АЛГОРИТМЫ РАСТРОВОЙ ГРАФИКИ

Гуревич О. В , Шатилова О. О., Кукин Д. П., Коршикова Д. В.
Кафедра вычислительных методов и программирования, Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: {o.gurevich, o.shatilova, kukin, korshikova}@bsuir.by

Статья посвящена обзору алгоритмов построения растровой графики на примере алгоритма Брезенхема для прямой, а также его модификации для построения кривых второго порядка.

Введение

Компьютерные технологии в современном мире присутствуют практически во всех сферах жизнедеятельности человека. Практически вся визуальная информация является результатом проектирования с помощью 2D и 3D графики: телевизионная реклама, продукция кино- и телеиндистрии, игровых приложения. Компьютерная графика предоставляет человеку принципиально новые возможности работать с привычным для него трехмерным пространством, а не с его двумерными проекциями, что существенно упрощает порой создание физического экземпляра модели.

I. Виды компьютерной графики

В настоящее время принято выделять три вида графики: растровую, векторную и фрактальную. Для растровой графики основным элементом изображения является пиксель – минимальный по размеру логический элемент цифрового изображения, для векторной - линии. Для фрактальной графики характерно бесконечное повторение самоподобных геометрических фигур, каждая из которых уменьшается в масштабе. Иногда в отдельный вид графики выделяют трехмерную. которая отличается от двумерной тем, что подразумевает построение проекций трехмерной модели сцены на плоскость. Для описания геометрических объектов используют такие объекты и их свойства как прямые и плоскости, кривые, поверхности, кривизна линий на плоскости, криволинейные координаты. Поэтому смело можно утверждать, что компьютерная графика в части описания геометрических объектов в лице аналитической геометрии имеет достаточно мощный, готовый к использованию математический аппарат, позволяющий определять точки пересечения линий, поверхностей и линий и так далее.

II. Базовые алгоритмы растровой графики

Для рисования линий в растровой графике отвечают алгоритмы, которые учитывают соответствие координат начала и конца отрезка заданным параметрам, которые при отображении должны быть похожими на прямые. Однако, в силу дискретности дисплея этого трудно достичь, поэтому растровые алгоритмы преследуют цели воспроизведения максимального правдоподобия при минимальных ресурсозатратах. Самым прямым способом получения простейшего графического примитива, линии отрезка с координатами начала и конца (x_0, y_0) и(x_1, y_1) соответственно , является способ непосредственного получения координат через уравнение $y = F(x); y = y_0 + (x - x_0)\left(\frac{y_2 - y_1}{x_2 - x_1}\right)$. Для минимизации количества операций в цикле целесообразно получать коэффициент наклона линии до расчетов координат точек прямой $k = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)$. В таком случае в цикле получения координат пикселей остается вычисления $y+ = k * x$. Однако, такой способ является неточным за счет накопления ошибок округления при реализации операций деления и умножения, поэтому в 1962 американец Джек Брезенхем предложил алгоритм, который минимизирует накопления ошибок за счет введения параметра принятия решения и , соответственно, в цикле по расчету текущих координат остаются лишь целочисленные координаты. Положение пикселей на прямой определяется разбиением на единичные отрезки по координате x, если тангенс угла наклона прямой больше единицы по модулю, либо по y в обратном случае. В точке выбора координаты следующего пикселя проводится утрированный расчет расстояния до условно нижнего и верхнего предполагаемого пиксели и на основании полученных результатов производится выбор (см. рис. 1).

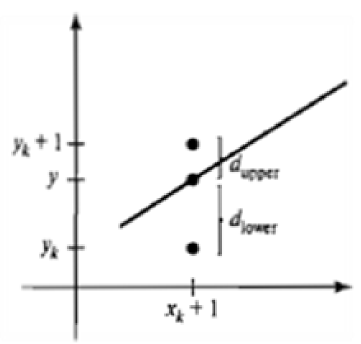

Рис. 1 – Графическая интерпретация расчета расстояния

В общем виде алгоритм Брезенхема для построения прямой с тангенсом угла наклона больше единицы по модулю выглядит следующим образом:

1. Вводятся два конца отрезка, помечая левый конец отрезка как (x0, y0);
2. Вычисляются постоянные deltaX, deltaY, 2 · deltaY и 2 · deltaY, 2 · deltaX и находится начальное значение параметра принятия решения p0 = 2 · deltaY − deltaX;
3. для каждого xk вдоль прямой, начиная с k = 1, производятся проверки: если pk < 0, то следующую точку следует изобразить на месте пикселя (xk+1, yk) и pk+1=pk + 2 · deltaY; если pk > 0, то следующую точку следует изобразить на месте пикселя (xk+1, yk+1) и pk+1 = pk + 2 · deltaY − 2 · deltaX;
4. п.3 выполняется deltaX − 1 раз.

Существует обобщение алгоритма Брезенхэма для построения кривых второго порядка, в частности окружности и эллипса. Окружность определяется как набор точек, равноудаленных от цента (x_c, y_c) на радиус r. Расчет координат можно производить непосредственно через уравнение окружнности либо находить через полярные координаты, однако такие подходы приводят к большому объему ресурсозатратных расчетов с вычислением либо корней квадратного уравнения либо к привлечению внешних функций по поиску синусов и косинусов.

Поэтому, взяв за основу алгоритм Брезенхема, был выведен алгоритм получения координат криволинейного участка, соответствующего окружности и эллипса с учетом того, что имеется кривизна.

В случае с окружностью принимается в расчет симметричность фигуры и все расчеты ведутся в первом октанте с последующим отображением полученных координат в оставшиеся семь октантов (см. рис. 2).

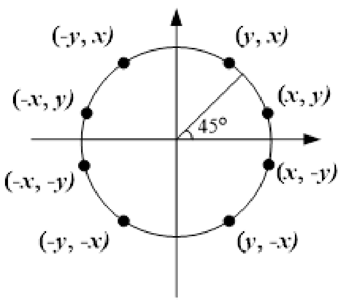

Рис. 2 – Симметрия октантов окружности

Для окружности параметр принятия решения рассчитывается для определения возможности перехода на нижнюю горизонталь по ординате.

Для эллипса измененный алгоритм Брезенхема учитывает изменение скорости роста соответствующей координаты. Все расчеты ведутся в первой четверти декартовой системы координат вначале на априорно изменяющемся x, а затем в момент, когда тангенс угла наклона касательной к эллипсу становится равным единице, ведущая координата заменяется на y и решение об изменении принимается уже по абсциссе. Далее полученные координаты отображаются в оставшиеся три четверти (см. рис. 3).

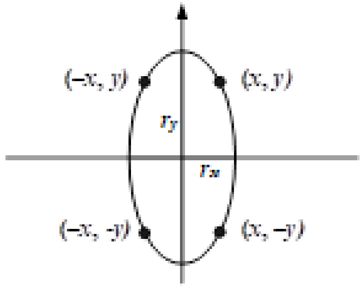

Рис. 3 – Симметрия эллипса

При реализации всеэ модификаций алгоритма Брезенхема для построения примитивов наряду с непосредственным расчетом координат соответствующих пикселей, как правило проводятся расчеты по аппроксимации цвета заливки примитива, чтобы сделать изображение максимально реалистичным.

III. Заключение

Описанные алгоритмы построения растровой графики не являются полным списком, но дают представление о том, как можно применять математические модели при проектировании компьютерной графики. Наряду с решением проблемы построения изображения, такие алгоритмы дают основу для создания алгоритмов сканирования, отсечения, удаления поверхностей и линий, а также закраски, определения текстур создаваемых геометрических объектов и управления их прозрачностью.

IV. Список литературы

1. Вольф, Д. OpenGL 4. Язык шейдеров. Книга рецептов / Д. Вольф // N. Engl. J. Med. – 2015. – 368 с.
2. Сиденко, Л. А. Компьютерная графика и геометрическое моделирование: учеб. пособие / Л. А. Сиденко. – СПб.: Питер, 2009. – 224 с.