

**А.В. ТЕРЕХИН,  
С.В. САВИЧЕВА**  
**Алгоритм формирования  
косоугольной проекции  
трехмерного объекта по модели  
окто-дерева**

*УДК 004.932.2*

Муромский институт  
(филиал) ФГБОУ ВПО  
«Владимирский  
государственный  
университет имени  
А.Г. и Н.Г. Столетовых»,  
г. Муром

*В статье предложен алгоритм построения косоугольной проекции трехмерного объекта по его модели окто-дерева, приводятся примеры работы реализованной подсистемы.*

Компьютеры стали незаменимым инструментом моделирования и симуляции. С увеличением вычислительной мощности, пользователи и приложения требуют все большие уровни реализма в этих областях. Наиболее ярко эта тенденция видна в компьютерной графике, где появляется возможность моделирования все более сложных геометрических объектов и разнообразных физических процессов. Существующие подходы моделирования деформируемых объектов позволяют использовать как геометрические методы, так и методы, использующие физические законы, учитывающие свойства материи, влияние внешних сил и ограничения окружающей среды при моделировании деформации объекта [1].

Существует множество способов представления трехмерных (3D) объектов, такие как полигональные модели, воксели, CSG (конструктивная стереометрия), и т.д. Полигональные модели — наиболее распространенный формат среди других в силу их повсеместного использования в компьютерной графике и видеоиграх, а также поддержке этого представления аппаратными средствами графических процессоров GPU. У вокселей есть много преимуществ по сравнению с полигональными моделями. Основным преимуществом является неограниченный уровень детализации. Основным недостатком вокселей, препятствующий широкому применению, является их статичность.

Компьютерные системы, в которых пользователь может взаимодействовать с виртуальным миром, стали очень популярными в последние годы. Помимо компьютерных игр, есть большое количество других примеров интерактивных виртуальных сред, таких как средства моделирования полета, медицинские и авто симуляторы.

Чтобы предоставить пользователю ощущения реального физического мира, важно, чтобы объекты вели себя так, как они ведут себя в реальном мире. Виртуальные сцены, которые встречаются в компьютерных играх, включают в себя большое количество объектов. Чтобы смоделировать физические объекты – особенно, когда эти объекты деформированы или разрушены на части – могут эффективно использоваться воксели [2-6].

Целью данной работы является реализация и исследование алгоритма формирования окто-дерева по его трем эталонным проекциям, и генерация косоугольной проекции трехмерного объекта под заданным углом.

### Алгоритм

1. Объем сцены представляется в виде куба.
2. Этот куб разбивается на восемь (окта) подобластей (рис. 1) делением пополам по трем координатам [7].
3. Каждая подобласть аналогично может быть разбита на восемь подобластей и т.д., до необходимого уровня детализации объема. При разбиении объема сцены возникают ситуации, когда некоторые подобласти не содержат информации (пустые области), дальнейшее их разбиение не имеет смысла и занимает избыточную память при представлении в окто-дереве.

Та же ситуация возникает, когда подобласть содержит полностью однородную информацию. Например, подобласть охватывает внутренний объем сплошного тела.

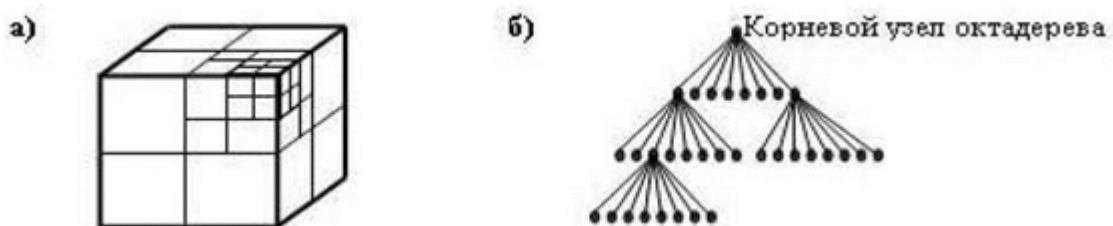


Рис. 1. Рекурсивное разбиение объема отображения на октанты а) исходная исследуемая область; б) схема построения окто-дерева.

4. Определяется три типа узлов (подобластей) для окто-дерева:
- full(полный);
  - empty (пустой);
  - half ("частично" заполненный).

Полный тип (full) характеризует узел, который полностью заполнен, содержит в себе информацию и далее не разбивается. Такой full-узел окто-дерева соответствует трехмерной подобласти, которая принадлежит полностью по своему объему какому-либо объекту сцены. Для данного узла характерно одинаковое представление какого-либо свойства или их совокупности (цвет и т. д.).

Пустой тип (empty) характеризует узел, который ничем не заполнен, не содержит в себе информацию и далее не разбивается. Такой empty-узел окто-дерева соответствует трехмерной подобласти, которая не содержит в своем объеме какой-либо объект сцены или его часть.

Частично заполненный тип (half) следует понимать как узел, который содержит в себе частично заполненное и частично пустое пространство. Такой half-узел разбивается дальше на более мелкие октанты и потому опять может содержать в себе узлы трех типов. Каждая из подобластей (октант) представляется узлом окто-дерева следующего уровня детализации. В half-узле хранится информация, которая указывает, какие типы подузлов соответствуют данному узлу.

Для работы с окто-деревом разработан алгоритм нумерации октантов. Суть его в следующем: каждому октанту куба отображения соответствует свой диапазон изменения координат. Если сравнить опорную координату подпространства со средней точкой куба отображения, то получим комбинацию из трех логических переменных  $C_0$ ,  $C_1$ ,  $C_2$ , где  $C_0$  связана с координатой  $x$ ,  $C_1$  - с координатой  $y$ ,  $C_2$  - с координатой  $z$ .

$$C_0 = (X \geq \frac{\Delta d}{2} + x), C_1 = (Y \geq \frac{\Delta d}{2} + y), C_2 = (Z \geq \frac{\Delta d}{2} + z)$$

где  $(X, Y, Z)$  - опорные координаты подпространства;  $(x, y, z)$  - опорные координаты куба отображения;  $\Delta d$  - линейный размер куба отображения.

Каждому октанту из восьми соответствует свой набор значений  $C_0C_1C_2$ , который будет являться номером октанта в нумерации от 0 до 7 (рис. 2).

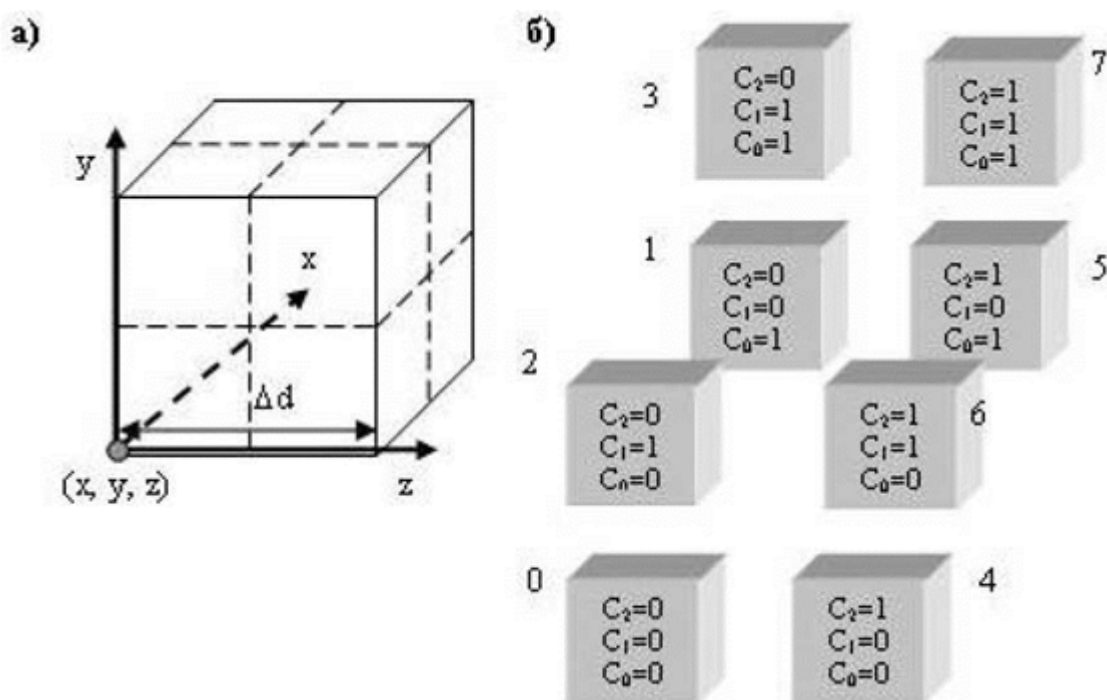


Рис. 2. Нумерация октантов подпространства: а) исходная исследуемая область (куб отображения); б) нумерация октантов и величины логических переменных  $C_0$ ,  $C_1$ ,  $C_2$ .

Октанное дерево объема проектирования хранится в двух линейных массивах. В одном массиве (Н) хранятся half-узлы - данные структуры дальнейшего разбиения на подузлы, в другом (F) хранятся full-узлы – ссылки на описание атрибутов.

В массиве Н half-узлы хранятся по уровням детализации. Первым элементом массива Н является узел верхнего уровня детализации (корневой узел окто-дерева). Причем если этот корневой узел имеет full или empty тип, то он все равно представляется как half-узел, в котором все подузлы имеют одинаковый тип full или empty. Далее в массиве Н идут half-узлы следующего уровня детализации и т. д.

Поскольку в half-узле не хранятся ссылки на дочерние узлы, то возникает задача вычисления функции перехода от текущего узла к дочернему.

Для функции перехода от одного half-узла к дочернему half-узлу необходимо знать количество представлений half-узлов, встретившихся с начала массива  $H$  до представления текущего half-узла. Для обращения к full-узлу в массиве  $F$  необходимо знать количество представлений full-узлов, встретившихся с начала массива до представления текущего full-узла (рис. 3).

Для уменьшения вычислений в функции перехода заводится двумерный массив с параметром  $level$ , в котором хранятся номера начальных и конечных элементов full и half для уровней в окто-дереве, где  $level$  - количество уровней в окто-дереве.

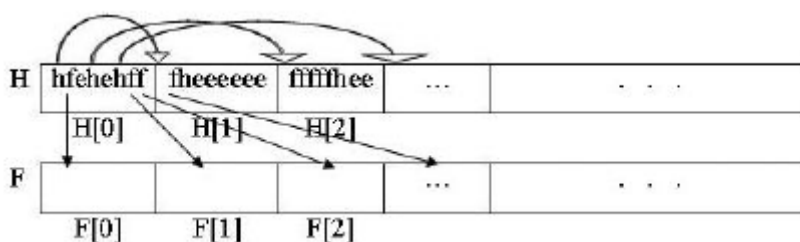


Рис. 3. Переход к соответствующему описанию узла в окто-дереве

При таком представлении (рис. 4), зная текущий уровень детализации, можно определить, в какой диапазон элементов будет следующее обращение. Вычисляя количество вхождений кода для этого уровня с начала и до текущего элемента, определяют смещение от начала диапазона.

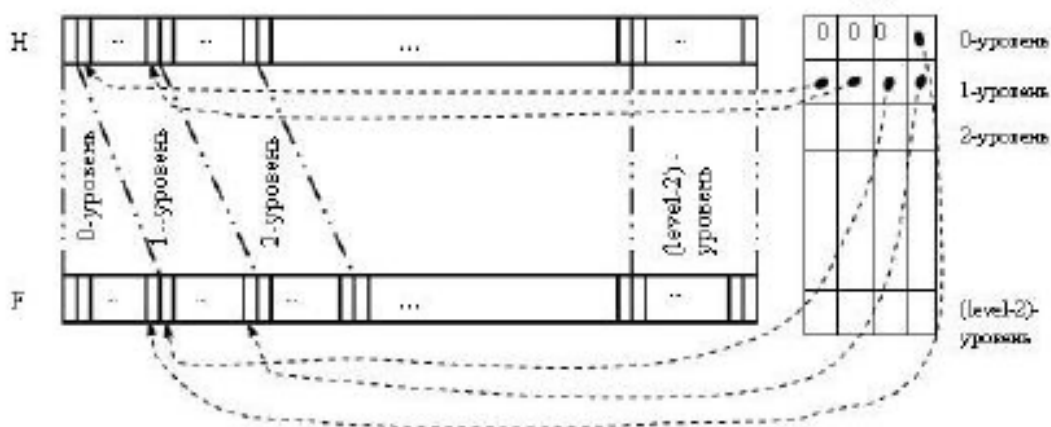


Рис. 4. Общая структура данных окто-деревя.

### Формирование проекционного изображения вокселя

Выбирается ортогональная проекция на плоскость отображения (рис. 5). Тогда важно не место наблюдателя, а вектор проецирования, который будет всегда перпендикулярен плоскости проецирования.

Если в самом начале определить видимые грани для куба (вокселя) и занести эту информацию в какую-нибудь общую управляющую структуру (ОУС), то можно остальные воксели выводить по этой структуре (аналогично примененному подходу в сортировке вокселей), т. к. все воксели ориентированы в пространстве одинаково.



Рис. 5. Схема построения ортогональной проекции

Тогда при изменении положения наблюдателя (вектор проецирования тоже изменяется) простым перерасчетом общей управляющей структуры изменяется способ отображения всех вокселей.

### Исследование

В ходе исследования были построены проекции объектов, расположенных под разными углами обзора (рис. 6 – 8).

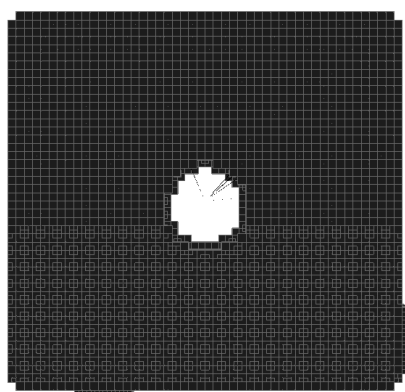


Рис. 6. Октодревесная модель объекта 1

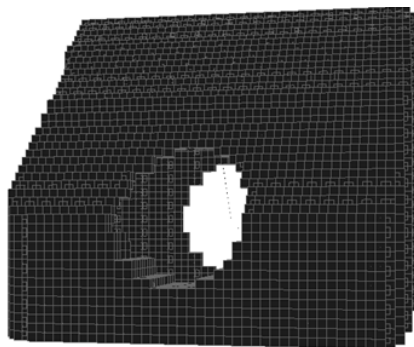


Рис. 7. Октодревесная модель объекта 2

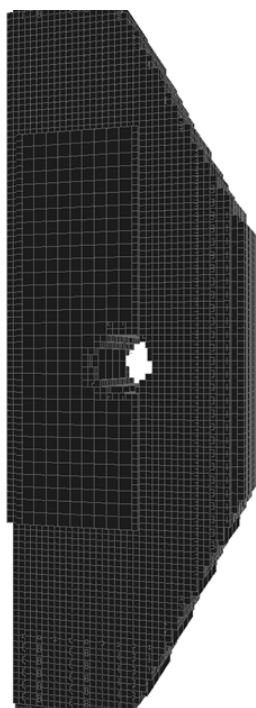


Рис. 8. Октодревесная модель объекта 3

### **Заключение**

В ходе работы был исследован и реализован алгоритм формирования окто-дерева по его трем эталонным проекциям, и формирование проекции объекта под определенным углом.

Данная разработка может быть использована в СТЗ для обработки снимков объектов, которые расположены к камере не перпендикулярно, а под произвольным углом.

## Литература

1. Прэтт У.: Цифровая обработка изображений: Пер. с англ. – М. Мир, 1982.
2. Цыганков М.А.: Эффективная визуализация объемов с помощью октантных деревьев. // Программирование. – 1999. – № 3. – С. 32-42.
3. Волобой А.Г.: Метод компактного хранения октантного дерева в задаче трассировки лучей. // Программирование. – 1992. – № 1. – С. 21-27.
4. Jackins, C.L., Tanimoto, S.L.: Octrees and Their Use in Representing Threedimensional Objects. CGIP. 14 (1980) p. 249-270
5. Noborio, H., Fukuda, S., Arimoto, S.: Construction of the Octree Approximating Three-dimensional Objects by Using Multiple Views. IEEE Trans. PAMI. Vol. 10 No. 6 (1988) p. 769-782
6. Chien, C.H., Aggarwal, J.K.: Volume/Surface Octrees for The Representation of 3-D Objects. CGIP. Vol. 36 (1986) p. 100-113
7. Вяткин С.И.: Эффективные растеризационные методики с рекурсивным делением плоскости изображения и объектного пространства. //www.ixbt.com

E-MAIL: TEREHIN\_MUROM@MAIL.RU

E-MAIL: SAVICHEVA.SVETLANA@GMAIL.COM