

# Создание баз данных на основе скалярных функций возмущения

С.И. Вяткин, Б.С. Долговесов  
Институт автоматизации и электрометрии СО РАН,  
Новосибирск, Россия  
[sivser@mail.ru](mailto:sivser@mail.ru), [bsd@iae.nsk.su](mailto:bsd@iae.nsk.su)

## Аннотация

Рассматриваются проблемы подготовки баз данных на основе скалярных функций возмущения. Предлагается эффективный метод преобразования данных в новый формат и программный комплекс для подготовки баз данных на основе моделей CAD/CAE/CAM систем, обеспечивающих высокий уровень интерактивности.

**Ключевые слова:** Создание баз данных, скалярные функции возмущения.

## 1. ВВЕДЕНИЕ

В Лаборатории синтезирующих систем визуализации был разработан свой метод задания объектов и визуализации - "Voxel Volumes" [1]. Принципиальное отличие предлагаемого метода от остальных заключается в неполигональном представлении поверхностей свободных форм [2] и использовании алгоритма многоуровневого отслеживания лучей для растеризации сцены. Объекты представляются либо функционально [3], что обеспечивает компактность задания, либо на основе скалярных функций возмущения [4, 5]. Использование алгоритма многоуровневого отслеживания лучей решает проблемы характерные для систем с растриванием на плоскости и проблемы больших объемов данных в воксельно - базируемых системах.

Однако задание поверхностей свободных форм на основе скалярных функций возмущения, наряду с достоинствами, имеет недостаток, связанный со спецификой данных, являющихся двумерной таблицей чисел, модификация которой человеком при больших размерах очень сложна.

Исходя из вышеописанных требований к системе визуализации "Voxel-Volumes" и специфики представления геометрических объектов, был разработан программный комплекс для подготовки баз данных на основе моделей CAD/CAE/CAM систем, обеспечивающих высокий уровень интерактивности с целью использования уже готовых моделей и облегчения процесса построения новых моделей для системы Voxel-Volumes с использованием широкого спектра инструментов современных моделирующих программных комплексов класса 3D-Studio, bCAD, обеспечивающих пользователю высокий уровень интерактивности при построении модели.

## 2. ПРОБЛЕМЫ ПОДГОТОВКИ БАЗ ДАННЫХ В СИСТЕМЕ VOXEL- VOLUMES

Из геометрической модели объектов системы Voxel-Volumes [1], в которой используется задание объектов в виде неявных функций и на основе скалярных функций возмущения, очевидно, что процесс подготовки баз данных для создания моделей реальных объектов, без средств автоматизации

моделирования будет очень трудоемким. Он осложнен тем, что в процессе моделирования человек имеет дело не с объектом, который он может видеть, а с его функциональным представлением. Как правило, для простых поверхностей, представляемых квадратами и аналитическими функциями возмущения процесс моделирования идет по схеме

1. изменение функции
2. генерация изображения
3. результат удовлетворителен ?
  - если да, то моделирование прекращается.
  - если нет, то возвращаемся в 1.

Данный процесс осложнен тем, что на обычном персональном компьютере процесс генерации больших моделей занимает время, недостаточное для обеспечения интерактивности процесса моделирования. Подготовка баз данных для поверхностей, использующих задание объектов на основе скалярных функций возмущения, вдобавок, осложнено тем, что скалярные функции задаются двумерной таблицей чисел, например, в текстовом файле. Соответственно, редактировать файл, состоящий даже из нескольких десятков строк очень трудно, при количестве их в несколько сотен, это становится практически невозможным, а ведь такие данные представляют очень небольшую таблицу. Проблема частично снимается тем, что в качестве баз данных для земного ландшафта можно использовать DEM (digital elevation model) модели. Этот стандарт разработан U.S. Geological Survey и, по сути, является таблицей высот земной поверхности с отсчетами через 7.5 или 15 минут. DEM модель состоит из двух файлов, бинарного файла данных в котором записаны высоты в виде 16- разрядных целых чисел и файла заголовка который описывает формат записи чисел используемый в файле данных (BigEndian или SmallEndian ), а так же участок на земной поверхности, который описывают высоты в файле данных.

Но для создания прочих поверхностей, отличных от земного ландшафта, очень трудно найти способ приемлемой автоматизации процесса подготовки баз данных. Для решения этой проблемы в работе [4] был предложен способ задания поверхностей на основе скалярных функций возмущения.

В данной работе предлагается метод создания карт высот из баз данных программ 3D-моделирования и 3D-проектирования (CAM/CAD), таких как 3d-studio и bCAD. Эти программы обладают достаточной степенью интерактивности благодаря тому, что используют проволочные модели для визуализации объектов на этапе проектирования, это и обеспечивает высокую скорость отрисовки изображения модели. Так же, такие программы предоставляют широчайший спектр инструментов для

создания и редактирования сложных поверхностей. Кроме того, эти программы предоставляют программный интерфейс (API) для написания подключаемых модулей, с помощью которых можно автоматизировать процесс моделирования. Так же, с помощью этого интерфейса можно получить описание геометрических свойств объектов, например, таких как координаты вершин треугольников триангулированной поверхности. Как правило, для написания этих модулей используются языки высокого уровня. В системе AutoCAD для написания модулей используется язык AutoLisp, в bCAD для этого используется язык Java.

### 3. МЕТОД ПОДГОТОВКИ БАЗ ДАННЫХ

Как известно, объекты в системах проектирования представляются триангулированными поверхностями, либо позволяют получить такое представление. Как правило, для сложных поверхностей используется нерегулярно-триангулированное представление. В этом случае поверхность представляется в виде списка вершин и списка ссылок на вершины. Список ссылок на вершины используется для определения элементов триангуляции через список вершин. Запись поверхности в таком случае имеет следующий вид

$$V=\{x_1, y_1, z_1, \dots, x_i, y_i, z_i, \dots, x_n, y_n, z_n\}, \quad F=\{0, 1, 2, \dots, k, m, l\}, \quad (1)$$

где  $V$  представляет собой массив координат вершин элементов триангуляции, а  $F$  - список граней; соответственно, количество вершин этой поверхности равно  $n$ , количество элементов триангуляции равно  $L_F/3$ , где  $L_F$  - количество элементов в  $F$ . Каждый элемент триангуляции определяется как тройка вершин  $T_{v_1}, T_{v_2}, T_{v_3}$ , а вершина определяется тройкой координат  $x, y, z$ . Соответственно, первый элемент триангуляции определяется через массивы  $V$  и  $F$  следующим образом.

$$T_{v_{0x}}=V[F[0]*3], \quad T_{v_{0y}}=V[F[0]*3+1], \quad T_{v_{0z}}=V[F[0]*3+2],$$

$$T_{v_{1x}}=V[F[1]*3], \quad T_{v_{1y}}=V[F[1]*3+1], \quad T_{v_{1z}}=V[F[1]*3+2],$$

$$T_{v_{2x}}=V[F[2]*3], \quad T_{v_{2y}}=V[F[2]*3+1], \quad T_{v_{2z}}=V[F[2]*3+2],$$

где  $i$  - элемент триангуляции ( $j=i*3$ ) как

$$T_{v_{ix}}=V[F[j]*3], \quad T_{v_{iy}}=V[F[j]*3+1], \quad T_{v_{iz}}=V[F[j]*3+2]$$

$$T_{v_{y_{i+1}}}=V[F[j+1]*3], \quad T_{v_{y_{i+1}}}=V[F[j+1]*3+1], \quad T_{v_{z_{i+1}}}=V[F[j+1]*3+2]$$

$$T_{v_{x_{i+2}}}=V[F[j+2]*3], \quad T_{v_{y_{i+2}}}=V[F[j+2]*3+1], \quad T_{v_{z_{i+2}}}=V[F[j+2]*3+2],$$

где  $F[j]$  -  $j$ -й элемент в массиве граней  $F$ , а  $V[F[j]*3]$  - тройка в массиве координат  $V$ , соответствующая этому элементу.

Представление поверхностей в таком виде помогает уменьшить количество дублирующейся информации за счет того, что элемент триангуляции задается через ссылки на тройки координат вершин, которые являются целыми числами, в то время, как непосредственное задание одной вершины (задание ее координат) требует тройки вещественных чисел.

Пусть мы имеем объект  $O$ , и триангуляцию  $T$  представляющую поверхность объекта  $O$  в виде (1). Исходя из этого, нам нужно построить представление объекта в виде

двух карт высот, т.е. в виде двух *ориентированных* плоскостей  $P_U$  и  $P_D$ , и соответствующих им двух двумерных таблиц чисел  $t_u$  и  $t_d$ .  $t_u$  характеризует отклонение исходной поверхности, от плоскости  $P_U$  в направлении нормали  $P_U$ .  $t_d$  характеризует отклонение исходной поверхности от плоскости  $P_D$  в направлении нормали  $P_D$ . Другими словами, нам нужно построить модель объекта  $O$ , используя метод раздельного моделирования, описанный в [4]. Соответственно, поверхность объекта  $O$  должна удовлетворять требованиям, которые предъявляются к поверхностям моделируемым с его помощью. Далее будет описан метод построения модели методом раздельного моделирования использующий разбиение объекта *одной* плоскостью. Использование разбиения только одной плоскостью было выбрано потому, что реализация нижеописанного метода для большого количества разбивающих плоскостей не имеет особого смысла. Так как в случае не удовлетворения моделью условиям используемого метода, интерфейс САМ систем позволяет разбить модель на составляющие ее группы поверхностей, каждая из которых уже будет удовлетворять этим условиям, и построить по отдельности модели этих групп предлагаемым методом. В то время, как реализация удобного интерфейса для обеспечения такого построения в случае использования разбиения более чем одной плоскостью кажется бессмысленной.

Плоскость  $B_p$  разбивающую объект на две части далее будем называть *плоскостью разбиения* триангуляции  $T$ . Два треугольника триангуляции  $T$  назовем *связанными по ребру* или *по вершине*, если они имеют, по крайней мере, одно общее ребро, либо одну общую вершину, соответственно. Треугольники, которые связаны по двум вершинам, являются *связанными по ребру*. Треугольник назовем *ориентированным* в направлении вектора  $v$ , или просто *ориентированным*, если угол между нормалью треугольника  $n$  и вектором  $v$  меньше либо равен  $\pi/2$ . Другими словами, если скалярное произведение:

$$(\vec{n} \cdot \vec{v}) \geq 0.$$

По аналогии, *ориентированной* назовем плоскость со строго определенным вектором нормали. Соответственно, плоскость  $P_U$  ориентирована в направлении вектора нормали  $B_p$ ,  $P_D$  - ориентирована в противоположном направлении. Далее, карту высот для плоскости  $P_U$  будем называть *верхней* картой высот, карту высот соответствующую плоскости  $P_D$  будем называть *нижней* картой высот. Не теряя общности, можно считать, что плоскостью разбиения  $B_p$ , триангуляции  $T$ , будет плоскость  $XOY$ , ориентированная в направлении оси  $Z$ .

Процесс построения состоит из четырех этапов.

1. Нахождение прямоугольного параллелепипеда, содержащего триангуляцию.
2. Внешняя ориентация триангуляции.
3. Разбиение триангуляции.
4. Построение карты высот.

Нахождение прямоугольного параллелепипеда, который содержит исходный объект, происходит путем выборки максимальных и минимальных значений координат из массива вершин триангуляции  $T$ . Координатами вершин параллелепипеда определяются как

$P_{\text{модели}} = \{x_{\min} y_{\min} z_{\min}, x_{\min} y_{\max} z_{\min}, x_{\max} y_{\min} z_{\min}, x_{\max} y_{\max} z_{\min}, x_{\min} y_{\min} z_{\max}, x_{\min} y_{\max} z_{\max}, x_{\max} y_{\min} z_{\max}, x_{\max} y_{\max} z_{\max}\}$ , где  $x_{\min}, y_{\min}, z_{\min}$  – минимальные координаты вершин,  $x_{\max}, y_{\max}, z_{\max}$  – максимальные.

Далее производится внешняя ориентация треугольников исходной поверхности. Она заключается в вычислении нормали к треугольнику так, чтобы эта нормаль была внешней к поверхности. Для этого происходит выборка наиболее высокого треугольника, то есть треугольника, одна из вершин которого имеет самую большую  $z$  – координату в массиве вершин триангуляции  $T$ . Если  $z$  – координата вершины этого треугольника находится выше плоскости разбиения  $B_p$ , то он ориентируется в направлении  $z$ , иначе – в направлении  $-z$ . После ориентирования этого треугольника, связанные с ним треугольники ориентируются в направлении его нормали, и помечаются как обработанные. После чего, ориентируются все смежные треугольники этих треугольников и т. д. Критерий связности (по ребру или по вершине) устанавливается пользователем. Если после ориентирования в  $T$  остались необработанные треугольники, из них выбирается самый верхний и процесс повторяется до тех пор, пока не останется ни одного неориентированного треугольника.

Процесс ориентирования треугольника  $t$ , заданного 3-мя вершинами  $v_1, v_2, v_3$ , в направлении вектора  $n$ , заключается в вычислении нормали треугольника  $n_{tr}$  по следующей формуле

$$n_{tr} = [(\vec{v}_1 - \vec{v}_2) \otimes (\vec{v}_2 - \vec{v}_3)] (\text{sign}((\vec{v}_1 - \vec{v}_2) \otimes (\vec{v}_2 - \vec{v}_3)) \bullet n), \quad (2)$$

где  $\otimes$  обозначает векторное произведение,  $\bullet$  – скалярное,  $\text{sign}$  – функцию знака, а  $v_1, v_2, v_3$  – вершины треугольника.

После ориентирования триангуляции по критерию соподнаправленности нормалью треугольника и разбивающей плоскости  $B_p$ , производится разделение исходного массива треугольников триангуляции  $T$  на два массива, “верхний”  $T_U$  и “нижний”  $T_D$ .

То есть  $T_i \subset T_u$  если  $(N_{T_i} \bullet N_{B_p}) \geq 0$ ,  
 $t_i \subset D$  если  $(N_{T_i} \bullet N_{B_p}) < 0$ ,

где  $N_{B_p}$  – нормаль разбивающей плоскости,  $N_{T_i}$  нормаль  $i$ -го элемента триангуляции  $T$ .

После этого строятся верхняя и нижняя карты высот из групп треугольников  $T_U$  и  $T_D$  ориентированных в направлении вектора  $N_{B_p}$  и  $-N_{B_p}$ , соответственно.

Для этого создается карта высот заданного разрешения в предположении, что плоскостью разбиения является плоскость  $ХОУ$ . Она представляет собой прямоугольник, являющийся проекцией объемлющего параллелепипеда на плоскость  $ХОУ$ . Узлы карты высот расположены на расстоянии  $dl = \max(x_{\max} - x_{\min}, y_{\max} - y_{\min}) / \text{map\_dim}$ , где  $x_{\max}, x_{\min}, y_{\max}, y_{\min}$ , соответственно, максимальные и минимальные  $x$  и  $y$  координаты проекции объемлющего параллелепипеда, а  $\text{map\_dim}$  – размерность карты высот, указанная пользователем.

Из каждого узла сетки выпускается луч в направлении вектора  $N_p$  для верхней группы и в направлении  $-N_p$  для нижней. Луч представляет собой два вектора – вектор начала и единичный нормализованный вектор направления, то есть

$ray = \{pos, dir\}$ . Таким образом, луч это множество точек в  $\mathbb{R}^3$  определяемое следующим векторным уравнением

$$\vec{v} = pos + dir * t; t \in \mathbb{R}, t \geq 0 \quad (3)$$

Луч выходящий из  $[i, j]$ -го узла верхней карты высот

$$ray_{ijU} = \{pos_{ij}, n\}, pos_{ij} = (x_{\min} + dl * i, y_{\min} + dl * j, z_{\min}), n = N_{B_p},$$

луч выходящий из  $[i, j]$ -го узла нижней карты высот

$$ray_{ijD} = \{pos_{ij}, n\}, pos_{ij} = (x_{\min} + dl * i, y_{\min} + dl * j, z_{\min}), n = -N_{B_p}.$$

Для каждого луча выпущенного верхней карты высот производится тест на пересечение с каждым треугольником из верхней группы. Если луч пересекает треугольник, то вычисляется расстояние от узла карты высот, из которого он был выпущен, до точки его пересечения с треугольником. Далее, модуль этого расстояния сравнивается с числом находящегося в данном узле карты высот, и в узел записывается максимальное из этих двух чисел.

Так как луч перпендикулярен плоскости карты высот, то  $x$  и  $y$  координаты точки пересечения равны  $x$  и  $y$  координатам вектора  $pos$  данного луча, следовательно, расстояние от точки пересечения луча с треугольником до узла карты высот равно модулю разницы  $z$  координат точки пересечения и узла карты высот

$t_u[i][j] = \max(|z_{inter} - z_{pos}|)$ , где  $z_{inter}$  –  $z$ -координата точки пересечения луча с треугольником. Таким же образом производится построение нижней карты высот.

Точка пересечения луча с треугольником определяется по следующему алгоритму. Сначала находится точка пересечения луча с плоскостью треугольника, если луч пересекает плоскость треугольника, то ищется точка пересечения луча с треугольником.

На первом этапе сначала вычисляется угол между нормалью треугольника и направляющим вектором луча. Если скалярное произведение  $n_{tr} \bullet dir_{ray} = 0$ , то луч параллелен плоскости треугольника. В этом случае считается, что треугольник лучом не пересекается. Далее вычисляется расположение точки пересечения луча с плоскостью треугольника относительно начала луча

$b = (n_{tr} \bullet pos_{ray}) - d$ , где  $d$  – расстояние от начала координат до данного треугольника, затем вычисляется параметр  $t$ , соответствующий  $t$  в (3)

$$t = b/a,$$

если  $t < 0$ , точка пересечения с плоскостью треугольника не принадлежит лучу (3). Затем вычисляется вектор точки пересечения луча с плоскостью треугольника  $I_p$ .

$$I_p = pos_{ray} + dir_{ray} * t$$

Затем координаты вершин треугольника преобразованием сдвига на вектор  $-I_p$  переводятся в систему координат, начало которой находится в точке  $I_p$ . Далее, все вершины проецируются на плоскость  $ХОУ$ , и проверка на пересечение идет уже только с проекциями вершин. Она производится в два этапа

1. Если все  $x$  или все  $y$  координаты проекции вершин одного знака, то луч не пересекает треугольник.
2. Если 1 не выполняется, то из точки  $O$  в направлении оси  $x$  выпускается луч и производится тест на пересечение им ребер треугольника; если количество пересечений луча с ребрами равно 1, луч пересекает треугольник, и в этом случае вектор точки пересечения лучом треугольника равен  $I_p$ , иначе луч не пересекает треугольник.

#### 4. ЗАКЛЮЧЕНИЕ

Программный комплекс подготовки баз данных представляет собой главный модуль, осуществляющий построение моделей для системы Voxel-Volumes из vdm (voxel digital model file) файла, написанный на языке C++ и подключаемый модуль для bCAD написанный на языке Java. Подключаемый модуль для bCAD в соответствии с концепцией программной системы осуществляет процесс получения геометрического описания объектов в bCAD и сохранение данных в формате vdm (voxel digital model file). Данный файл является стандартным файлом для главного модуля и используется для построения моделей для системы Voxel-Volumes.

#### 5. БИБЛИОГРАФИЯ

- [1] S.I. Vyatkin, B.S Dolgovesov, A.V. Yesin, et al. Voxel Volumes volume-oriented visualization system, International Conference on Shape Modeling and Applications (March 1-4, 1999, Aizu-Wakamatsu, Japan) IEEE Computer Society, Los Alamitos, California, 1999, pp. 234-241.
- [2] Vyatkin S.I., Dolgovesov B.S., Yesin A.V. Non-Polygonal Graphics and Volume-Oriented Rendering // Proceedings of the IASTED International Conference on Automation, Control, and Information Technology 2002, Novosibirsk, Russia, pp. 447-451.
- [3] Вяткин С.И., Долговесов Б.С., Есин А.В. и др. Геометрическое моделирование и визуализация функционально-заданных объектов // Автометрия. 1999. №6. стр. 65-73.
- [4] Вяткин С.И., Долговесов Б.С. Поверхности и патчи свободных форм на основе скалярных и аналитических функций возмущения// Труды 12-й Междунар. конф. "Графикон-2002". Нижний Новгород, 2002. <http://www.graphicon.ru/2002/papers.htm>
- [5] Sergei I. Vyatkin, Boris S. Dolgovesov "Voxel-Based Terrain Generation Using Scalar Perturbation Functions", 13-th International Conference on Computer Graphics GraphiCon '2003, Moscow, pp. 165-168. <http://www.graphicon.ru/2003/Proceedings/technical.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](mailto: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](mailto:bsd@iae.nsk.su)

## Database Creation Based on Scalar Perturbation Functions

### Abstract

The problem of database creation based on scalar perturbation functions is considered. The method of database creation is considered as well.

**Keywords:** Database creation, scalar 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](mailto: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](mailto:bsd@iae.nsk.su)