Приднетровский государственный университет им. Т.Г. Шевченко

Инженерно-технический институт

Кафедра “Программное обеспечение вычислительной техники
и автоматизированных систем

Башкатов А.М.

Компьютерная графика

Для студентов направлений: “Программная инженерия”,
“Информатика и вычислительная техника”,
“Информационные системы и сети”

Лекция 4.
Формирование растра. Понятие связности пикселей. Растровое представление отрезка. Алгоритмы растризации. Построение кривых различных порядков. Закраска области цветом. Заполнение многоугольников

Тирасполь, 2019

РАСТР И СВЯЗНОСТЬ

Растр — (нем. Raster) - решетка для структурного преобразования направленного светового пучка, т.е. изображение, построенное из отдельных элементов (точек), как правило, расположенных регулярно. В большинстве приложений компьютерной графики, растровое изображение представляется двумерным массивом точек, цвет и яркость каждой из которых задаются независимо

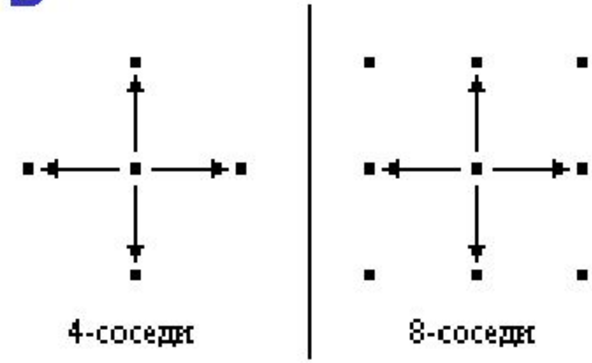


Рис.1

Связность (рис.1) – возможность соединения двух пикселей растровой линией, т. е. последовательным набором пикселей. Различают 4-х и 8-ми связность.

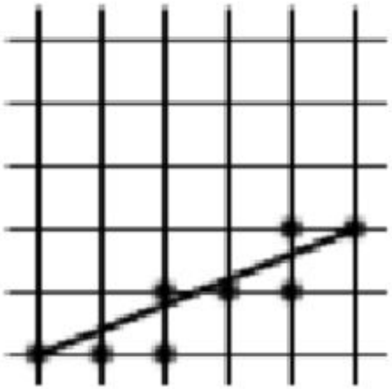


Рис.2

Четырехсвязность: пиксели считаются соседними, если либо их *x*-координаты, либо их *y* – координаты отличаются на единицу, т.е.: $|x1- - x2| + |y1 - y2| \leq 1$.

Четырехсвязная (рис.2) и восьмисвязная (рис.3) линии

Восьмисвязность: пиксели считаются соседними, если их *x*-координаты и *y*-координаты отличаются не более чем на единицу, или: $|x1- - x2| \leq 1$, $|y1 - y2| \leq 1$.

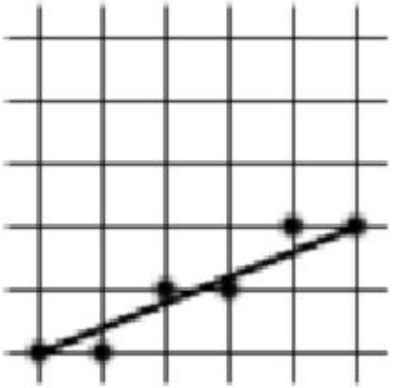
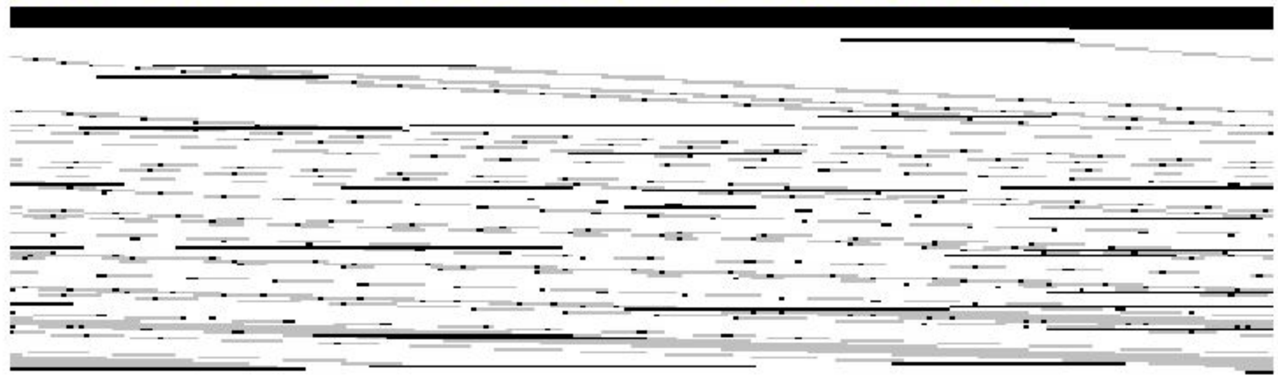


Рис.3

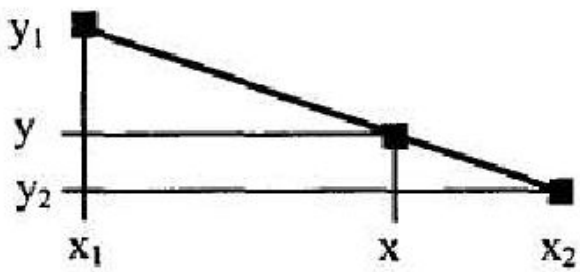
Классификация растровых алгоритмов компьютерной графики

АЛГОРИТМЫ РАСТРОВОЙ ГРАФИКИ

Рис.4 Варианты соединения смежных пикселов растра



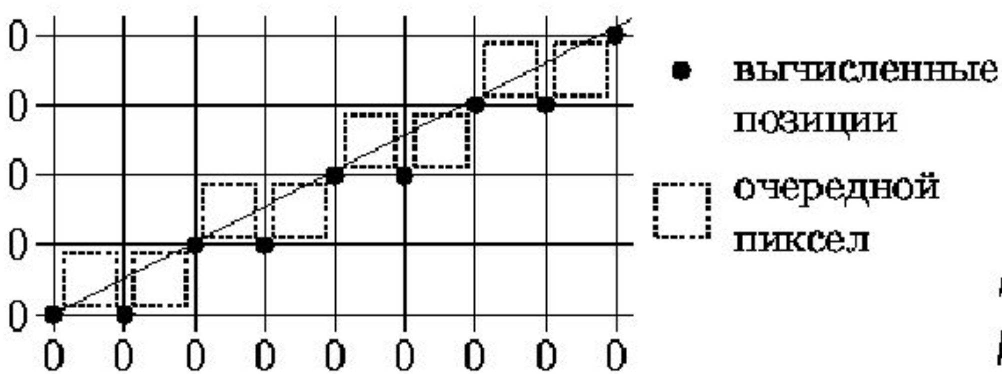
ПРЯМОЕ ВЫЧИСЛЕНИЕ КООРДИНАТ



$$\frac{x - x_1}{y - y_1} = \frac{x_2 - x_1}{y_2 - y_1}$$

Откуда в цикле вычисляют значения *x* и *y*, как *y=F(x)* и *x=f(y)* пока не достигнут граничных значений

МЕТОД ЦИФРОВОГО ДИФФЕРЕНЦИАЛЬНОГО АНАЛИЗАТОРА (DDA - Digital Differential Analyzer)



В основе метода - вычисление:

$$\frac{dy}{dx} = \frac{Py}{Px}$$, где $$Py = y_k - y_n$$ $$Px = x_k - x_n$$

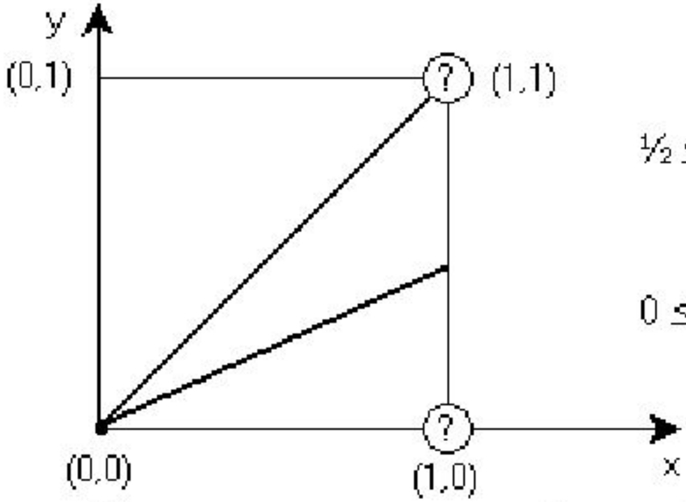
Py - приращение координат отрезка по оси Y

Px - приращение координат отрезка по оси X

Рис.6 Генерация отрезка несимметричным ЦДА

http://alglib.sources.ru

АЛГОРИТМ БРЕЗЕНХЕМА ДЛЯ ГЕНЕРАЦИИ ОТРЕЗКА



$$\tfrac{1}{2} \le \frac{\Delta y}{\Delta x} \le 1 \text{ (ошибка } \ge 0)$$

$$0 \le \frac{\Delta y}{\Delta x} \le \tfrac{1}{2} \text{ (ошибка } < 0)$$

Инициализировать ошибку в $-\tfrac{1}{2}$

$$\text{ошибка} = \text{ошибка} + \frac{\Delta y}{\Delta x}$$

Рис.7 Определение знака ошибки

Если угловой коэффициент прямой < 1/2, то естественно точку, следующую за точкой (0,0), поставить в позицию (1,0) (рис. 8, а), а если угловой коэффициент > 1/2, то - в позицию (1,1) (рис. 8, б).

Если E < 0, то точное Y-значение округляется до последнего меньшего целочисленного значения Y, т.е. Y-координата не меняется по сравнению с предыдущей точкой. В противном случае Y увеличивается на 1.

Для принятия решения куда заносить очередной пиксел вводится величина отклонения E (ошибки) (Рис.7) точной позиции от середины между двумя возможными растровыми точками в направлении наименьшей относительной координаты. Знак E используется как критерий для выбора ближайшей растровой точки.

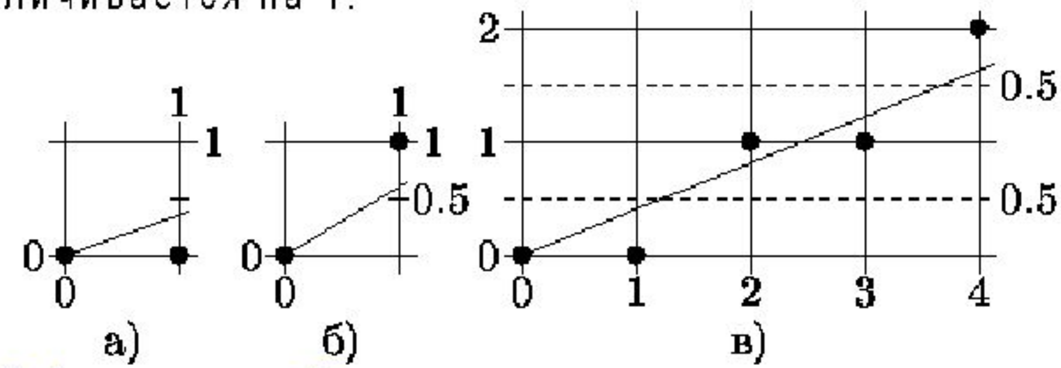


Рис.8 Алгоритм Брезенхема генерации отрезков

http://compgraph.tpu.ru

http://www.bourabai.kz

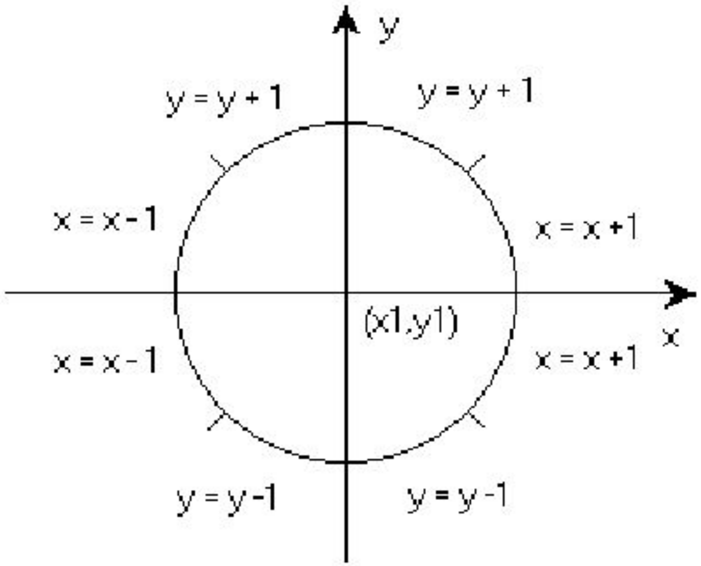


Рис.9 Схема обобщенного варианта Алгоритма Брезенхема

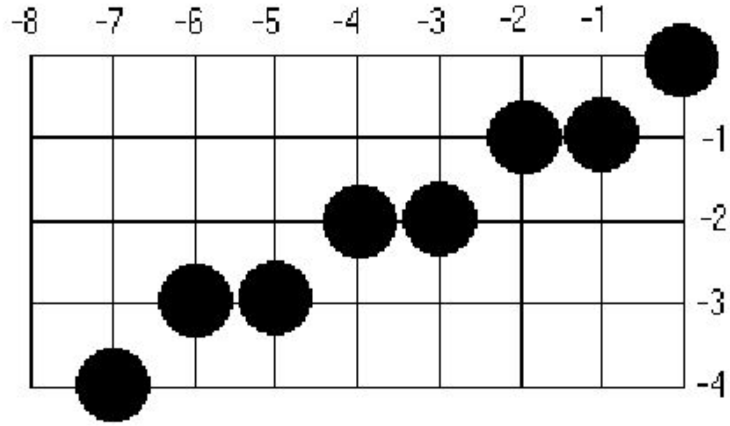


Рис.10 Результат работы обобщенного алгоритма Брезенхема для III октанта

АЛГОРИТМ БРЕЗЕНХЕМА ДЛЯ ГЕНЕРАЦИИ ОКРУЖНОСТИ

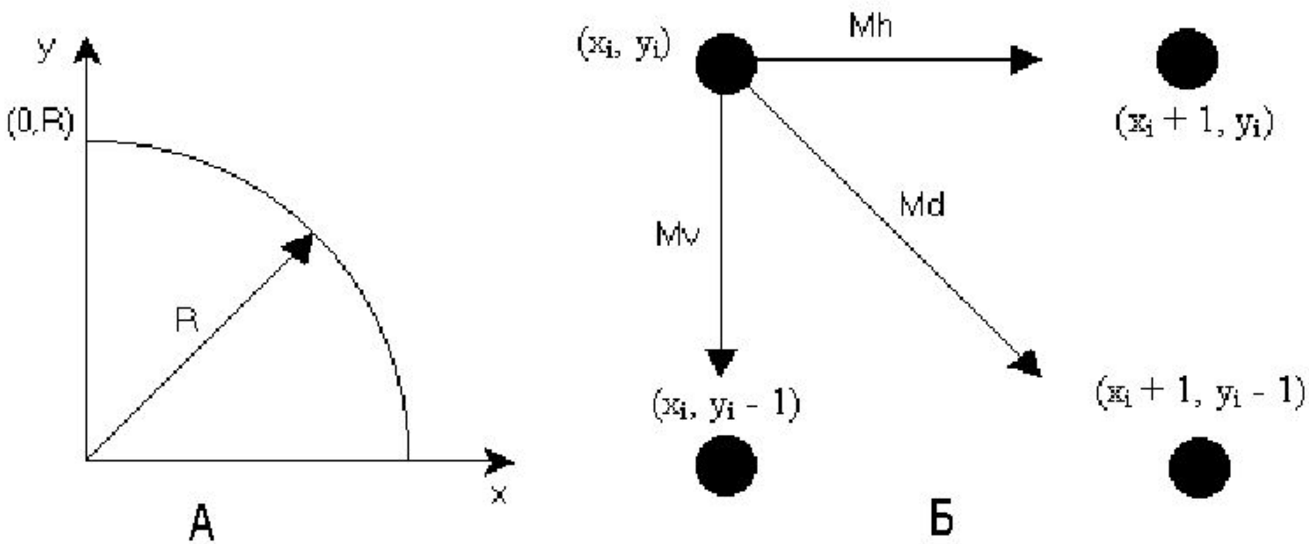


Рис.11 Алгоритм Брезенхема для генерации окружности
А – окружность в I квадранте;
Б – выбор пикселов в I квадранте

АЛГОРИТМ БРЕЗЕНХЕМА ДЛЯ ГЕНЕРАЦИИ ОКРУЖНОСТИ

http://umup.ru

$$Mh = |(x_i + 1)^2 + (y_i)^2 - R^2|;$$
$$Md = |(x_i + 1)^2 + (y_i - 1)^2 - R^2|;$$
$$Mv = |(x_i)^2 + (y_i - 1)^2 - R^2|;$$
$$\Delta i = (x_i + 1)^2 + (y_i - 1)^2 - R^2;$$
$$\delta = 2(\Delta_i + y_i) - 1;$$
$$\delta' = 2(\Delta_i - x_i) - 1.$$

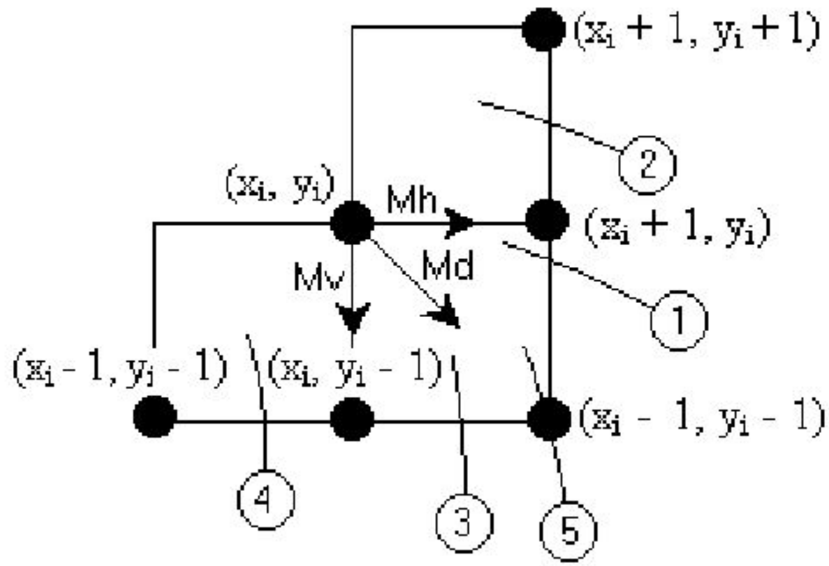


Рис. 12 Пересечение окружности и сетки растра

$\Delta i < 0$

$\delta <= 0$ выбираем пиксел (xi +1, yi) ==> Mh

$\delta > 0$ выбираем пиксел (xi +1, yi -1) ==> Md

$\Delta i > 0$

$\delta' <= 0$ выбираем пиксел (xi +1, yi -1) ==> Md

$\delta' > 0$ выбираем пиксел (xi, yi - 1) ==> Mv

$\Delta i = 0$

выбираем пиксел (xi + 1, yi - 1) ==> Md

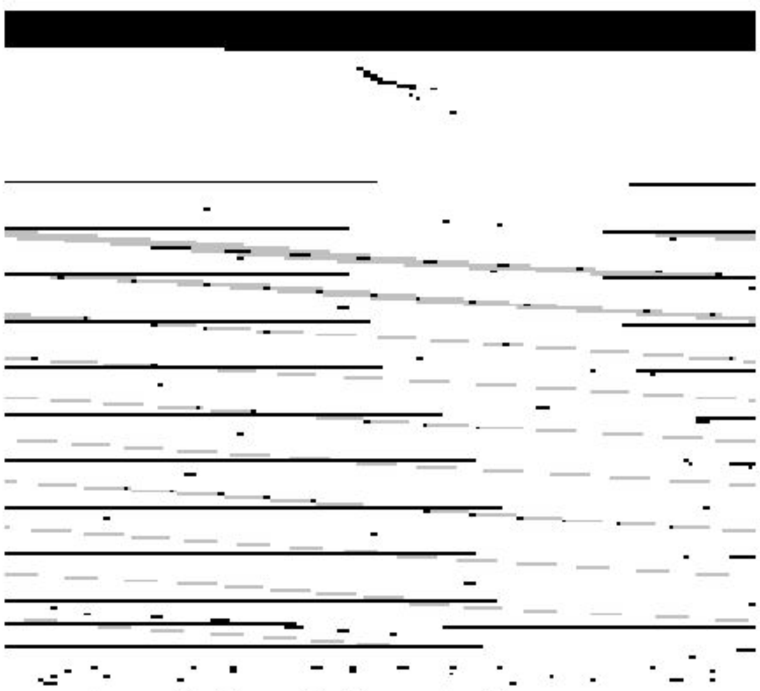


Рис.13 Результат работы алгоритма брезенхема для генерации дуги окружности (I квадрант)

АЛГОРИТМЫ РАСТРОВОЙ ГРАФИКИ

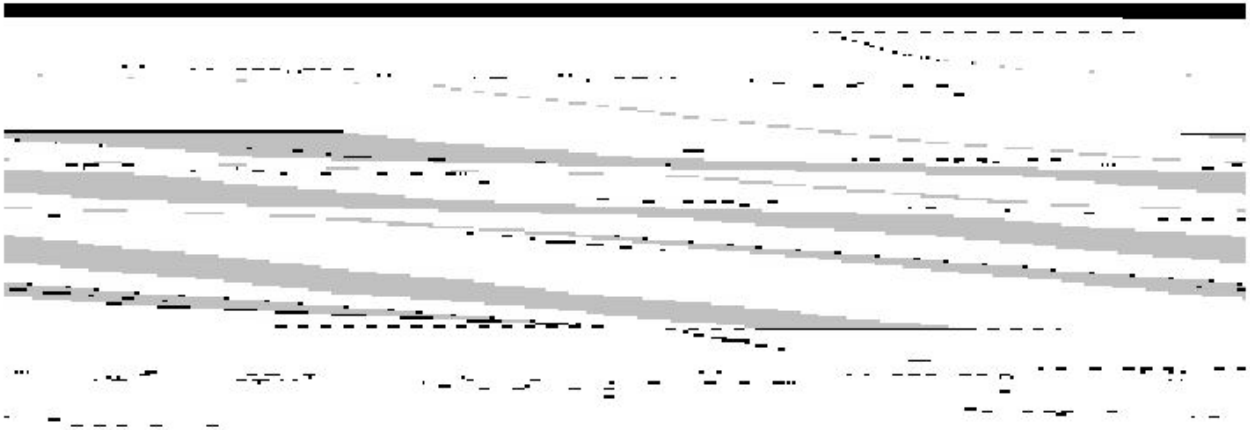


Рис.14 Растровая развертка сплошных областей (использование прямоугольной оболочки)

<u>ВОЗМОЖНЫЙ РЕЗУЛЬТАТ</u>:
Нет пересечений с многоугольником ==> точка внешняя (точка 1).
Нечетное число пересечений границы ==> точка внутренняя (точки 3 и 4).
Четное число пересечений границы ==> точка внешняя (точка 5).
Одно пересечение ==> вершина (точка Pn)

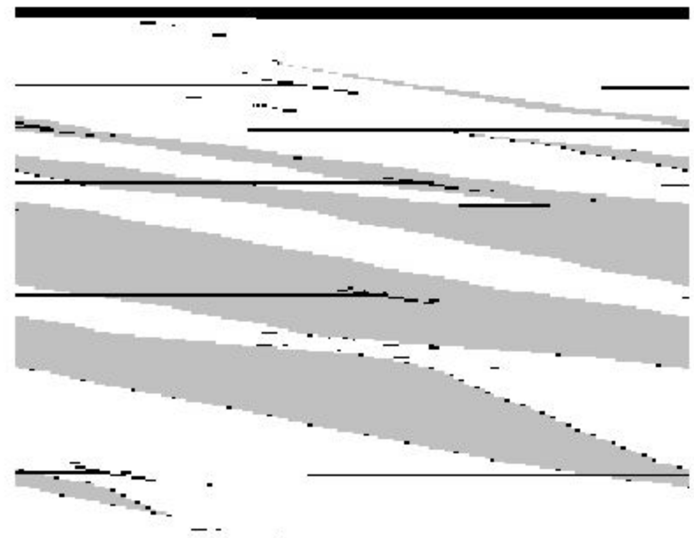
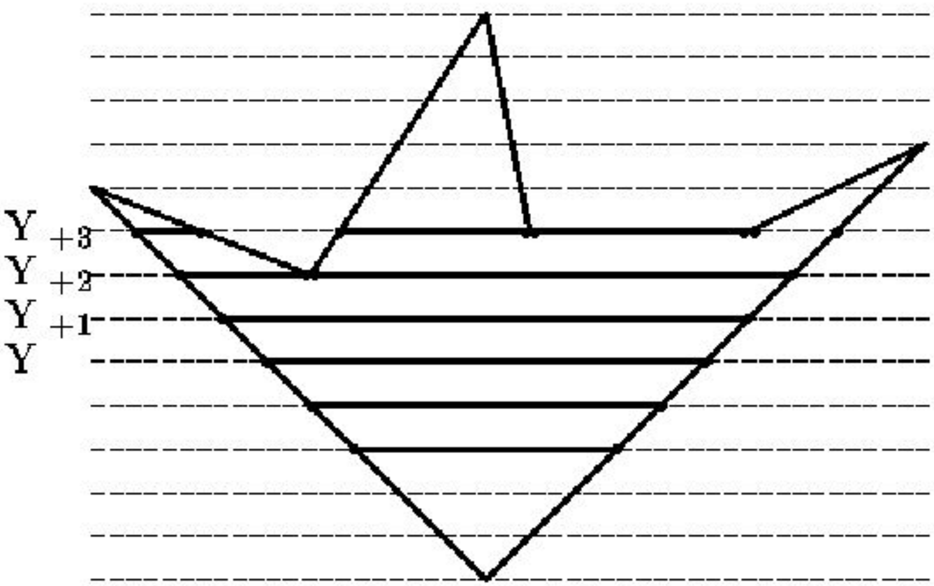


Рис.15 Заполнение многоугольника в порядке сканирования строк

Такие алгоритмы функционируют в пространстве изображений, причем образ в них генерируется построчно

Роджерс Д. Алгоритмические основы машинной графики

Алгоритм построчного сканирования



Т.к. соседние пикселы в строке скорее всего одинаковы и меняются только там где строка (рис.16) пересекается с ребром многоугольника. Это называется когерентностью растровых строк (строки сканирования Yi, Yi+1, Yi+2 на рис.). При этом достаточно определить X-координаты пересечений строк сканирования с ребрами. Пары отсортированных точек пересечения задают интервалы заливки.

Рис.16 Построчная закраска многоугольника

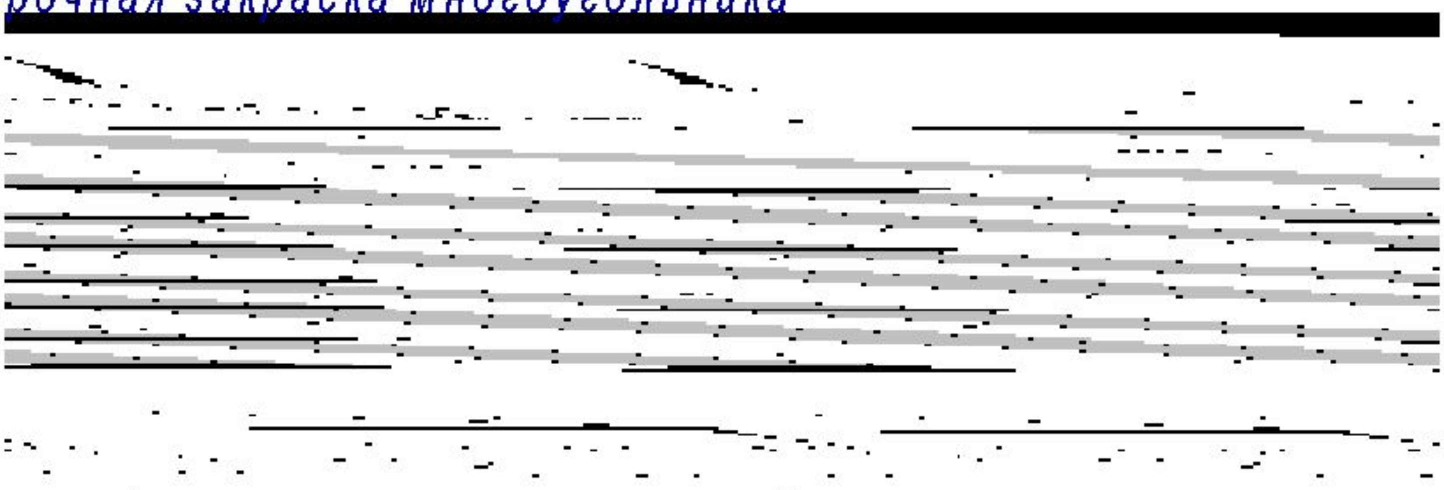
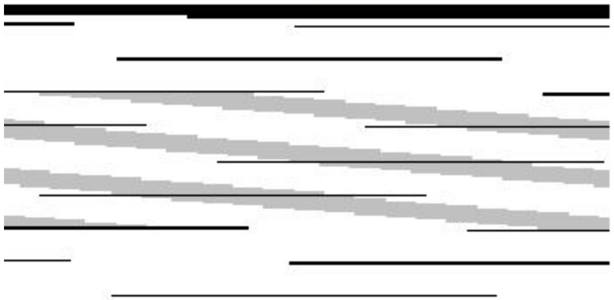


Рис.17 Изменение системы координат строк сканирования
а – при использовании адресов пикселов
б - при прохождении сканирующей строки через центр пикселов

Алгоритмы заполнения растровых областей



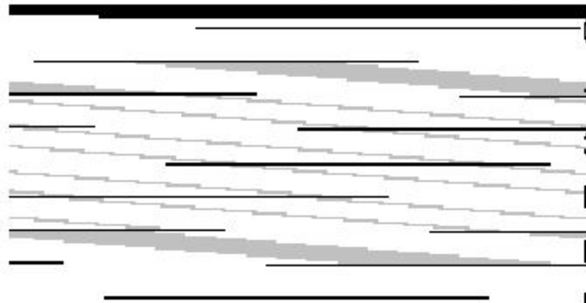


Рис.18 Типы областей:

Внутренне определенная область

Гранично определенная область

Основаны на анализе зон:

- <u>гранично-определенных</u>, т.е. задаваемые своей (замкнутой) границей так, что коды пикселов границы отличны от кодов внутренней, перекрашиваемой области;
- <u>внутренне-определенных</u>, нарисованных одним заданным кодом пиксела. При заливке этот код заменяется на новый код закраски

На коды пикселы внутренней части области налагаются два условия - они должны быть отличны от кода пикселов границы и кода пиксела перекраски. Если внутри гранично-определенной области имеется еще одна граница, нарисованная пикселами с тем же кодом, что и внешняя граница, то соответствующая часть области не должна перекрашиваться

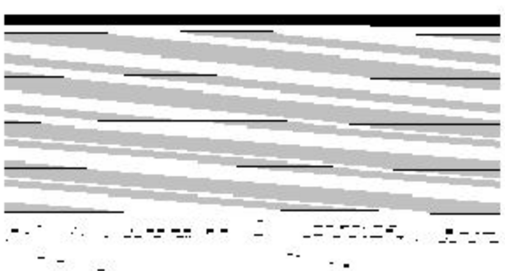
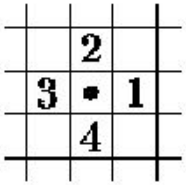
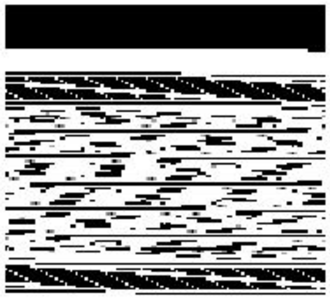


Рис.19 Варианты связности зон заполнения

АЛГОРИТМ С ЗАЛИВКОЙ (ДЛЯ 4-Х СВЯЗНОЙ ОБЛАСТИ)



А



Б

Рис.20 Порядок:
А - перебора соседних пикселов;
Б - заливки области

1. Поместить координаты затравки в стек
2. Пока стек не пуст извлечь координаты пиксела из стека.
3. Перекрасить пиксел.
4. Для всех четырех соседних пикселов проверить является ли он граничным или уже перекрашен.
5. Если нет, то занести его координаты в стек.

АЛГОРИТМ ЗАПОЛНЕНИЯ С ЗАТРАВКОЙ

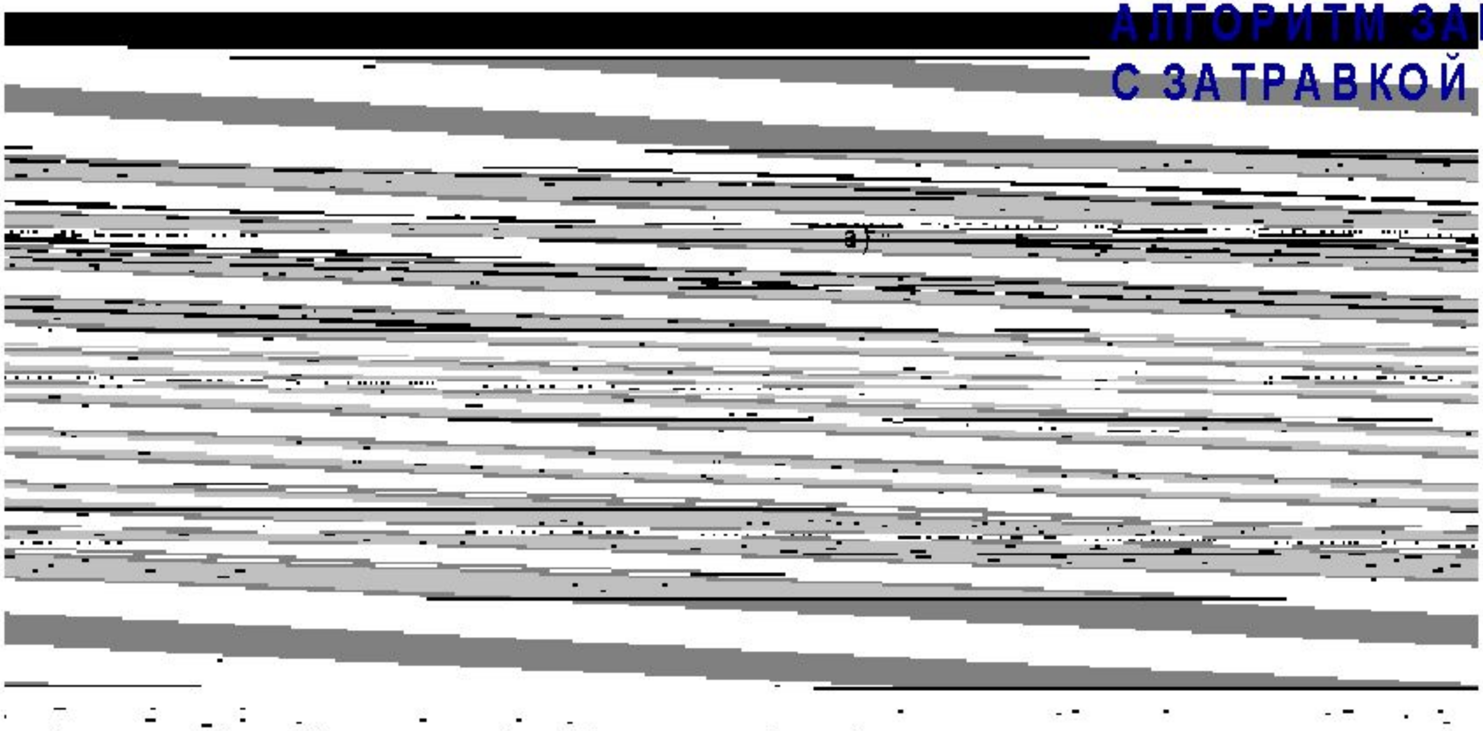


Рис.21 Простой стековый алгоритм заполнения с затравкой

Использует пространственную когерентность:

- пикселы в строке меняются только на границах;
- при перемещении к следующей строке размер заливаемой строки скорее всего или неизменен или меняется только на 1 пиксел.

http://edu.nstu.ru/education/educourses/Complm/Lit/Graphic/kg/3.htm

АЛГОРИТМ ЗАТРАВОЧНОГО ЗАПОЛНЕНИЯ МНОГОУГОЛЬНИКА

Рис.22 ПОРЯДОК РАБОТЫ АЛГОРИТМА
а – помещение в стек затравочного пиксела;
b – заполнение на 3-ей сканирующей строке;
c - // - // - на 4-ой;
d - // - // - на 5-ой;
e – извлечение пиксела 2 и завершение (заполнение стека). Цикл.

http://edu.nstu.ru/education/educourses/CompIm/Lit/Graphic/kg/3.htm

КРИВЫЕ БЕЗЬЕ

Кривая Безье — параметрическая кривая, задаваемая выражением

$$\mathbf{B}(t) = \sum_{i=0}^{n} \mathbf{P}_i \mathbf{b}_{i,n}(t), \quad 0 < t < 1$$

где P_i — функция компонент векторов опорных вершин, а $\mathbf{b}_{i,n}(t)$ — базисные функции кривой Безье, называемые также полиномами Бернштейна. $\mathbf{b}_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}$

где $\binom{n}{i} = \frac{n!}{i!(n-i)!}$ — число сочетаний из n по i, где n — степень полинома, i — порядковый номер опорной вершины.

Историческая справка

Кривые Безье или Кривые Бернштейна-Безье были разработаны в 60-х годах XX века независимо друг от друга Пьером Безье из автомобилестроительной компании «Рено» и Полем де Кастельжо из компании «Ситроен», где применялись для проектирования кузовов автомобилей.

Несмотря на то, что открытие де Кастельжо было сделано несколько ранее Безье, его исследования не публиковались и скрывались компанией как производственная тайна до конца 1960-х.

Кривая Безье является частным случаем многочленов Бернштейна, описанных Сергеем Натановичем Бернштейном в 1912 году.

Впервые кривые были представлены широкой публике в 1962 году французским инженером Пьером Безье, который, разработав их независимо от де Кастельжо, использовал их для компьютерного проектирования автомобильных кузовов. Кривые были названы именем Безье, а именем де Кастельжо назван разработанный им рекурсивный способ определения кривых.

Впоследствии это открытие стало одним из важнейших инструментов систем

Виды кривых Безье

Линейные кривые

При n = 1 кривая представляет собой отрезок прямой линии, опорные точки P_0 и P_1 определяют его начало и конец. Кривая задаётся уравнением:

$$\mathbf{B}(t) = (1-t)\mathbf{P}_0 + t\mathbf{P}_1 \quad t \in [0,1]$$

Квадратичные кривые

Квадратичная кривая Безье задаётся 3-мя опорными точками: P_0, P_1 и P_2.

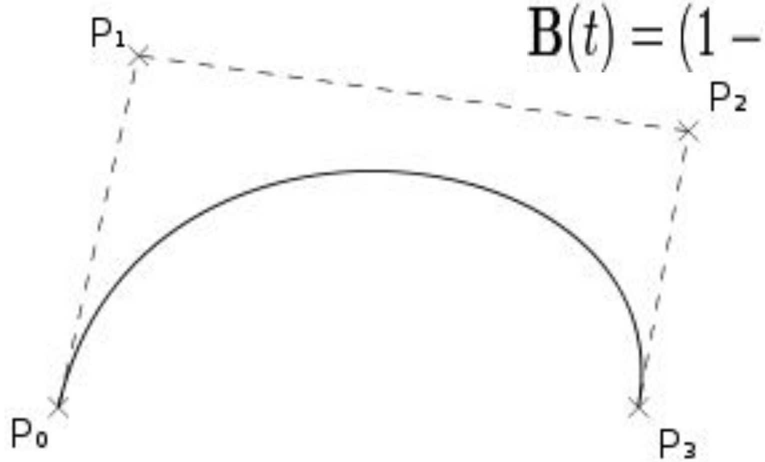
$$\mathbf{B}(t) = (1-t)^2\mathbf{P}_0 + 2t(1-t)\mathbf{P}_1 + t^2\mathbf{P}_2, \quad t \in [0,1]$$

Квадратичные кривые Безье в составе сплайнов используются для описания формы символов в шрифтах TrueType и в SWF файлах.

Кубические кривые

В параметрической форме кубическая кривая Безье описывается следующим уравнением:

$$\mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3t(1-t)^2\mathbf{P}_1 + 3t^2(1-t)\mathbf{P}_2 + t^3\mathbf{P}_3, \quad t \in [0,1]$$



Кубическая кривая Безье

Четыре опорные точки P0, P1, P2 и P3, заданные в 2-х или 3-мерном пространстве определяют форму кривой. Линия берёт начало из точки P0 направляясь к P1 и заканчивается в точке P3 подходя к ней со стороны P2.

Т.е. кривая не проходит через точки P1 и P2, они используются для указания её направления. Длина отрезка между P0 и P1 определяет, как скоро кривая повернёт к P3.

В матричной форме кубическая кривая Безье записывается следующим образом:

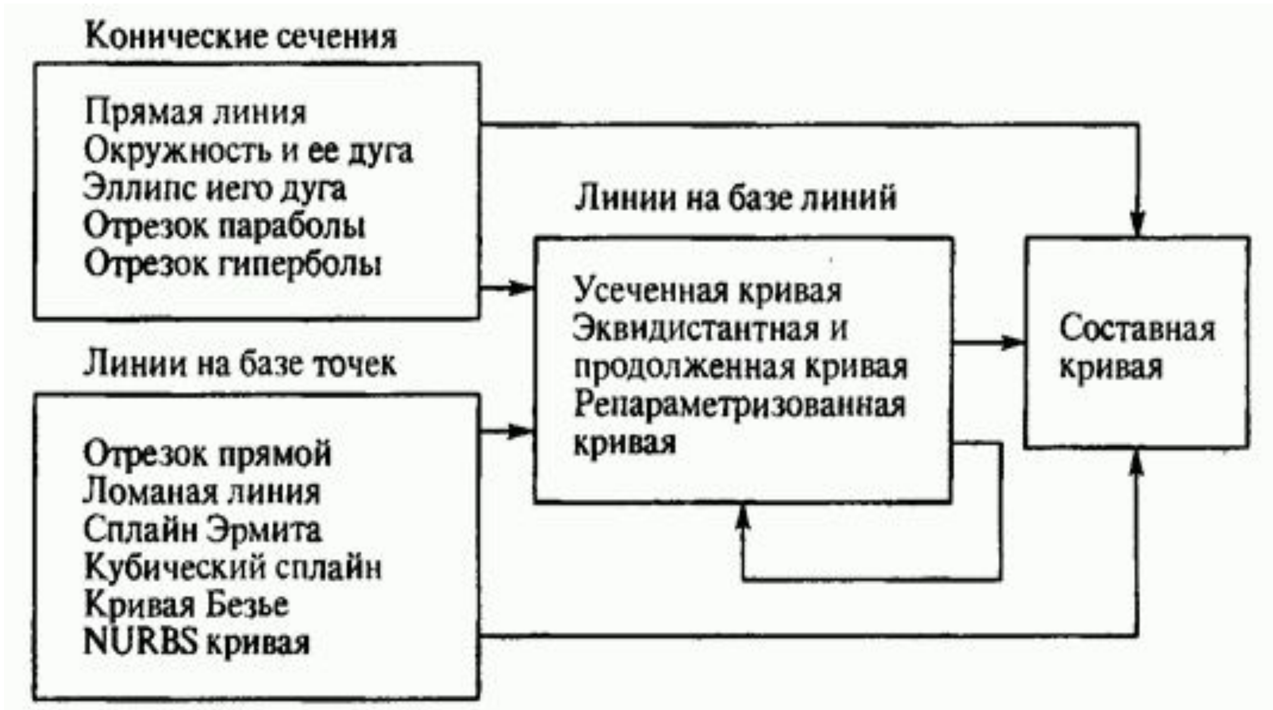
$$\mathbf{B}(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \mathbf{M}_B \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix}$$

где M_b называется базисной матрицей Безье:

$$\mathbf{M}_B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

В современных графических системах, таких как PostScript, Metafont и GIMP для пред-ставления криволинейных форм используются сплайны Безье, составленные из кубических кривых.

Способы построения кривых линий



http://www.chinapads.ru

ПОСТРОЕНИЕ КРИВЫХ БЕЗЬЕ

Линейные кривые

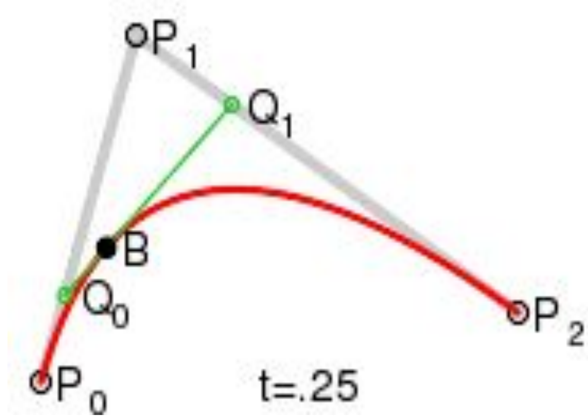
Параметр t в функции, описывающей линейный случай кривой Безье, определяет где именно на расстоянии от P0 до P1 находится B. Например, при t = 0,25 значение функции B соответствует четверти расстояния между точками P0 и P1. Параметр t изменяется от 0 до 1, а B описывает отрезок прямой между точками P0 и P1.

Квадратичные кривые

Для построения квадратичных кривых Безье требуется выделение двух промежуточных точек Q0 и Q1 из условия чтобы параметр t изменялся от 0 до 1:

Алгоритм построения следующий

1. Точка Q0 изменяется от P0 до P1 и описывает линейную кривую Безье.
2. Точка Q1 изменяется от P1 до P2 и также описывает линейную кривую Безье.
3. Точка B изменяется от Q0 до Q1 и описывает квадратичную кривую Безье.

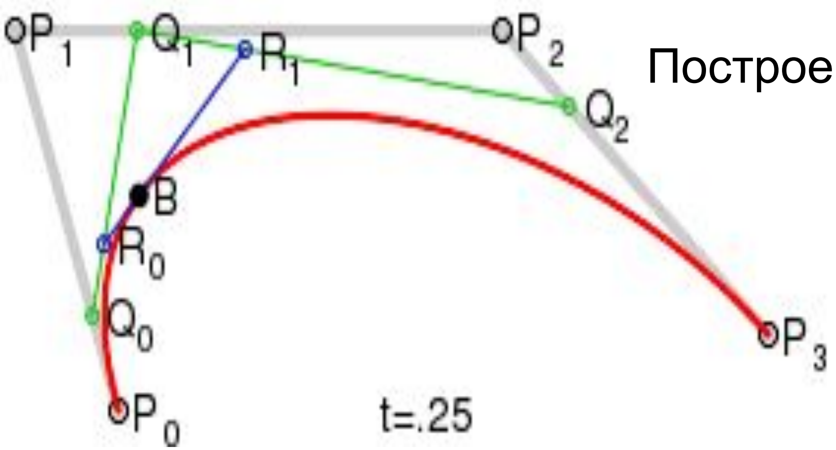


Построение квадратичной кривой Безье

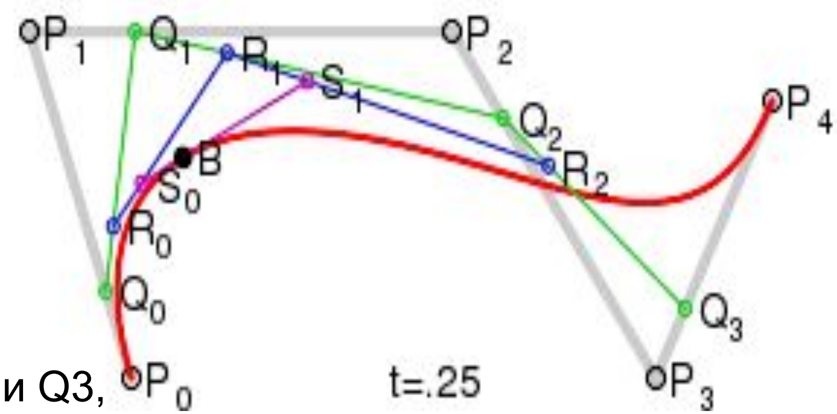
Кривые высших степеней

Для построения кривых высших порядков соответственно требуется и больше промежуточ-ных точек.

Для кубической кривой это промежуточные точки Q0, Q1 и Q2, описывающие линейные кривые, а также точки R0 и R1, которые описывают квадратичные кривые: более простое уравнение p0q0/p0q1=q1p1/p1p2=bq0/q1q0



Построение кубической кривой Безье



Для кривых четвёртой степени это будут точки Q0, Q1, Q2 и Q3, описывающие линейные кривые, R0, R1 и R2, которые описывают квадратичные кривые, а также точки S0 и S1, описывающие кубические кривые Безье:

Построение кривой Безье 4-й степени

Свойства кривой Безье

- непрерывность заполнения сегмента между начальной и конечной точками;
- кривая всегда располагается внутри фигуры, образованной линиями, соединяющими контрольные точки;
- при наличии только двух контрольных точек сегмент представляет собой прямую линию;
- прямая линия образуется при коллинеарном размещении управляющих точек;
- кривая Безье симметрична, то есть обмен местами между начальной и конечной точками не влияет на форму кривой;
- масштабирование и изменение пропорций кривой Безье не нарушает ее стабильности, так как она с - математической точки зрения «аффинно инвариантна»;
- изменение координат хотя бы одной из точек ведет к изменению формы всей кривой Безье;
- степень кривой всегда на одну ступень ниже числа контрольных точек. Например, при трех контрольных точках форма кривой — парабола;
-окружность не может быть описана параметрическим уравнением кривой Безье;