

УДК 004.925.84

## МЕТОДЫ ИНТЕРАКТИВНОГО МОДЕЛИРОВАНИЯ И ВИЗУАЛИЗАЦИИ ФУНКЦИОНАЛЬНО ЗАДАННЫХ ОБЪЕКТОВ ДЛЯ 3D ВЕБ-ПРИЛОЖЕНИЙ

© С. И. Вяткин, Б. С. Долговесов

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

Предложены методы интерактивного моделирования и визуализации функционально заданных объектов с помощью веб-браузера, использующего интернет-стандарт HTML5 и библиотеку программного обеспечения WebGL. Разработано приложение для моделей высокой реалистичности, которое способно работать с интерактивной частотой кадров в реальном времени. Пользователь будет иметь возможность мгновенно изменять внешний вид данных, манипулируя различными свойствами отображения, доступными через пользовательский интерфейс на экране. Предложены и реализованы функциональные описания объектов, которые позволяют определять зависящие от времени геометрические формы, их внешний вид и преобразования с помощью функций возмущения.

*Ключевые слова:* функционально заданная поверхность, функции возмущения, рендеринг, конструктивная твердотельная геометрия (Constructive Solid Geometry, CSG), 3D веб-визуализация, WebGL.

DOI: 10.15372/AUT20220112

**Введение.** Компьютерная графика, связанная с интерактивным моделированием и редактированием трёхмерных объектов, развивается быстрыми темпами. Коммерческие системы моделирования (SoftImage, Maya, 3DStudioMax и др.) были разработаны для интерактивного редактирования полигональных моделей. Известны системы моделирования, основанные на неявных функциях [1], треугольных сетках [2], изображениях [3], объёмах [4] и функциональных моделях [5].

В САПР [6] применяются методы твердотельного моделирования и объёмные теоретико-множественные (булевы) операции. Эти математические методы используются для построения трёхмерных твёрдых тел. Существует несколько способов математического описания 3D-объектов [7, 8].

В методах представления границ (B-ger) трёхмерный объект определяется с применением ограничивающих поверхностей, которые его окружают. Модели B-гер широко используются для визуализации в 3D компьютерной графике, а также в программах САПР, поскольку они могут быть быстро обработаны с помощью соответствующих алгоритмов.

В подходе конструктивной геометрии твёрдого тела (Constructive Solid Geometry, CSG) трёхмерный объект представлен комбинацией примитивных тел, таких как сфера, кубоид, цилиндр и т. д., с использованием булевой алгебры. Для создания сложных поверхностей и твёрдых тел применяются логические операции для объединения различных объектов. В евклидовом пространстве логические операции объединения ( $\cup$ ), пересечения ( $\cap$ ) и др. составляют так называемую булеву алгебру, включающую закон коммутативности.

Функциональные методы задания объектов позволяют значительно уменьшить размер базы данных по сравнению с полигональным описанием. Однако визуализировать такие объекты в реальном времени проблематично, поскольку требуются большие вычисления.

Эти методы определения объектов включают поверхности свёртки [9], капельные модели [10], мягкие объекты [11] и скелетные модели [12].

При синтезе графических сцен необходимо решать двуединую задачу — обеспечение высокой реалистичности воспроизведения графических объектов и достижение приемлемого для конкретной задачи времени формирования графических сцен.

Одним из важных приложений функциональных объектов является веб-визуализация.

Цель представленной работы — создание метода интерактивного моделирования и визуализации функционально заданных объектов с использованием WebGL для рендеринга моделей высокой реалистичности с помощью современного интернет-браузера.

**Функциональное задание объектов и геометрические операции.** Для описания поверхностей используются функции отклонения (второго порядка) от базовой квадрики [13]. Функция задаётся алгебраическим неравенством второй степени с тремя неизвестными  $x, y, z$  в виде  $F(x, y, z) \geq 0$ . Рассматриваются поверхности как замкнутые подмножества евклидова пространства ( $E^3$ ), определяемые описывающей функцией  $F(x, y, z) \geq 0$ , где  $F$  — непрерывная вещественная функция;  $x, y, z$  — координаты точки в  $E^3$ . Поверхности строятся с помощью квадрик и представляются 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;  $R_i(x, y, z)$  — возмущение:

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

Здесь  $Q_i(x, y, z)$  — возмущающая квадрака.

Используя логические операции, можно создавать объекты и композиции объектов неограниченной сложности.

Объединение:  $A \cup B$  — операция, объединяющая объекты  $A, B$  в один объект.

Пересечение:  $A \cap B$  — операция, сохраняющая только те части, которые являются общими для обоих объектов  $A, B$ .

Разница:  $A \cap (-B)$  — операция, вычитающая  $B$  из  $A$ .

Отрицательное пересечение:  $-A \cap B$ , если  $B$  пересекается с  $A$  (отрицательно), то результатом будет  $B - A \cap B$ .

Применяя булевы операторы из комбинаций нескольких твёрдых тел, можно создать более сложные 3D-объекты.

**Интерактивное моделирование функционально заданных объектов.** Сцена имеет вид дерева, каждый узел которого является объектом-конструктором, осуществляющим логические операции над своими потомками, а вершинами дерева — 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. При этом потомком может являться как примитив, так и другой объект-конструктор. При применении геометрических операций, поворотов, перемещений, масштабирования к объекту-конструктору он производит все эти операции со своими потомками.

Таким образом, сцена включает в себя двоичное дерево, 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 считается объектом, а объекты в виде квадрик — листьями.



Рис. 1. Алгоритм нахождения наименьшего общего предка для объектов

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 в течение синтаксического анализа 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 закончена.

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.

Деформация состоит в возможности добавления в любую точку на поверхности возмущения с параметрами, задаваемыми инструментом, который также определяет область действия и вид возмущения. 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 вращения, сжатия-растяжения, движения и др. На рис. 1 приведён алгоритм нахождения наименьшего общего предка для объектов, а на рис. 2 — алгоритм добавления объекта (возмущения) в сцену. С помощью операции пересечения объектов сцене с инструментом, помещённым в координаты  $x, y, z$ , можно рассматривать объекты для воздействия инструментом (см. рис. 2). Для функции определения принадлежности точки к объекту, добавления уровней глубины в объект и указателей на предка применяется флаг, который будет сигнализировать об участии поддерева в рендеринге. Перед возвратом из функции запоминается указатель на



Рис. 2. Алгоритм добавления возмущения в сцену

объект, если у него нет потомков. Таким образом, находятся корни дерева. Создаётся поле в `VxObject`, которое будет показывать на глубину дерева (см. рис. 2). 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, для чего потребуется много времени. Вычисляется глубина только корней, а затем — нужная глубина узла. На рис. 3 показан алгоритм выбора объектов в сцене.

Важным моментом является возможность работать только с возмущениями, чтобы при деформации менялись существующие возмущения, а не только добавлялись новые.

Частью интерактивной системы является программа Демо, предназначенная для визуализации функционально заданных объектов. Входными данными для Демо является текстовый файл (с расширением `SCN`), созданный в каком-либо текстовом редакторе или интерактивно. Файл содержит описание сцены. 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 Опции, которое вызывается через одноимённый пункт меню Вид.

К основным особенностям класса рендеринга относятся: выделение рендеринга в отдельный поток, поддержка рендеринга OpenGL (совместимость буферов, теста глубины и т. д.). Отдельный поток также необходим для того, чтобы при любом изменении сцены пользователем рендеринг можно было оперативно прервать и перезапустить.

Разработанный инструмент интерактивного моделирования позволяет создавать формы путём таких постепенных интерактивных модификаций, как скульптура, резьба и жи-

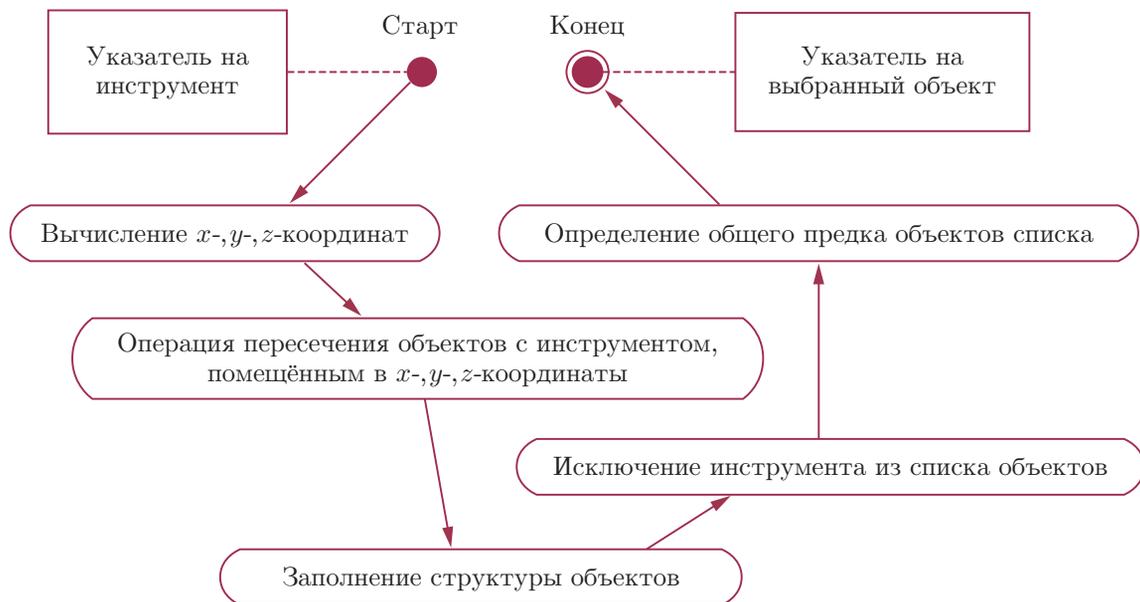


Рис. 3. Алгоритм выбора объектов в сцене



Рис. 4. Высокоуровневые модули системы интерактивного моделирования

вопись. Начальная форма может быть либо выбрана в одной из базовых форм, либо определена аналитической неявной функцией. Полученная модель содержит информацию обо всех изменениях, внесённых в геометрию и внешний вид формы, и перемещается в трёх измерениях. Можно определить любой уровень детализации, который изменяет размер треугольников, используемых для полигонизации, и соответствующим образом перерисовывает форму [14]. В альтернативном методе [15] преобразование полигональных моделей в функциональные (модуль преобразования полигональных моделей в функциональные (рис. 4)) происходит путём строгого математического вычисления, а не итерационно и не приближениями. На рис. 4 показана общая схема системы. Чтобы визуализировать объекты, определяемые функциями, требуется преобразовать их в стандартные примитивные объекты (точки, линии и полигоны), которые затем будут отображаться браузером (модуль полигонизации функциональных моделей).

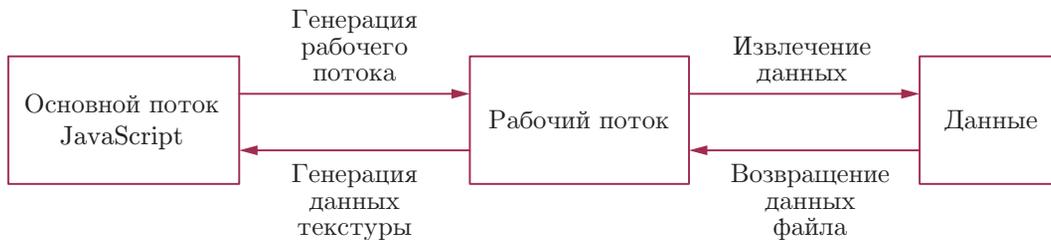


Рис. 5. Загрузчик данных Web Worker

**3D веб-визуализация.** Методы веб-визуализации можно разделить на три класса: передача изображений, передача необработанных данных и передача моделей. Передача изображений предполагает, что 3D веб-визуализация может быть реализована путём передачи клиентам 2D-изображений, отображаемых на серверах. Передача необработанных данных предполагает, что геометрические модели находятся и отображаются на клиентском компьютере, в то время как только необходимые данные (например, координаты объекта, состояние объекта, события и т. д.) передаются через Интернет для управления визуализацией и взаимодействием. При использовании передачи моделей 3D-объекты отображаются на клиентских компьютерах, но модели передаются через Интернет вместе с другими данными связи. Каждый раз, когда требуется обновление или создание новых моделей, соответствующие данные могут быть легко доставлены на клиентские компьютеры и кэшированы там. С помощью современных компьютеров и программного обеспечения для веб-визуализации при таком подходе можно достичь довольно высокого визуального реализма. Функцией, которая используется в данной работе, является элемент HTML5 canvas, позволяющий использовать библиотеку WebGL внутри веб-страниц. Элемент HTML5 — это интернет-стандарт, который реализуется в современных браузерах, таких как Google Chrome, Mozilla Firefox, Apple, Safari и др. Одна из основных целей HTML5 — свести к минимуму необходимость в плагинах для отображения мультимедийного контента или полностью устранить необходимость в них.

WebGL — это библиотека программного обеспечения, которая использует JavaScript для реализации OpenGL. Разработано приложение для визуализации реалистичных функционально заданных объектов, которое можно использовать через браузер с поддержкой WebGL в режиме реального времени. Приложение требуется для отображения изображения и графического пользовательского интерфейса (GUI) для взаимодействия. Поскольку это программное обеспечение запускается через браузер, оно реализовано таким образом, чтобы избежать замораживания любых страниц, что, в свою очередь, может привести к зависанию браузера и всех других открытых вкладок. Приложение разбито на несколько элементов. Основными частями приложения являются подсистемы загрузки данных и визуализации. Раздел подсистемы загрузки отвечает за последовательную передачу данных с сервера в приложение на локальном клиентском компьютере. Система визуализации оптимизирована для работы с WebGL, а также с использованием пиксельных шейдеров GLSL, доступных через WebGL, для улучшения визуальной иллюстрации данных.

Чтобы решить проблему зависания приложения во время загрузки, наш подход использует элемент браузера, называемый Web Worker, который позволяет приложению запускать сценарии JavaScript в фоновом режиме системы. Эти потоки выполняются независимо от основного потока и, следовательно, не влияют на пользовательский интерфейс приложения. В нашем приложении Web Worker создан для загрузки данных с сервера, в то время как основной поток будет взаимодействовать с пользователем (рис. 5). Это означает, что пользователь сможет изменять данные в режиме реального времени по мере загрузки. Браузер не сообщит пользователю, что страница перестала отвечать на запро-

Таблица 1

## Производительность Google Chrome

Число итераций	Время загрузки	Частота кадров, кадр/с	Генерация текстур
1	12,7	39	22,1
2	8,5	40	18,2
3	8,4	38	18,1
4	8,2	35	17,9

сы. Приложение использует комбинацию HTML, CSS, JavaScript и WebGL. Два помощника для WebGL были использованы в системе. Это утилита WebGL JavaScript, которая включает функции для инициализации WebGL и проверку совместимости браузера с WebGL. Второй также является JavaScript, которая эффективно обрабатывает операции с матрицей и вектором. В Web Worker есть функция включения сообщения, когда основной поток отправляет рабочему сообщению. Когда Web Worker получает сообщение, оно запускает загрузку системы. Web Worker использует информацию, переданную в сообщении, для загрузки правильных данных. Чтобы предоставить пользователю визуально полный набор данных во время загрузки, необходимо применять подход «разделяй и властвуй». Первые загруженные данные — это те, индекс которых делится на половину количества, потом на четверть, потом на восемь и т. д. Когда данные, содержащиеся в файле, возвращаются, они отправляются в функцию для обработки и анализа. Если сообщений об ошибке не было, то данные передаются в другую функцию. Одной из основных задач функций является анализ данных и включение их в 3D-описание объектов. Другая важная обязанность этой задачи — передача массивов, содержащих информацию о текстурах.

**Результаты работы.** Представлены результаты тестирования при запуске приложения, которое было сделано с использованием ПК ASUS D500SA-3101000550 [90PF0231-M16330], Intel Core i3 10100, 4x3600 МГц, 8 ГБ DDR4, SSD 256 ГБ и NVS 5100M. Видеокарта NVS 5100M имеет архитектуру Tesla 2,0 с памятью 1 Гб (GDDR3) на частоте 1,6 ГГц со 128-битным интерфейсом и пропускной способностью 25,60 Гб/с.

Браузер с поддержкой WebGL, установленный на компьютере для тестирования, принадлежал Google Chrome версии 13.x. Результаты взяты из тестовых запусков четырёх итераций из Chrome. Используемое интернет-соединение имело скорость загрузки 95 Мбит/с и скорость загрузки 25,9. В табл. 1 описаны результаты теста для браузера Chrome.

**Заключение.** Предложены методы интерактивного моделирования и визуализации функционально заданных объектов для 3D веб-приложений. Методы позволяют определять формы с помощью функций возмущения. Эти функции задают геометрию объекта, трёхмерную геометрическую текстуру, цвет и их преобразование во времени. Функции могут использоваться одновременно и группироваться в сценарии. Пользовательские операции, зависящие от времени, также могут быть определены аналитическими функциями. Сложные объекты могут быть установлены путём ввода кода, конвертирования из полигональных моделей или в интерактивном режиме без увеличения размера файла, как это обычно происходит при использовании моделей на основе полигонов. Основная цель состояла в том, чтобы представить систему, которая могла бы работать в современных веб-браузерах. Это достигается за счёт использования HTML5 и WebGL в браузерах с поддержкой WebGL.

## СПИСОК ЛИТЕРАТУРЫ

1. **Perry R. N., Frisken S. F.** Kizamu: A system for sculpting digital characters // Proc. of the Intern. Conference on Computer Graphics and Interactive Techniques (Siggraph 2001). Los Angeles, USA, Aug. 12-17, 2001. P. 47–56. DOI: 10.1145/383259.383264.
2. **Agrawala M., Beers A. C., Levoy M.** 3D painting on scanned surfaces // Proc. of the Symposium on Interactive 3D Graphics. Monterey, USA, April 9-12, 1995. P. 145–150. DOI: 10.1145/199404.199429.
3. **Oh B. M., Chen M., Dorsey J., Durand F.** Image-based modeling and photoediting // Proc. of the Intern. Conference on Computer Graphics and Interactive Techniques (Siggraph 2001). Los Angeles, USA, Aug. 12-17, 2001. P. 433–442. DOI: 10.1145/383259.383310.
4. **Ferley E., Cani M. P., Gascuel J. D.** Practical volumetric sculpting // Vis. Comput. 1999. **16**, N 8. P. 469–480. DOI: 10.1007/PL00007216.
5. **Sourin A.** Functionally based virtual embossing // Vis. Comput. 2001. **17**, N 4. P. 258–271. DOI: 10.1007/s003710100109.
6. **Farin G., Hoschek J., Kim M.-S.** The Handbook of Computer Aided Geometric Design. eBook. Elsevier, 2002.
7. **Pottmann H., Brell-Cokcan S., Wallner J.** Discrete Surfaces for Architectural Design. Archived 2009-08-12 at the Wayback Machine. Avignon: Nashboro Press, 2006. P. 213–234.
8. **Farin G.** Curves and Surfaces for CAGD: A practical Guide. San Francisco — San Diego — London: Morgan-Kaufmann publishers, 2002. 502 p.
9. **Vyatkin S., Romanyuk A., Troianovska T. et al.** Convolution surfaces using volume bounding // Proc. of the 9th Intern. Conference on Advanced Computer Information Technologies. Ceske Budejovice, Czechia, June 5–7, 2019. P. 461–465. DOI: 10.1109/ACITT.2019.8779894.
10. **Muraki S.** Volumetric shape description of range data using «Blobby model» // Comput. Graph. 1991. **25**, N 4. P. 227–235. DOI: 10.1145/122718.122743.
11. **Wyvill G., McPheeters C., Wyvill B.** Data structure for soft objects // Vis. Comput. 1986. **2**, N 4. P. 227–234. DOI: 10.1007/BF01900346.
12. **Tagliasacchi A., Delame T., Spagnuolo M. et al.** 3D skeletons: A state-of-the-art report // Comput. Graph. Forum. 2016. **35**, N 2. P. 573–597. DOI: 10.1111/cgf.12865.
13. **Вяткин С. И.** Моделирование сложных поверхностей с применением функций возмущения // Автометрия. 2007. **43**, № 3. С. 40–47.
14. **Vyatkin S. I.** Polygonization method for functionally defined objects // Intern. Journ. Automation, Control and Intell. Syst. 2015. **1**, N 1. P. 1–8.
15. **Вяткин С. И., Долговесов Б. С.** Метод сжатия геометрических данных с применением функций возмущения // Автометрия. 2018. **54**, № 4. С. 18–25. DOI: 10.15372/AUT20180403.

*Поступила в редакцию 29.11.2021*

*После доработки 29.11.2021*

*Принята к публикации 28.12.2021*