрические операции для функционально заданных объектов с применением функций возмущения // Автометрия. 2004. **40**, № 1. С. 65–73.
16. **Вяткин С. И., Долговесов Б. С., Корсун А. С.** Определение столкновений функционально заданных объектов в задачах компьютерной графики // Автометрия. 2003. **39**, № 6. С. 119–126.

*Поступила в редакцию 13 ноября 2014 г.*

---