

**Е.А. Башков** (д-р техн. наук, проф.), **О.А. Авксентьева** (ассист.),  
**Н.О. Харченко** (магистрант)

Донецкий национальный технический университет  
bashkov@pmi.dgtu.donetsk.ua

## **МЕТОД ВОКСЕЛЬНОГО РАЗЛОЖЕНИЯ ПРОИЗВОЛЬНОЙ ДУГИ ДЛЯ ОБЪЕМНЫХ 3D ДИСПЛЕЕВ**

Рассмотрены вопросы выбора оптимального метода генерации дуги, заданной произвольными параметрами в трехмерном пространстве для 3D дисплеев, с учетом цели минимизации количества сложных и комплексных математических преобразований. Приведен алгоритм решения поставленной задачи.

**воксель, алгоритм, растровое разложение, 3D дисплей**

### ***Введение***

Повышение информативности и улучшение изобразительных возможностей визуального канала обмена данными между компьютером и человеком приводит к созданию трехмерных устройств отображения. Современная классификация [1] 3D устройств визуализации объектов и изображений выделяет как особый класс системы, построенные на базе объемных технологий. Такие устройства позволяют создавать виртуальное изображение объектов в трехмерном пространстве, причем, эффект трехмерности они обеспечивают для множества точек наблюдения, расположенных в широком диапазоне как по горизонтали, так и по вертикали.

Создание и использование такого рода устройств требует разработки специальных аппаратных и программных средств. Особенностью устройств отображения на базе объемных технологий является наличие объемного воксельного запоминающего устройства - аналога растровой памяти двумерных устройств отображения. При генерации трехмерного изображения в воксельной памяти программными средствами создается «вокселизированная» модель реальных объектов, состоящая из совокупности трехмерных графических примитивов: отрезков трехмерных прямых, трехмерных плоскостей, дуг, окружностей, сферических треугольников, эллипсоидов и т.п [2]. В данной работе предлагается и исследуется ускоренный метод воксельного разложения произвольной дуги в трехмерном пространстве.

## **Постановка задачи растрового разложения дуги**

Следуя [3] положим, что некоторая часть трехмерного евклидова пространства, которое отображается 3D дисплеем, имеет вид трехмерного параллелепипеда:

$$\Omega \in R^3, 0 \leq x \leq X, 0 \leq y \leq Y, 0 \leq z \leq Z.$$

С учетом возможности масштабирования, будем считать, что  $X=Y=Z=H$ , то есть  $\Omega$  – трехмерный куб. Положим, что  $\Omega$  заполнена вокселями – атомарными элементами, которые отображаются 3D дисплеем. Определим воксель как куб с единичной стороной. Центры «соседних» вокселей удалены друг от друга по координатным осям на единичное расстояние. Множество вокселей, заполняющих  $\Omega$  можно представить как трехмерный массив вокселей  $V_{i,j,l}$ , где  $i,j,l$  – индексы, принимающие значения  $0, \dots, \text{int}(H)$ .

Координаты центра вокселя определяются как

$$x_{i,j,l} = 0.5 \cdot (i+1), y_{i,j,l} = 0.5 \cdot (j+1), z_{i,j,l} = 0.5 \cdot (l+1). \quad (1)$$

Соответственно, воксель, которому принадлежит точка  $L \in \Omega$  координатами  $X_L, Y_L, Z_L$ , имеет индексы

$$i = \text{int}(X_L), j = \text{int}(Y_L), l = \text{int}(Z_L). \quad (2)$$

В  $\Omega$  задана дуга своими начальной точкой  $N=[X_N, Y_N, Z_N]$ , точкой центра дуги  $C=[X_C, Y_C, Z_C]$ , некоторой точкой  $P=[X_P, Y_P, Z_P]$ , однозначно определяющей плоскость, в которой лежит дуга, и углом дуги  $\alpha$ .

Необходимо найти такую последовательность вокселей, каждый из которых лежит на минимальном расстоянии от дуги, и плотно заполняет промежуток между точкой  $N$  и конечной точкой генерируемой дуги и, будучи отображенной на экране 3D дисплея, воспринимается наблюдателем как дуга в трехмерном пространстве (с учетом масштабирования).

Определим два разных вокселя  $V_U$  и  $V_W$  как соседние, если у них

- хотя бы один индекс не равен;
- индексы отличаются не более чем на 1.

Обозначим воксели  $V_N$  и  $V_K$  как начальный и конечный воксели растрового разложения дуги. Обозначим воксель  $V_C$  как воксель, содержащий центр дуги. Индексы вокселей  $V_N, V_C$  и  $V_P$  определяются по (2).

Задача растрового разложения дуги формулируется как задача определения множества вокселей  $V^{<q>}$ ,  $q=1,2,\dots,M$ , к которому принадлежат воксели  $V_N=V^{<1>}$  и  $V_K=V^{<M>}$ , каждый из которых (кроме начального и конечного) имеет два и только два соседа, центр каждого из которых лежит

на минимальном расстоянии до дуги и количество вокселей во множестве минимально.

В [3] предложен метод построения такого множества, суть которого кратко заключается в следующем. Предполагается, что генерируемая дуга лежит в первом октанте. Пусть найден  $q$ -й воксель  $V^{<q>}$  последовательности. Требуется определить следующий  $q+1$ -й воксель  $V^{<q+1>}$ . Рассматриваются семь соседних с  $V^{<q>}$  вокселей - претендентов  $V_{(k)}^{<q+1>}$ ,  $k=1,2,\dots,7$  в октанте, соответствующем направлению вектора, исходящего из центра  $V^{<q>}$  и параллельного касательной в точке дуги, ближайшей к центру  $V^{<q>}$  (рис.1).

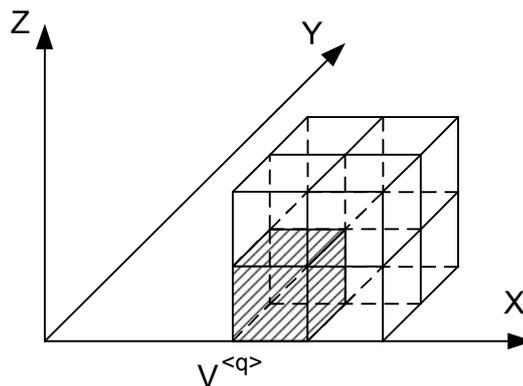


Рисунок 1 – Воксели-претенденты для первого октанта

Окончательный выбор следующего вокселя  $V^{<q+1>}$  зависит от значений двух показателей погрешности вычислений разрабатываемого алгоритма:

$E1_{(k)}^{<q+1>}$  - расстояния от центра  $k$ -ого вокселя-претендента до плоскости дуги, где  $k=1,2,\dots,7$ ;

$E2_{(k)}^{<q+1>}$  - разницей между радиусом дуги и расстоянием от центра  $k$ -ого вокселя-претендента до центра окружности, где  $k=1,2,\dots,7$ ;

Окончательно,  $V^{<q+1>}$  в [3] предлагается выбирать как воксель-претендент с минимальной суммой  $E1_{(k)}^{<q+1>}$  и  $E2_{(k)}^{<q+1>}$ .

Разница между длиной радиуса дуги и расстоянием от вокселя-претендента до центра дуги рассчитывается как:

$$E1_{(k)}^{<q+1>} = R - \left| CV_{(k)}^{<q+1>} \right|, \quad (3)$$

где  $R$  – радиус дуги,  $\left| CV_{(k)}^{<q+1>} \right|$  - модуль вектора, соединяющего точку центра дуги и  $k$ -й воксель-претендент,  $k=1,2,\dots,7$ . Радиус дуги рассчитывается как

$$R = \sqrt{(X_N - X_C)^2 + (Y_N - Y_C)^2 + (Z_N - Z_C)^2}. \quad (4)$$

Расстояние до плоскости от каждого вокселя-претендента рассчитывается как

$$E2_{(k)}^{<q+1>} = (n_X \cdot V_{X(k)}^{<q+1>} + n_Y \cdot V_{Y(k)}^{<q+1>} + n_Z \cdot V_{Z(k)}^{<q+1>} + D) \cdot \frac{1}{|\vec{n}|}, \quad (5)$$

где  $n_X, n_Y, n_Z$  – координаты вектора нормали к заданной плоскости,  $D$  – коэффициент уравнения плоскости,  $V_{X(k)}^{<q+1>}, V_{Y(k)}^{<q+1>}, V_{Z(k)}^{<q+1>}$  – координаты  $k$ -го вокселя-претендента,  $k=1,2,\dots,7$ ,  $|\vec{n}|$  – модуль вектора нормали к заданной плоскости.

### **Модифицированный алгоритм 3D растрового разложения дуги**

С целью ускорения исходного подхода из [3] предлагается следующие модификации алгоритма.

Во-первых, можно существенно упростить вычисление координат конечной точки дуги с помощью умножения вектора

$$\overline{CN} = \{X_N - X_C, Y_N - Y_C, Z_N - Z_C\} = \{X_{CN}, Y_{CN}, Z_{CN}\} \quad (6)$$

на матрицу поворота  $M(\vec{n}, \alpha)$

$$\begin{bmatrix} \cos(\alpha) + (1 - \cos(\alpha))n_X^2 & (1 - \cos(\alpha))n_X n_Y - \sin(\alpha)n_Z & (1 - \cos(\alpha))n_X n_Z + \sin(\alpha)n_Y \\ (1 - \cos(\alpha))n_X n_Y + \sin(\alpha)n_Z & \cos(\alpha) + (1 - \cos(\alpha))n_Y^2 & (1 - \cos(\alpha))n_Y n_Z - \sin(\alpha)n_X \\ (1 - \cos(\alpha))n_X n_Z - \sin(\alpha)n_Y & (1 - \cos(\alpha))n_Z n_Y + \sin(\alpha)n_X & \cos(\alpha) + (1 - \cos(\alpha))n_Z^2 \end{bmatrix}, \quad (7)$$

где  $\vec{n} = \{n_X, n_Y, n_Z\}$  – вектор нормали к плоскости, т.е. единичный вектор оси вращения, а  $\alpha$  – угол дуги. В [3] эта задача решалась с помощью решения системы уравнений, прибегая к использованию циклического подбора значений и комплексных математических операций, вычислению квадратных корней.

Во-вторых, предлагается унифицировать определение множества вокселей-претендентов на каждом шаге алгоритма за счет вычисления матрицы приращений

Приоритет	1	2	3	4	5	6	7
M	$S_x$	0	0	$S_x$	0	$S_x$	$S_x$
	0	$S_y$	0	$S_y$	$S_y$	0	$S_y$

(8)

	0	0	$S_z$	0	$S_z$	$S_z$	$S_z$
--	---	---	-------	---	-------	-------	-------

с помощью координат вектора касательной к дуге, проведенного из центра текущего вокселя. Координаты вектора касательной  $\bar{k} = \{k_x, k_y, k_z\}$  находятся с помощью векторного произведения радиус-вектора  $\overline{CN}$  и вектора нормали к плоскости генерации дуги  $\bar{n}$ . Приращения для каждого вокселя-претендента определяются как

$$S_x = \text{sign}(k_x), S_y = \text{sign}(k_y), S_z = \text{sign}(k_z). \quad (9)$$

Координаты центров вокселей - претендентов  $V_{(k)}^{<q+1>}$ ,  $k=1,2...7$  при этом могут быть найдены как

$$\begin{bmatrix} V_{X(k)}^{<q+1>} \\ V_{Y(k)}^{<q+1>} \\ V_{Z(k)}^{<q+1>} \end{bmatrix} = \begin{bmatrix} V_X^{<q>} \\ V_Y^{<q>} \\ V_Z^{<q>} \end{bmatrix} + M^{(k)} + 0.5. \quad (10)$$

В-третьих, можно отказаться от вычисления квадратных корней при расчете расстояний  $E1_{(k)}^{<q+1>}$  и  $E2_{(k)}^{<q+1>}$ , т.к. абсолютное значение расстояний не важны для выбора следующего вокселя последовательности из множества вокселей-претендентов.

В-четвертых, для каждого вокселя в матрице приоритетов  $M$  введены приоритеты, уточняющие правила выбора следующего вокселя последовательности из множества вокселей-претендентов, имеющих одинаковые значения  $E1_{(k)}^{<q+1>}$  и  $E2_{(k)}^{<q+1>}$ , причем суммарное значение этих показателей минимально. В этом случае выбирается воксель, имеющий максимальную связность с  $V^{<q>}$ , т.е. воксель с максимальным количеством отличающихся индексов.

Предлагаемый модифицированный алгоритм воксельного разложения дуги в трехмерном пространстве, с учетом вышеизложенного, может быть записан на псевдоязыке из [4] следующим образом.

#### **Алгоритм воксельного представления дуги**

**Входные данные:**  $N$  начальная точка дуги,  $C$  центральная точка дуги,  $P$  точка на плоскости,  $\alpha$  угол дуги.

**Выходные данные:** массив вокселей  $V$

**Begin**

**Input**  $C, P, N, \alpha;$

Определение  $V_C, V_P, V_N$  по (2)  
 Вычисление координат векторов  $\overline{CP}, \overline{CN}$  по (6)  
 Вычисление координат вектора нормали  $\overline{n}$  к плоскости  
 Вычисление радиуса дуги по (4)  
 Вычисление координат точки К по (6) и (7)

**Repeat**

Вычисление координат вектора касательной  $\overline{k}$   
 Заполнение матрицы приращений **M** по (8) и (9)

**Do k = 1, 2 ... 7**

$V_{(k)}^{<q+1>} := V_N + M^{(k)} + 0.5$  по (1) и (10)

Вычисление погрешности  $E1_{(k)}^{<q+1>}$  по (3)

Вычисление погрешности  $E2_{(k)}^{<q+1>}$  по (5)

**End Do**

$MIN = MIN ( E1_{(k)}^{<q+1>} + E2_{(k)}^{<q+1>} )$

**Do k = 1, 2 ... 7**

**If**  $( E1_{(k)}^{<q+1>} + E2_{(k)}^{<q+1>} = MIN )$

**If** (Приоритет максимален)

$V_{points} := V_N + M^{(k)}$

$points := points + 1$

**End if**

**End if**

**End Do**

**Until**  $V_N \neq V_K$

**Output** V

**End**

### **Экспериментальные исследования модифицированного алгоритма 3D растрового разложения дуги**

Качество генерации трехмерного представления дуги естественно оценить через расстояния каждого вокселя воксельного разложения дуги до самой дуги [5]. Для этого на каждом шаге находится минимальное расстояние от центра вокселя последовательности до дуги. Это и есть погрешность  $O_q$  представления дуги для вокселя  $V^{<q>}$ . Тогда минимальная, средняя и максимальная ошибки воксельного разложения некоторой дуги могут быть найдены как

$$O_{\min} = \min_q (O_q), q = 1, 2 \dots M,$$

$$O_{av} = \frac{1}{I} \sum_{q=1}^M O_q,$$

$$O_{\max} = \max_q (O_q), q = 1, 2 \dots M.$$

Условие правильности работы генератора (все воксели разложения «пронизаны» заданной дугой) требует, чтобы  $O_{\max} < a \cdot \frac{\sqrt{3}}{2}$ , где  $a$  – сторона вокселя. При  $a=1$  максимальная ошибка по разложения должна не превышать  $\frac{\sqrt{3}}{2}$ . Экспериментальное исследование предложенных модификаций заключалось в генерации тысячи случайных дуг в  $\Omega$  с  $N=1024$  (1 мегавоксель). Эксперименты выполнялись на компьютере на базе Intel(R) Celeron CPU 2.3GHz, 1 Гб ОЗУ.

Обобщенные результаты приведены ниже:

Количество сгенерированных дуг = 1000
Количество точек = 1949224
Суммарная E1 = 435864.58976
Суммарная E2 = 433187.11029
Суммарная E3 = 696534.57655
=====
Средняя ошибка по радиусу (E1av) = 0.22361
Средняя ошибка по расстоянию до плоскости дуги (E2av) = 0.22224
Средняя ошибка по расстоянию до функции (E3av) = 0.35734
=====
MAX ошибка по радиусу (E1max) = 0.84506
MAX ошибка по расстоянию до плоскости дуги (E2max) = 0.84604
MAX ошибка по расстоянию до функции (E3max) = 0.86601
=====
MIN ошибка по радиусу (E1min) = 0.00010
MIN ошибка по расстоянию до плоскости дуги (E2min) = 0.00000
MIN ошибка по расстоянию до функции (E3min) = 0.00016
Время, затраченное на генерацию 1000 дуг = 20593 миллисекунд

Таблица 1. Сравнительные характеристики алгоритмов генерации

	Базовый алгоритм	Модифицированный алгоритм
Скорость генерации, вокселей / с	16695	96153
Количество экспериментов	1000	1000
Максимальная ошибка	0.84812	0.84506
Время генерации 1000 дуг, с	94.499	24.781
Количество рассчитанных вокселей	1577000	1577000

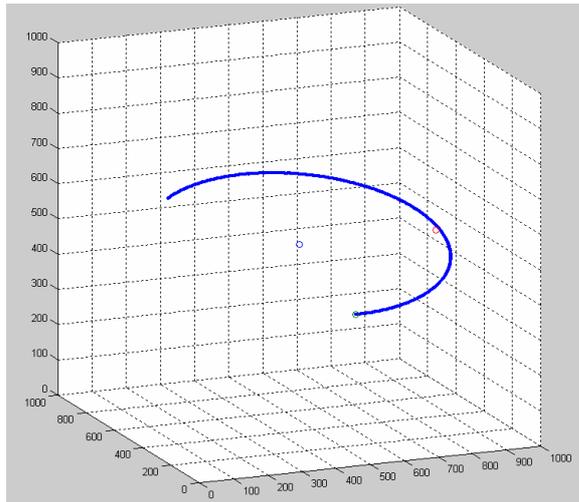


Рисунок 2 – Пример генерации дуги ( $\alpha = 217.90$ )

## **Выводы**

Анализ результатов работы 1000 экспериментальных запусков генерации дуги, заданной случайными начальными данными, показал, что:

- предложенный алгоритм генерирует дугу с ошибкой, не превышающей половину диагонали вокселя – 0.86602, что свидетельствует о корректности работы алгоритма.

- модификация алгоритма уменьшает время генерации в 5.7 раза. Такой показатель достигается за счет уменьшения количества процессорного времени, затраченного на вычисление сложных математических операций.

Предложенная модификация базового алгоритма 3D растрового разложения дуги повышает быстродействие, но для генерации сложных сцен с учетом требования реального времени этого недостаточно. Дальнейшие исследования следует направить на оптимизацию алгоритмов с целью сокращения временных затрат путем исследования возможности распараллеливания на специальных структурах вычислительных систем.

## **Список литературы**

1. Favolora G.E., Volumetric 3D Displays and Application Infrastructure // “Computer”, 2005, August, pp 37-44.
2. Башков Е.А. К построению генератора графических примитивов для трехмерных дисплеев / Е.А.Башков, О.А.Авксентьева, Аль-Орайкат Анас М. // Наукові праці Донецького національного технічного університету. Серія: Проблеми моделювання та автоматизації проектування динамічних систем. - 2008.- Вип. 7(150). – С. 203-214.
3. Мілютін М.О. Алгоритм генерації дуги для 3-D дисплеїв / М.О.Мілютін, Є.О.Башков, О.О. Авксентьева, Аль-Орайкат А. – Донецьк: ДВНЗ «ДонНТУ», 2009.
4. Препарата Ф. Вычислительная геометрия: введение; пер. с англ. / Ф. Препарата, М. Шеймос. – М.: Мир, 1989. - 478 с.
5. Авксентьева О.А. Улучшенный алгоритм генерации прямой для 3D – дисплеев / О.А. Авксентьева, Аль-Орайкат Анас, Д.И. Хлопов // материалы 14 – 10 международного молодежного форума «Радиоэлектроника и молодежь в XXI веке», 18 -20 марта 2010, Харьков 2010, сб. материалов 4.1. – Харьков: ХНУРЭ. 2010. – 527с.

**Е.А. Башков, О.А. Авксентьева, Н.О. Харченко**  
Донецкий национальный технический университет

**Метод воксельного разложения произвольной дуги для объемных 3D дисплеев.** В статье рассматриваются вопросы выбора оптимального метода генерации дуги, заданной произвольными параметрами в трехмерном пространстве для 3D дисплеев, с учетом цели минимизации количества сложных и комплексных математических преобразований. Рассматривается метод генерации заданного объекта на базе классического алгоритма Брезенхема, использующего поэтапное вычисление координат всех точек, принадлежащих множеству точек объекта. Выполнен анализ относительных и абсолютных погрешностей генерации объекта.

**воксель, алгоритм, растровое разложение, 3D дисплей**

**E.A. Bashkov, O.A. Avksentieva, N.O. Hartchenko**  
Donetsk National Technical University

**Method of Voxelized Decomposition of Random Arc for Volumetric 3D Displays.** The problems that are connected with the choice of optimal arc generation algorithm are considered in this article. The main aim of the considered algorithm was to minimize the amount of complex and complicated mathematical operations. This algorithm is based on classical Bresenham's algorithm which uses step-by-step calculation of all points coordinates that belong to the set of object's points

**Voxel, algorithm, rasterized decomposition, 3D display**