Конические дуги - их свойства и метод растрирования

Чириков С.В. (Институт Точной Механики и Оптики - технический университет, Санкт-Петербург, Россия), Палташев Т.Т. (S3 Incorporated, USA).

Рассматриваются алгоритм и вычислительная структура генератора конических дуг. Предлагаемый алгоритм способен выполнять растрирование геометрически заданных конических дуг в многомерном пространстве с быстродействием алгоритма Брезенхема для окружностей. Коническая дуга, вписанная в треугольник общего положения, рассматривается в качестве графического примитива вывода. Предлагаемый алгоритм ориентирован на аппаратную реализацию, он представляет собой вычислительную структуру, состоящую из вычислительных элементов двух видов - одного конического и двух линейных интерполяторов, и допускает эффективную реализацию в виде БИС. Исследуются свойства конических дуг и возможности их применения в компьютерной графике.

Ключевые слова: *алгоритм Брезенхема, растрирование линий, лекальные кривые, L-кривые, коническая дуга, конический интерполятор, линейный интерполятор, кривая Безье, B-сплайн, конический сплайн.*

1. Обзор предшествующих работ.

Для растрирования квадратичных форм предложено несколько методов, однако полученные в результате различных подходов алгоритмы, с точки зрения быстродействия и аппаратных затрат, при их реализации, не имеют, на наш взгляд, существенных преимуществ, по сравнению с алгоритмами Брезенхема, Питтвея, Боттинга [2-4]. Во всех этих алгоритмах общим является задание растрируемой кривой алгебраическим уравнением F(x,y)=0, причем, функция F(x,y) должна представлять собой линейную [1], квадратичную [2] или кубическую [3] форму.

Предложенные в 70-х годах методы растрирования **2D** кривых также требовали алгебраического задания генерируемых кривых, причем, оказалось, что эти алгоритмы принимают компактный, удобный для аппаратной реализации вид, лишь для случаев вектора и окружности, а для растрирования иных кривых могла быть предложена лишь программная реализация. Поэтому в данной работе алгоритм Брезенхема будет рассматриваться в качестве эталона, относительно которого будет оцениваться быстродействие и аппаратные затраты предлагаемого алгоритма и вычислительной структуры.

В 80-х и 90-х годах разработке новых графических примитивов вывода, судя по снижению числа публикаций, стало уделяться меньше внимания, что, по-видимому, объясняется успехами в разработке СБИС, способных выполнять десятки миллионов операций в секунду. Задача генерации растрового образа кривой, особенно сложной кривой, все чаще стала решаться методом «грубой силы» - вычислением по формулам.

Перечислим достоинства и недостатки алгоритмов [1-4] растрирования алгебраических кривых. Безусловным достоинством этих алгоритмов является их простота и целочисленность. Для расчета положения одного пикселя требуется выполнить одну операцию сложения целых чисел для векторов, три операции для квадратичных форм и пять операций для кубических форм. К недостаткам следует отнести следующее. Все эти алгоритмы пригодны лишь для растрирования **2D** кривых, заданных аналитически в виде алгебраической формы, в то время как для реализации закраски необходимо выполнять генерацию линии не только в координатном пространстве, но и в пространстве цвета или в пространстве текстурных координат.

В задачах геометрического моделирования кривые и поверхности, как правило, задаются геометрически: заданием положения управляющих или интерполируемых точек, заданием направления касательных в точках сопряжения смежных кривых и т.д.. Получение алгебраического представления геометрически заданной кривой, за исключением очевидных частных случаев - векторов, окружностей, эллипсов, не является элементарной, с вычислительной точки зрения, задачей. Возможны ситуации, когда подготовка данных для генератора примитивов, т.е. его инициализация, потребует больше времени, чем собственно процесс генерации линии.

Алгоритмы Питтвея-Боттинга [2,3] являются обобщением методики Брезенхема для векторов [1] на случай квадратичных и кубических форм. Однако, получаемые в рамках этой методики, простые алгоритмы могут быть использованы для генерации лишь достаточно гладких **2D** кривых. При резком изменении кривизны генерируемой кривой, а также в точках возврата, эти алгоритмы теряют вычислительную устойчивость и этот их недостаток является неустранимым. Алгоритмы требуют также корректировки параметров при смене октанта, т.о., на их основе возможна лишь программно-аппаратная реализация генератора графических примитивов.

В данной работе предлагается методика растрирования конических дуг совмещающая в себе как преимущества методов растрирования **2D** квадратичных форм, так требования генерации кривых в цветовом пространстве RGB, что необходимо для получения реалистических изображений.

2. Генерация векторов в многомерном пространстве.

Геометрически вектор однозначно задается положением его концов $\mathbf{P}_{H,}$ $\mathbf{P}_K$. Для генерации **2D** векторов чаще всего используют алгоритм Брезенхема или метод цифрового дифференциального анализатора (ЦДА), которые на каждой итерации обеспечивают единичное или нулевое приращение координат вектора. При этом большая составляющая вектора получает приращение на каждой итерации, а меньшие лишь на некоторых. Представим многомерный вектор последовательностью **2D** векторов (X,R), (Y,R), (Z,R), ... , где $R=\max(|X_K-X_H|, |Y_K-Y_H|, |Z_K-Z_H|)$. Теперь алгоритм генерации многомерного вектора можно представить в виде нескольких параллельно работающих **2D** алгоритмов Брезенхема или ЦДА, каждый из которых за R итераций сгенерирует **1D** векторы X,Y,Z, ... , причем, каждая составляющая многомерного вектора будет генерироваться с постоянной скоростью, и на каждой итерации вектор будет получать, с точностью до погрешности дискретизации, фиксированное приращение координат.

В дальнейшем генератор векторов условимся называть линейным интерполятором (**LI**). Так как нам придется суммировать текущие координаты генерируемых векторов, то **LI** целесообразно реализовывать на основе метода ЦДА, поскольку этот метод в сущности, выполняет вычисления в формате с фиксированной точкой и каждая координата текущего пиксела до момента его отображения является его точной координатой. Таким образом, суммируя эти координаты, мы не суммируем погрешности округления, что неизбежно при использовании метода Брезенхема.

Будем считать, что **LI** выполняет одну итерацию, когда на его вход подается единичный управляющий сигнал. Введем управляющий параметр t=i/R, где i - количество выполненных **LI** итераций, тогда **LI** можно представить, как функциональный преобразователь с одним входом и одним выходом (см. Рис. 1).

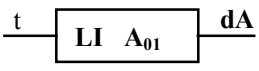

Рис. 1. Функциональная схема линейного интерполятора.

При изменении входного сигнала t от 0 до 1, с выхода преобразователя снимаются единичные приращения, которые, после их суммирования, дают значения текущей координаты вектора, изменяющейся с постоянной скоростью от 0 до X (Y,Z,...). Аналитически работу **LI** можно описать следующей формулой

$$\vec{A}(t) = \vec{A}_0 + \vec{A}_{01} \cdot t; \quad t \in [0,1] \qquad (1)$$

где $\vec{A}_{01} = \vec{A}_1 - \vec{A}_0$ - приращение координат вектора $\vec{A}_0 \vec{A}_1$.

t - параметр, пропорциональный числу итераций алгоритма ЦДА.

При аппаратной реализации **LI**, его быстродействие не будет зависеть от размерности пространства, а аппаратные затраты будут кратны размерности пространства.

3. Геометрический способ задания конической дуги.

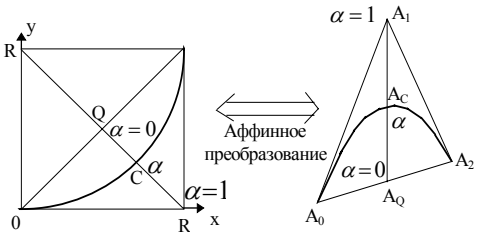

Рис.2. Алгебраический и геометрический способы задания конической дуги.

Дуга, вписанная в произвольный треугольник $A_0A_1A_2$ (см. Рис.2), может быть задана параметрической формулой

$$\vec{A}(t) = \vec{A}_0 + \vec{A}_{01}x(t) + \vec{A}_{12}y(t); \qquad (2)$$
$$x(t), y(t), t \in [0,1].$$

Строго говоря, функции $x(t), y(t)$ могут быть любыми, они могут быть даже начерчены от руки, необходимо лишь, чтобы они изменялись от 0 до 1, были непрерывными и могли быть представлены кодом Фримена, то есть, закодированы единичными приращениями. Следует отметить, что для того, чтобы коническая дуга могла быть сгенерирована без коррекции параметров, необходимо выбрать четырех точечную стратегию выбора шага, когда приращения координат возможны либо вдоль оси X, либо вдоль оси Y. Одновременное приращение координат X и Y недопустимо.

Если в качестве $x(t), y(t)$ выбрать квадратичную форму, то алгебраически она будет описываться следующим уравнением

$$x^2 + y^2 - 2Zxy - 2y(1+Z) = 0, \qquad (4)$$

где параметр Z определяет тип коники и является ее эксцентриситетом. Этот факт доказывается в курсе аналитической геометрии вычислением инвариантов квадратичной формы. Эксцентриситет коники однозначно определяет ее вид:

Эллипс: $Z \in (-1,1)$; (Окружность: $Z = 0$);
Парабола: $Z = 1$;
Гипербола $Z > 0$;

Полученное уравнение необходимо преобразовать к вид

$$nX^2 + nY^2 - 2mXY - 2YR(m+n) = 0, \qquad (5)$$

где $R = \max\{\|\vec{A}_{01}\|, \|\vec{A}_{12}\|\}$;
$\|d\vec{A}\| = \max\{|dX|, |dY|, \ldots\}$.
$X = x \cdot R$; $Y = y \cdot R$;
$Z = m / n$;

Для того, чтобы реализовать растрирование уравнение (5), все его коэффициенты должны быть целочисленными. Единственным рациональным коэффициентом этого уравнения является эксцентриситет Z и поэтому его следует аппроксимировать рациональной дробью $Z = m / n$. Коника является плоской кривой независимо от размерности пространства в котором она задается, поэтому для ее описания необходимо задать 5 граничных условий. В качестве таких условий мы выбираем

$\frac{dY}{dX} = 0$ - в начальной точке (0,0);

$\frac{dX}{dY} = 0$ - в конечной точке (R,R).

В качестве последнего условия достаточно задать положение любой точки на дуге, исключая ее начало и конец. Для определенности условимся выбирать в качестве такой точки, точку пересечения медианы A_1A_Q треугольника $A_0A_1A_2$ с вписанной в него кривой (см. рис.1).

Принимая во внимание симметрию коники (5), выразим ее эксцентриситет Z через положение точки $\vec{A}_C$, задаваемой коэффициентом α

$$\frac{m}{n} = \frac{\alpha^2 + 2\alpha - 1}{(1-\alpha)^2}, \qquad (6)$$

где: $\vec{A}_C = \vec{A}_Q(1-\alpha) + \vec{A}_1\alpha$;
$\vec{A}_Q = (\vec{A}_0 + \vec{A}_2)/2$; $Z = m/n$;
$\alpha \in [0,1]$.

Из данной формулы видно, что вид конической дуги однозначно задается положением ее средней точки $\vec{A}_C$:

Эллипс: $\alpha \in (0,1/2)$;
Парабола: $\alpha = 1/2$;
Гипербола: $\alpha \in (1/2,1)$.

При генерации коники (5) на каждой итерации будет получать единичное приращение либо координата X, либо Y. Из этого следует, что величина ($X_i + Y_i$) будет равна числу выполненных алгоритмом итераций. Поскольку алгоритм должен сгенерировать R единичных приращений вдоль оси X и столько же вдоль оси Y, то, очевидно, что для генерации коники (5) необходимо выполнить 2R итераций. В нормированном представлении уравнения коники (4) все параметры должны изменяться в диапазоне [0,1]. Введем параметр $t = (x + y)/2$, пропорциональный числу итераций, который также будет изменяться от 0 до 1 и получим явную зависимость нормированных координат $x(t), y(t)$ от нормированного числа итераций t.

$$x(t) = t - \sigma + sign(1-Z)\sqrt{\sigma(\sigma + 2t(1-t))}; \quad (7)$$

$$y(t) = t + \sigma - sign(1-Z)\sqrt{\sigma(\sigma + 2t(1-t))}, \quad (8)$$

где $x(t), y(t), t \subseteq [0,1]$,

$t = \frac{i}{2R}$; $i \subseteq [0,2R]$ - текущий индекс итерации алгоритма.

$$\sigma = \frac{1}{2} \cdot \frac{1+Z}{1-Z}; \ (Z \neq 1);$$

4. Конический интерполятор.

Для синтеза алгоритма растрирования квадратичной формы (5) с четырех точечной стратегией выбора шага, можно применить метод средней точки. Полученный по этой методике алгоритм приведен в виде блок-схемы на следующем рисунке. Инициализация данного алгоритма выполняется присвоением его переменным следующих начальных значений:

$u = 4n + 2m$;
$v = u - 4R(n+m)$;
$D = (n+m)(1-2R)$; $k_1 = 4n$;
$k_2 = 4m$.

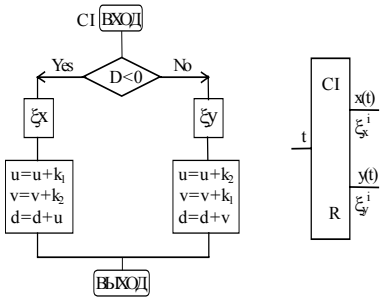

Рис. 3. Блок-схема алгоритма конического интерполятора и его функциональная схема.

На данном рисунке приведена только динамическая часть алгоритма, которая, при аппаратной реализации, должна быть реализована в виде самостоятельного функционального элемента. Этот элемент в дальнейшем будем называть коническим интерполятором, и обозначать **CI**. Конический интерполятор имеет один вход T, на которой подаются управляющие сигналы, и два выхода X и Y, с которых снимаются единичные управляющие сигналы.

Если просуммировать все выходные сигналы, X и Y по отдельности, то мы получим растровый образ вписанной в квадрат коники X(t), Y(t). Для генерации коники любого вида необходимо выполнить 2R итераций алгоритма, т.к. при выводе данного алгоритма использовалась четырех точечная стратегия выбора смежного пикселя. Этот момент принципиален для дальнейшего анализа.

5. Генерация конической дуги, вписанной в произвольный треугольник.

Как было показано выше коническая дуга, вписанная в треугольник, описывается уравнением

$$\vec{A}(t) = \vec{A}_0 + \vec{A}_{01}\frac{X(t)}{R} + \vec{A}_{12}\frac{Y(t)}{R}; \qquad (9)$$

$$x(t), y(t), t \in [0,1].$$

Представим это уравнение в инкрементной форме

$$d\vec{A}(t_i) = \frac{\vec{A}_{01}}{R}\cdot\varepsilon_x^i + \frac{\vec{A}_{12}}{R}\cdot\varepsilon_y^i, \qquad (10)$$

где $i \subseteq \{0,2R\}$ - номер итерации;

$$t_i = \frac{i}{2R};$$

$\varepsilon_x^i, \varepsilon_y^i \subseteq \{0,1\}$ - приращение координаты X,Y коники (5).

$d\vec{A}(t_i)$ - приращение координат генерируемой конической дуги на i-той итерации.

Из формулы (10) видно, что генерация конической дуги состоит в реализации одновременной генерации двух векторов $\vec{A}_{01}$ и $\vec{A}_{12}$. Каждый из этих векторов может быть реализован методом ЦДА, однако, скорость их генерации определяется управляющей коникой (5). Вектор $\vec{A}_{01}$ получает фиксированное приращение координат при изменении координаты X коники (5), а вектор $\vec{A}_{12}$ получает фиксированное приращение координат при изменении координаты Y коники (5). Таким образом, реализуется квази-одновременная генерация двух векторов с разными скоростями. Функциональная схема данного алгоритма приведена на рис. 4.

$$d\vec{A}(t) = \vec{A}_{01}x(t) + \vec{A}_{12}y(t)$$

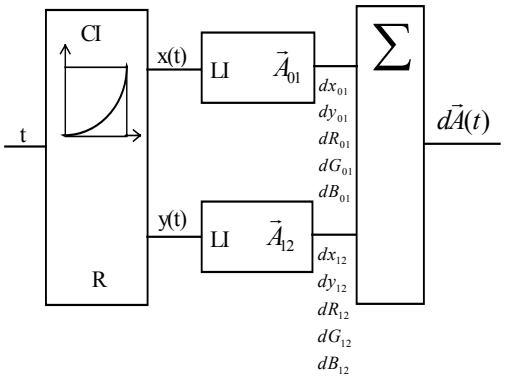

Рис.4. Функциональная схема генератора конических дуг.

То, что дуга начинается в начальной точке $\mathbf{A_0}$ и заканчивается в конечной точке $\mathbf{A_2}$ мы показали выше, однако, какая именно дуга будет сгенерирована, необходимо уточнить. Мы задали вид управляющей коники (5) двумя параметрами R и Z. Параметр R, как было показано выше, зависит от длины векторов и определяет число итераций алгоритма. Параметр Z задает вид кривой и связан с отношением трех точек α соотношением (6). Это отношение остается постоянным при аффинном преобразовании (9). Таким образом, задавая параметр α, мы однозначно задаем вид конической дуги.

Оценим аппаратные затраты на реализацию генератора дуг. Очевидно, что конический интерполятор по числу операций эквивалентен генератору окружностей Брезенхема [4], однако разрядность переменных должна быть выше, поскольку эксцентриситет Z задается рациональной дробью, и чем точнее его аппроксимация, тем больше будут переменные D, k_1, k_2, u, v. Из численных экспериментов видно, что алгоритм работает устойчиво при использовании 32-разрядного представления переменных. Линейные интерполяторы целесообразно реализовывать на основе алгоритма ЦДА, чтобы избежать суммирования ошибок округления. При этом также использовались 32-разрядное представление данных. 16 разрядов отводилось под целую часть числа и 16 разрядов под дробную часть. Отметим, что это в двое превышает необходимую разрядность алгоритма Брезенхема для векторов [1]. Поскольку каждый вектор дает приращение в пределах единицы растра, то при генерации двух векторов приращение будет вдвое превышать допустимый уровень. Чтобы избежать этого, параметр R следует по меньшей мере удвоить. При этом число итераций также удвоится, а скорость генерации дуги, по сравнению с генератором окружностей [4], будет вдвое меньше.

На рис.5 приведен пример задания семейства конических дуг вписанных в треугольник общего положения. Нижние дуги ($\alpha = \{1/9, 2/9, 3/9, 4/9\}$) являются эллиптическими. Верхние дуги ($\alpha = \{5/9, 6/9, 7/9, 8/9\}$) являются гиперболическими.

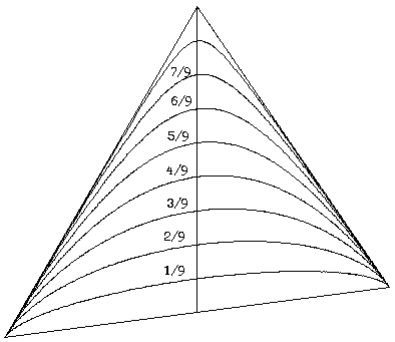

Рис.5. Семейство конических дуг, вписанных в треугольник.

6. Расщепление конической дуги пополам.

При отображении кривых, возникает ряд проблем, многие из которых решаются проще, если кривая представляет собой короткий незначительно изогнутый фрагмент, близкий по своим геометрическим свойствам к отрезку прямой. Для представления конической дуги последовательностью конических сегментов необходимо расщепить исходную кривую пополам и, если необходимо, повторить эту операцию над каждым сегментом требуемое число раз. При каждом расщеплении дуги ее тип остается неизменным, однако эксцентриситет коники приближается к единице, то есть, сегменты коники не только укорачиваются, но и приближаются к параболам. Докажем эти утверждения.

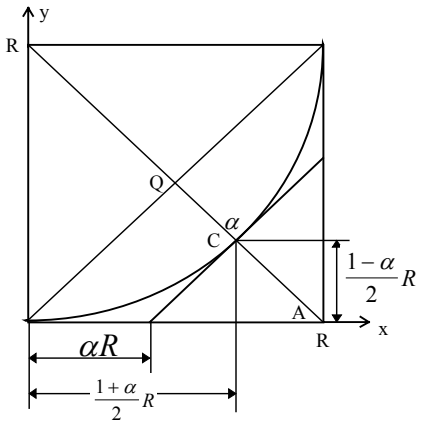

Рис. 6. Расщепление конической дуги пополам.

Рассмотрим данный рисунок. Поскольку, при аффинном преобразовании коники, отношение трех точек не изменяется, из этого следует, что все соотношения, полученные для данного частного случая, останутся справедливыми для конической дуги общего положения.

Исходная коника задается тремя точками {(0,0), (R,0), (R,R)} и имеет эксцентриситет Z, определяемый значением коэффициента α , задающего отношение трех точек **Q**,**C**,**A**. После расщепления образуются две одинаковые дуги, поэтому мы рассмотрим только первую, а все доказанные свойства автоматически будут справедливы и для второго сегмента. Вторичная коника будет задаваться будет задаваться тремя точками

$$\{(0,0),\ (\alpha R,0),\ (\frac{1+\alpha}{2}R, \frac{1-\alpha}{2}R)\}. \qquad (11)$$

Этот факт следует из очевидных геометрических соображений. Нашей целью является определение эксцентриситета вторичной кривой. Поскольку исходная и вторичная дуги являютсяониками, аналитически их можно представить, как конические дуги, вписанные в треугольники общего положения, координаты вершин которых нам известны.

$$\vec{A}(t) = \begin{pmatrix}0\\0\end{pmatrix} + R\begin{pmatrix}1\\0\end{pmatrix}x(t) + R\begin{pmatrix}0\\1\end{pmatrix}y(t); \qquad (12)$$

- исходная коника.

$$\hat{A}(t) = \begin{pmatrix}0\\0\end{pmatrix} + R\begin{pmatrix}\alpha\\0\end{pmatrix}\hat{x}(t) + R\frac{(1-\alpha)}{2}\begin{pmatrix}1\\1\end{pmatrix}\hat{y}(t); \qquad (13)$$

- вторичная коника.

Поскольку эти различные параметрические функции задают одну и ту же геометрическую линию $\vec{A}(t/2) \equiv \hat{A}(t)$, мы можем найти соотношения между управляющими кониками исходной и вторичной дуг.

$$x = \alpha\hat{x} + \frac{1-\alpha}{2}\hat{y}; \qquad y = \frac{1-\alpha}{2}\hat{y}; \qquad (14)$$

$$\hat{x} = \frac{x-y}{\alpha}; \qquad \hat{y} = \frac{2y}{1-\alpha}; \qquad (15)$$

Для обоих управляющих коник, также справедливо их алгебраическое представление

$$F(x,y) = x^2 + y^2 + 2Zxy - 2y(1+Z) = 0; \qquad (16)$$

$$F(\hat{x},\hat{y}) = \hat{x}^2 + \hat{y}^2 + 2\hat{Z}\hat{x}\hat{y} - 2\hat{y}(1+\hat{Z}) = 0. \qquad (17)$$

Подставив выражения $\hat{x}(x,y), \hat{y}(x,y)$ в квадратичную форму $F(\hat{x},\hat{y})$, мы находим следующие соотношения:

$$\hat{Z} = \sqrt{\frac{1+Z}{2}} = \frac{\alpha}{1-\alpha}; \qquad (18)$$

$$\hat{\alpha} = \frac{1}{1+\sqrt{2(1-\alpha)}}; \qquad (19)$$

Получение этих соотношений и было целью нашего анализа. Отметим еще раз, что коэффициент $\hat{\alpha}$, рассчитанный по приведенной выше формуле, будет корректным для любой конической дуги, вписанной в треугольник общего положения. Итеративно вычисляя $\hat{\alpha}$, мы можем видеть, что он стремится к единице. Отсюда делаем вывод, что по мере расщепления, сегменты любой конической дуги (эллиптической или гиперболической), приближаются к параболе, которая, в свою очередь, приближается к отрезку прямой. Найдем явное представление расщепленной конической дуги:

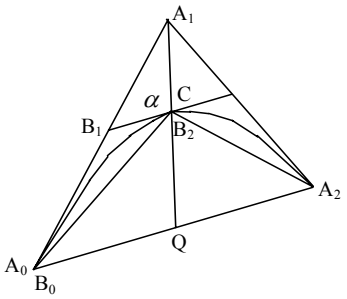

Рис.7. Расчет параметров расщепленной дуги.

$$\vec{A}(t) = \vec{A}_0 + \vec{A}_{01}x(t) + \vec{A}_{12}y(t); \qquad (20)$$

- исходная дуга.

$$\vec{B}(t) = \vec{B}_0 + \vec{B}_{01}\hat{x}(t) + \vec{B}_{12}\hat{y}(t); \qquad (21)$$

- вторичная дуга (1/2 исходной дуги).

Пересчет параметров расщепленной дуги:

$$\mathbf{B_0} = \mathbf{A_0}; \qquad (22)$$

$$\mathbf{B_1} = \mathbf{A_0}(1-\alpha)+\mathbf{A_1}\alpha; \qquad (23)$$

$$\mathbf{B_2} = \mathbf{Q}(1-\alpha)+\mathbf{A_1}\alpha; \qquad (24)$$

$$\hat{\alpha}=\frac{1}{1+\sqrt{2(1-\alpha)}}; \rightarrow \hat{Z}=\sqrt{\frac{1+Z}{2}}=\frac{\alpha}{1-\alpha}; \qquad (25)$$

Отметим, что при каждом расщеплении пополам, угол между касательными в начале и конце дуги уменьшается вдвое, что приводит к уменьшению нерегулярности ступенчатого эффекта и улучшает качество представления конической дуги на экране растрового дисплея.

7. Расщепление конической дуги в произвольной точке.

Для реализации операции клиппирования конической дуги, необходимо уметь расщеплять коническую дугу в произвольной точке, задаваемой параметром t. Рассмотрим следующий рисунок.

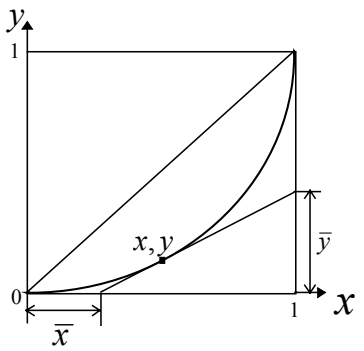

Рис. 8. Расщепление конической дуги вписанной в единичный квадрат.

Коническая дуга, вписанная в единичный квадрат, задается уравнением

$$x^2 + y^2 + 2Zxy - 2y(1+Z) = 0. \qquad (26)$$

Дифференцируя это уравнение по переменной x, находим производную $\frac{dy}{dx}$ равную тангенсу наклона коники в точке (x,y)

$$\frac{dy}{dx} = \frac{x+Zy}{(1+Z)-(y+Zx)} = \frac{\bar{y}}{1-\bar{x}}. \qquad (27)$$

Зная значения $\bar{x}$ и $\bar{y}$, составим уравнение прямой линии

$$\frac{y}{\bar{y}} = \frac{x-\bar{x}}{1-\bar{x}}. \qquad (28)$$

Решая систему из этих двух уравнений, находим выражение $\bar{x}$ и $\bar{y}$

$$\bar{x} = \frac{y(1+Z)}{x+Zy}; \qquad (29)$$

$$\bar{y} = \frac{x-y}{(1+Z)-(y+Zx)} = \frac{2y}{x+y} = \frac{y}{t}. \qquad (30)$$

Эта пара $(\bar{x},\bar{y})$ однозначно задает положение точки (x,y) коники и также может рассматриваться как координаты точки коники. Параметр t также можно выразить через координаты точки

$$t = \frac{1}{2}(x+y) = \frac{\bar{x}}{1+\bar{x}-\bar{y}}. \qquad (31)$$

Эксцентриситет Z является инвариантом коники и для любой точки кривой справедливо соотношение

$$Z = \frac{1}{2}\frac{x^2 + y^2 - 2y}{y(1-x)} = \frac{\bar{x}\bar{y} + \bar{y} - 2\bar{x}}{\bar{x}\bar{y} - \bar{y}}. \qquad (32)$$

Теперь запишем уравнение коники в диапазоне $[0, t]$, считая, что координаты конечной точки (x_t, y_t) и $(\bar{x}_t, \bar{y}_t)$ нам известны

$$x = \bar{x}\hat{x} + (x_t - \bar{x})\hat{y}; \qquad (33)$$

$$y = y_t \hat{y}, \qquad (34)$$

откуда получаем

$$\hat{x} = \frac{x}{x_t} - \frac{x_t - \bar{x}_t}{\bar{x}_t} \cdot \frac{y}{y_t}; \qquad (35)$$

$$\hat{y} = \frac{y}{y_t}, \qquad (36)$$

где: (x, y, Z) - исходная коника,

$(\hat{x}, \hat{y}, \hat{Z})$ - сегмент исходной коники.

(x_t, y_t) и $(\bar{x}_t, \bar{y}_t)$ - координаты конечной точки сегмента исходной коники.

Принимая во внимание, что алгебраическая форма уравнения коники не зависит от того, какой сегмент дуги мы рассматриваем, запишем уравнение коники $(\hat{x}, \hat{y}, \hat{Z})$ и найдем из него эксцентриситет $\hat{Z}$ сегмента дуги.

$$\hat{x}^2\bar{x}_t^2 + 2\bar{x}(x_t - \bar{x}_t)\hat{x}\hat{y} + (x_t - \bar{x}_t)^2\hat{y}^2 + 2Zx_t y_t \hat{x}\hat{y} + \\ + 2Zy_t(x_t - \bar{x}_t)\hat{y}^2 - 2(1+Z)y_t\hat{y} = 0; \qquad (37)$$

$$\hat{Z} = (1+Z)\frac{y_t}{\bar{x}_t^{\,2}} - 1. \qquad (38)$$

Вот и все. Нахождение $\hat{Z}$ и являлось целью проведенного анализа. Теперь мы можем любую коническую дугу (x, y, Z) представить в виде пары сопряженных конических дуг $(\hat{x}, \hat{y}, \hat{Z})$ и $(\hat{\bar{x}}, \hat{\bar{y}}, \hat{\bar{Z}})$, где $\hat{\bar{Z}}$ - эксцентриситет второго сегмента.

$$\hat{\bar{Z}} = (1+Z)\frac{1 - x_t}{(1 - \bar{y}_t)^2} - 1. \qquad (39)$$

$t, \bar{x}_t, \bar{y}_t$ можно рассчитать, используя конический интерполятор.

Теперь, принимая во внимание инвариантность отношения трех точек аффинного преобразования, можно привести полный алгоритм расщепления конической дуги общего положения.

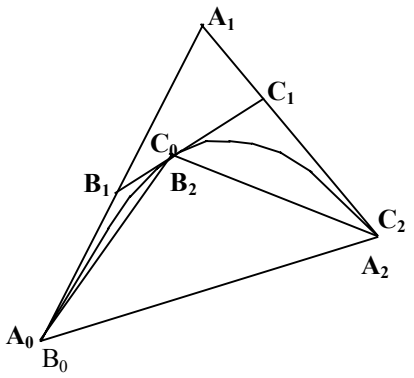

Рис. 9. Расщепление конической дуги вписанной в треугольник общего положения.

<**A**$_0$,**A**$_1$,**A**$_2$, Z> - исходная коническая дуга.
<**B**$_0$,**B**$_1$,**B**$_2$, Z_B> - вторичная коническая дуга.
<**C**$_0$,**C**$_1$,**C**$_2$, Z_C> - вторичная коническая дуга.
t - задает положение точки расщепления дуги <**A**$_0$,**A**$_1$,**A**$_2$, Z>.

$$\vec{B}_0 = \vec{A}_0; \quad \vec{C}_2 = \vec{A}_2; \qquad (40)$$

$$\vec{B}_1 = \vec{A}_0(1 - \bar{x}_t) + \vec{A}_1\bar{x}_t; \quad \vec{C}_1 = \vec{A}_1(1 - \bar{y}_t) + \vec{A}_2\bar{y}_t; \qquad (41)$$

$$Z_B = (1+Z)\frac{y_t}{\bar{x}_t^2} - 1; \quad Z_C = (1+Z)\frac{1 - x_t}{(1 - \bar{y}_t)^2} - 1; \qquad (42)$$

$$\vec{B}_2 = \vec{C}_0 = \vec{B}_1(1-t) + \vec{C}_1 t; \qquad (43)$$

Докажем справедливость этих соотношений. Из общего уравнения конической дуги получаем два уравнения для исходной и расщепленной дуг.

$$\vec{A}(t) = \vec{A}_0 + \vec{A}_{01}x + \vec{A}_{12}y; \qquad (44)$$

- исходная дуга.

$$\vec{B}(\tau) = \vec{B}_0 + \vec{B}_{01}\hat{x} + \vec{B}_{12}\hat{y}; \qquad (45)$$

- вторичная дуга.

Подставляя x, y через (33), (34) в уравнение исходной дуги, получаем

$$\vec{A}(t) = \vec{A}_0 + \vec{A}_{01}\left[\bar{x}_t\hat{x} + (x_t - \bar{x}_t)\hat{y}\right] + \vec{A}_{12}y_t\hat{y}; \qquad (46)$$

Поскольку исходная и вторичная дуга на начальном сегменте расщепления описывают одну и ту же линию, то из условия $\vec{A}(t) = \vec{B}(t)$ получаем связь между параметрами дуг.

$$\vec{B}(\tau) = \vec{B}_0 + \vec{A}_{01}\bar{x}_t\hat{x} + \left[\vec{A}_{01}(x_t - \bar{x}_t) + \vec{A}_{12}y_t\right]\hat{y}; \qquad (47)$$

$$\vec{B}_{01} = \vec{A}_{01}\bar{x}_t; \qquad (48)$$

$$\vec{B}_{12} = \vec{A}_{01}(x_t - \bar{x}_t) + \vec{A}_{12}y_t; \qquad (49)$$

Из последних трех выражений получаем соотношения.

$$\vec{B}_0 = \vec{A}_0; \quad (50)$$

$$\vec{B}_1 = \vec{A}_0(1-\bar{x}_t) + \vec{A}_1\bar{x}_t; \quad (51)$$

$$\vec{C}_1 = \vec{A}_1(1-\bar{y}_t) + \vec{A}_2\bar{y}_t; \quad (52)$$

$$\vec{B}_2 = \vec{C}_0 = \vec{B}_1(1-t) + \vec{C}_1 t; \quad (53)$$

$$Z_B = (1+Z)\frac{y_t}{\bar{x}_t^{\,2}} - 1; \quad (54)$$

8. Коническая дуга - как рациональная кривая Безье.

Из математики известно, что рациональная кривая Безье второго порядка задает коническую дугу. Найдем связь между управляющими параметрами формы Безье $\alpha_0, \alpha_1, \alpha_2$ и параметрами рассмотренной выше формы задания коники Z, как интерполяционной вычислительной структуры.

Рациональная кривая Безье второго порядка задается формулой

$$\vec{A}(t) = \frac{\vec{A}_0\alpha_0(1-t)^2 + 2\vec{A}_1\alpha_1(1-t)t + \vec{A}_2\alpha_2 t^2}{\alpha_0(1-t)^2 + 2\alpha_1(1-t)t + \alpha_2 t^2}, \quad (55)$$

где: $\vec{A}_0, \vec{A}_1, \vec{A}_2$ - вершины треугольника. $\alpha_0, \alpha_1, \alpha_2$ - скалярные параметры кривой Безье, они должны быть неотрицательны и не равны нулю все одновременно. Представим $\vec{A}(t)$ в форме

$$\vec{A}(t) = \vec{A}_0 + \vec{A}_{01}x(t) + \vec{A}_{12}y(t), \quad (56)$$

где:

$$x(t) = \frac{2\alpha_1(1-t)t + \alpha_2 t^2}{\alpha_0(1-t)^2 + 2\alpha_1(1-t)t + \alpha_2 t^2}, \quad (57)$$

$$y(t) = \frac{\alpha_2 t^2}{\alpha_0(1-t)^2 + 2\alpha_1(1-t)t + \alpha_2 t^2}. \quad (58)$$

Из проведенного выше анализа следует, что для любой точки коники должен иметь место инвариант

$$2(1+Z) = \frac{(x-y)^2}{y(1-x)}, \quad (59)$$

подставляя $x(t), y(t)$ в это выражение, получаем выражение эксцентриситета Z через параметры $\alpha_0, \alpha_1, \alpha_2$ рациональной кривой Безье второго порядка

$$Z = \frac{2\alpha_1^2}{\alpha_0\alpha_2} - 1, \quad (60)$$

что и требовалось найти.

Представляет интерес и обратная задача, имея эксцентриситет Z, найти набор коэффициентов $\alpha_0, \alpha_1, \alpha_2$ рациональной кривой Безье второго порядка. Отметим, что набор коэффициентов можно умножить на любое ненулевое число, следовательно, коэффициенты можно нормировать таким образом, чтобы выполнялось условие $\alpha_0 + 2\alpha_1 + \alpha_2 = 1$. Из этого ограничения следует оценка $\alpha_1 \le 1/2$. Задаваясь $\alpha_1 \in [0, 1/2]$ и решая систему уравнений

$$\begin{cases} Z = \dfrac{2\alpha_1^2}{\alpha_0\alpha_2} - 1 \\ \alpha_0 + 2\alpha_1 + \alpha_2 = 1 \end{cases} \quad (61)$$

Находим выражения α_0, α_2 через Z и α_1. Это решение является общим, но можно найти важное частное решение. Пусть $\alpha_1 = \alpha/2$, где α задает положение точки пересечения медианы треугольника с конической дугой. Как было показано выше, Z и α связаны следующим соотношением

$$Z = \frac{\alpha^2 + 2\alpha - 1}{(1-\alpha)^2}. \quad (62)$$

В этом случае α_0, α_2 имеют следующие простые выражения $\alpha_0 = \alpha_2 = \frac{1}{2}(1-\alpha)$.

Коническая дуга будет задаваться следующим уравнением

$$\vec{A}(t) = \frac{\vec{A}_0(1-\alpha)(1-t)^2 + 2\vec{A}_1\alpha(1-t)t + \vec{A}_2(1-\alpha)t^2}{(1-\alpha)(1-t)^2 + 2\alpha(1-t)t + (1-\alpha)t^2}. \quad (63)$$

9. Представление конического сплайна набором конических дуг.

Конический сплайн является частным случаем рационального B-сплайна степени 2 и его сегмент может быть представлен следующим образом

$$\vec{C}_i(t) = \frac{\sum_{k=-1}^{1} h_{i+k}B_k(t)\vec{V}_{i+k}}{\sum_{k=-1}^{1} h_{i+k}B_k(t)}, \qquad t \subseteq [0,1]. \quad (64)$$

Где $\vec{C}_i(t)$ - i-тый сегмент сплайна.

$<\vec{V}_i>$ - управляющие вершины сплайна.

$<h_i>$ - узловые коэффициенты сплайна.

$B_k(t)$ - базовый полином B-сплайна.

$$B_{-1}(t)=\frac{1}{2}(1+t)^2; \qquad (65a)$$

$$B_0(t)=\frac{1}{2}\left(-2t^2+2t+1\right); \qquad (65b)$$

$$B_1(t)=\frac{1}{2}t^2. \qquad (65c)$$

Представим сегмент сплайна в форме рациональной кривой Безье второго порядка

$$\vec{C}(t)=\frac{w_0\vec{K}_0(1-t)^2+2w_1\vec{K}_1(1-t)t+w_2\vec{K}_2}{w_0(1-t)^2+2w_1(1-t)t+w_2t^2}, \qquad (66)$$

.

Приравнивая коэффициенты при одинаковых степенях t обоих форм описания сегмента, находим:

$$\vec{K}_0=\frac{\vec{V}_{i-1}h_{i-1}+\vec{V}_ih_i}{h_{i-1}+h_i}; \qquad (67a)$$

$$\vec{K}_1=\vec{V}_i; \qquad (67b)$$

$$\vec{K}_2=\frac{\vec{V}_{i+1}h_{i+1}+\vec{V}_ih_i}{h_i+h_{i+1}}; \qquad (67c)$$

$$w_0=\frac{h_i+h_{i-1}}{h_{i-1}+6h_i+h_{i+1}}; \qquad (68a)$$

$$w_1=\frac{2h_i}{h_{i-1}+4h_i+h_{i+1}}; \qquad (68b)$$

$$w_2=\frac{h_i+h_{i+1}}{h_{i-1}+4h_i+h_{i+1}}; \qquad (68c)$$

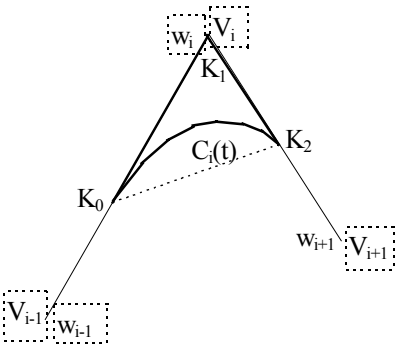

Рис. 10. Сегмент конического сплайна.

Ранее было доказано, что лекальная дуга может быть задана уравнением

$$\vec{C}(t)=\vec{K}_0+\vec{K}_{01}x(t)+\vec{K}_{12}y(t). \qquad (69)$$

Отметим, что параметр t в последнем уравнении не эквивалентен этому параметру в предшествующих формулах, однако, алгебраическая форма обоих уравнений является одной и той же
Сегмент сплайна:

$$x^2+y^2+2xy\frac{2w_1^2-w_0w_2}{w_0w_2}+\frac{4w_1^2}{w_0w_2}y=0. \qquad (70)$$

Коническая дуга:

$$x^2+y^2+2Zxy-2y(1+Z)=0. \qquad (71)$$

Приравнивая коэффициенты, при одинаковых степенях x и y, находим

$$Z=\frac{2w_1^2-w_0w_2}{w_0w_2}.$$

Теперь мы можем применить генератор конических дуг для растрирования сегмента сплайна, а следовательно, и самого сплайна.

Заключение.

В данной работе предложен целочисленный алгоритм растрирования конической дуги, вписанной в треугольник общего положения. Данный алгоритм позволяет выполнять не только растрирование дуги, но и ее закраску и текстурирование. Предложена вычислительная структура, реализующая данный алгоритм, и оценены аппаратные затраты на ее реализацию. Рассмотрен ряд задач, имеющих непосредственное отношение к использованию конических дуг в компьютерной графике и геометрическом моделировании. Показано, что коническая дуга может быть использована в качестве геометрического примитива вывода в современных видео картах.

Литература.

[1]. **Bresenham J.E.** *Algorithm for Computer Control of Digital Plotter.* // IBM System Journal. - 1965. - V. 4(1).- P.25-30.

[2]. **Pitteway M.** *Algorithm for drawing ellipses or hyperbolae with a digital plotter.* // Computer Journal. - 1966. - V. 10(3). - P. 282-289.

[3]. **Botting R.J.,Plitteway M.L.V.** *Algorithm for drawing ellipses or hyperbolae with a digital plotter.* // Computer Journal. - 1968. - V. 11 (1). - P. 120.

[4]. **Bresenham J.E.** *A Linear Algorithm for Incremental Digital Display of Circular Arcs.* // Communication of the ACM. - 1977. - V. 20(2). - P. 100-106.

[5]. **P.E.Danlelson.** *Incremental curve generation. // IEEE Transaction on Computers.* - 1970. - V.C-19. - P. 783-793.

[6]. **B. W. Jordan, W.J.Lennon, W.J.Holm.** *An Improved Algorithm for Generation of Nonparametric Curves*. // IEEE Transaction on Computers. - 1973. - V.C- 22(12). - P. 1052-1060.

[7]. **Yasuhlto Suenaga, Takahiko Kamae, Tomonori**

Kabayashi. *A High Speed Algorithm for the Circular Arcs.* // IEEE Trans. on Computers. - 1979. -V. C-28(10). - P. 728-736.

[8]. M.L.V.Plitteway. R.J.Bottlng. *Integer circles etc. - Three move of Bresenham's algorithm*. // Computer Graphics and Image Processing. - 1974. - V.3. - P. 260-261.

[9]. B.K.P.Horn. *Circle generators tor display devices.* //Computer Graphics, and Image Processing. -1976.- V.5.-P. 580-588.

[10]. N.I.Badler. *Disk generators for a raster display device*. // Computer, Graphics and Image Processing. - 1977. - V. 6. - P. 589-593.

[11]. W.L.Chung. *On circle generation algorithms.* // Computer Graphics and Image Processing. - 1977. - V.6. - P. 196-198.

[12]. M.Doros. *Algorithms tor Generation to Discrete Circles, Rings and Disks.* // Computer Graphics and Image Processing. - 1979. - V.10(4). - P. 366-371.

[13]. M.Plitteway, D.Watkinson. Breseriham's *Algorithm with Gray Scale.* // Communication of the ACM. - 1980. - V. 23(11).- P. 625-626.

[14]. G.Moller. *Past digital vector and circle generator with binary rate multipliers.* // Computer Graphics. - 1978.- V.12(4).- P. 81-91.

[15]. P.G.McCrea, P.W.Baker. *On DDA circle generation for computer graphics.* // IEEE Trans. on Computers. - 1975. - V. C-24. - P. 1109-1110.

[16]. F.Rubin. *Generation of Nonparametric Curves.* // IEEE Transaction on Computers. - 1976. - V.C-25(1). - P.103.

[17]. Ramot. *Nonparametric Curves.* // IEEE Transaction on Computers. - 1976. - V.C-250). - P.103-104.

[18]. P.C.Maxwell. P.W.Baker. *The Generation of Polygons Representing Circles. Ellipses and Hyperbolas.* // Computer Graphics end Image Processing. - 1979. - V.10. - P.84-93.

[19]. Н.С.Анишин, А. М.Тивков. *Оптимальный алгоритм цифровой линейной интерполяции*. // Изв. Вузов СССР. - Приборостроение. - 1983. - N8. - С.56-59.

[20]. С.В.Чириков. *Целочисленный алгоритм поточечной генерации эллипсов, парабол и гипербол - основа генератора графических примитивов для устройств отображения растрового типа*. // Вопросы радиоэлектроники. Серия ОВР. - 1990. - Вып.21.С.89-97.

[21]. С.В.Чириков. *Целочисленный алгоритм поточечной генерации геометрически заданных конических дуг - основа генератора графических примитивов для устройств отображения растрового типа.* //Вопросы радиоэлектроники. Серия ОВР. - 1991. - Вып. 7.