

АЛГОРИТМЫ ВЫВОДА НА ЭКРАН ВОКСЕЛЬНОГО ТРЕХМЕРНОГО ОБЪЕКТА

Айнур Жылкыбаевна Жумжуманова

магистрант

Павлодарского государственного университета им. С. Торайгырова

Түйіндеме

Мақалада воксельді үшөлшемді модельдің компьютер жадысында көріну әдісі қарастырылды. Үшөлшемді модельдерді визуалдайтын заманауи қолданбалы бағдарламалардың көбісі көпбұрыштардан тұратын графикалық объектілермен жұмыс жасайды және полигоналды үшөлшемді графикалардың пайда болу себепшісі болады.

Summary

In article the method of representation of vokselny three-dimensional models in memory of the computer is considered. The majority of the modern applied programs visualizing three-dimensional models work with the graphic objects consisting of polygons, and are the birth of polygonal three-dimensional graphics.

Большинство современных прикладных программ, визуализирующих трехмерные модели, работают с графическими объектами, состоящими из многоугольников (полигонов), и являются порождением полигональной трехмерной графики.

Воксельный объект в памяти компьютера представляет собой воксельный куб – это трехмерный массив, элементами которого являются цвета вокселей (рис. 1).

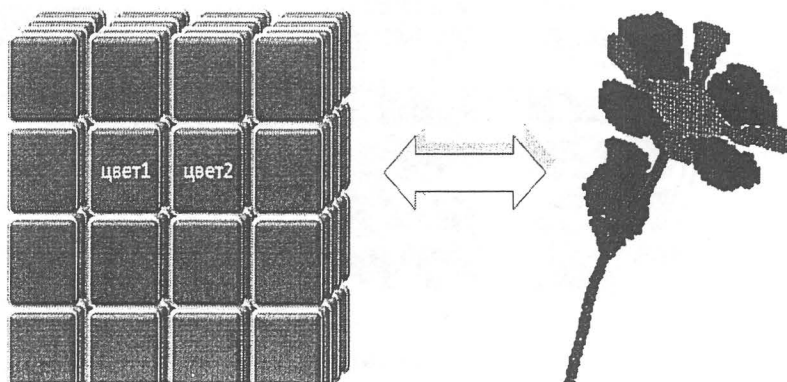


Рисунок 1

Положение элемента в массиве соответствует положению точки (вокселя) в пространстве. При таком способе хранения объектов обеспечивается высокая скорость операций добавления новых вокселей и считывания вокселей при рендеринге. Это позволяет обеспечить достаточную для удобной работы с приложением скорость обработки команд модулей редактирования и просмотра объектов.

Отрицательной стороной этой модели организации хранения объектов является большое количество оперативной памяти компьютера, требуемое для работы с воксельными объектами, но объема оперативной памяти большинства современных компьютеров (1 Гб) хватает для комфортной работы с несколькими воксельными кубами $256 \times 256 \times 256$, поэтому эта модель и используется.

Для оптимизации расходов оперативной памяти в будущем планируется использовать следующую модель хранения. Воксельный куб с четной длиной ребра будет представлен в памяти компьютера как октодереву. Вершиной дерева будет весь куб, разбитый на восемь равных частей (рис. 2). Каждая из частей указывает на подкуб, если в

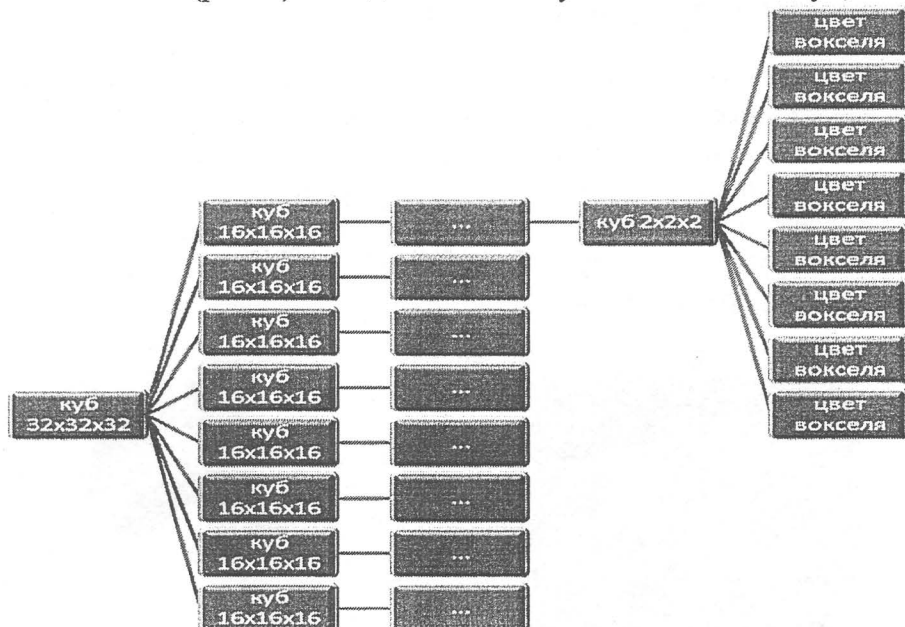


Рисунок 2

либо на null в противном случае. Далее подкуб делится на восемь равных частей аналогичным образом. Листьями октодерава являются цвета вокселей. При такой модели организации хранения объекта повышается рациональность использования вычислительных ресурсов компьютера, поскольку под те части куба, в которых вокселей не содержится, память не выделяется.

Изображением воксельного объекта на экране является его проекция. Для обеспечения отображения объекта в любом ракурсе реализованы алгоритмы вращения.

Поворот объекта в пространстве осуществляется путем поворота объекта в каждой из плоскостей XY, XZ, YZ вокруг осей вращения. Оси вращения рассчитываются как среднее арифметическое максимальной и минимальной координаты вокселя по каждой оси координат. Например, сдвиг по оси X_{sh} рассчитывается по следующей формуле: $X_{sh} = (X_{max} - X_{min})/2$, где X_{max} (X_{min}) – это крайние значения (максимальное и минимальное) по оси X . Затем точка пересечения осей вращения перемещается в начало координат путем вычитания из значений координат каждого вокселя x, y и z значений необходимых для этого сдвигов X_{sh}, Y_{sh} и Z_{sh} соответственно. После перемещения каждый воксель поворачивается во всех трех плоскостях. Например, для поворота объекта в плоскости XY на угол α необходимо для каждого вокселя умножить матрицу поворота $\begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$ на его координатный столбец $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$.

На экране каждый воксель отображается как объемный шарик. Эффект трехмерности шарика достигается путем затемнения цвета точек шарика по мере увеличения радиуса. При этом цвет разбивается на три компоненты R, G и B (красный, зеленый и синий соответственно) и значение каждой компоненты цвета рассчитывается по формуле $C_p = \left[C_c \cdot \left(1 - \frac{R_p}{R_s} \right) \right]$, где C_p – это искомое значение цветовой компоненты (R, G или B), C_c – значение цвета в центре

шарика, R_p – это расстояние от точки, цвет которой мы ищем, до центра шарика и R_s – радиус шарика.

Информация о каждой точке, выведенной на экран, заносится в Z-буфер – двухмерный массив с размером, соответствующим области экрана, на которую выводится изображение, содержащий расстояние от выведенной на экран точки до наблюдателя. Перед каждым выводом проекции объекта на экран этот массив инициализируется недостижимо большими для координаты Z вокселя значениями. При выводе точки на экран сопоставляется расстояние от наблюдателя до этой точки (координата Z вокселя, соответствующего этой точке) со значением элемента массива, положение которого соответствует координатам точки на экране. Если расстояние от наблюдателя до выводимой точки меньше соответствующего значения элемента массива, то точка выводится на экран, а это расстояние заменят старое значение элемента массива. В противном случае ничего не происходит. Также в массив записывается положение вокселя в воксельном кубе, соответствующего выведенной на экран точке. Это позволяет в процессе просмотра при любом угле наклона объекта изменять цвета его вокселей.

Список использованной литературы:

1. Адамс Дж., Роджерс Д. Математические основы машинной графики. – М.: Мир, 2001. – 604 с.