

Устранение ступенчатости

В основном существует два метода устранения искажений изображения такого рода.

Первый связан с увеличением частоты выборки, что достигается с помощью увеличения разрешения растра. Таким образом, учитываются более мелкие детали.

Однако существует определенное ограничение на способность растровых графических устройств с ЭЛТ выводить очень мелкие растры. Такое ограничение предполагает, что растр надо вычислять с более высоким разрешением, а изображать с более низким, используя усреднение некоторого типа для получения атрибутов пиксела с более низким разрешением.

Равномерное усреднение окружающих пикселей для уменьшения разрешения в 2 и 4 раза демонстрируется на рис. а. Каждый дисплейный пиксел делится на подпиксели в процессе формирования раstra более высокого разрешения. Для получения атрибутов дисплейного пиксела определяются атрибуты в центре каждого подпиксела, которые затем усредняются.

Можно получить лучшие результаты, если рассматривать больше подпикселей и учитывать их влияние с помощью весов при определении атрибутов.

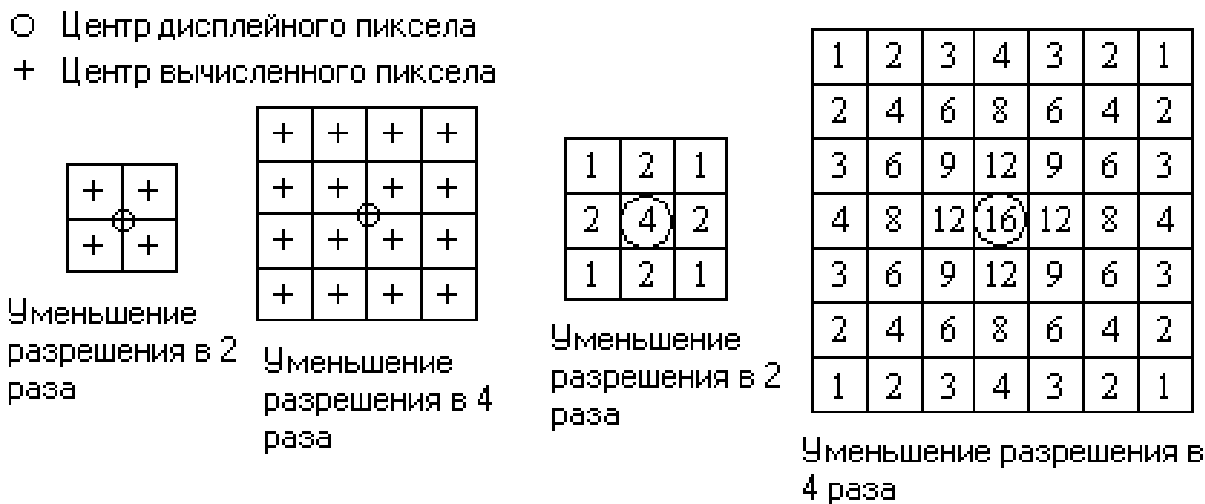


Рис. Усреднение характеристик пиксела: (а) равномерное, (б) взвешенное (числа обозначают относительные веса).

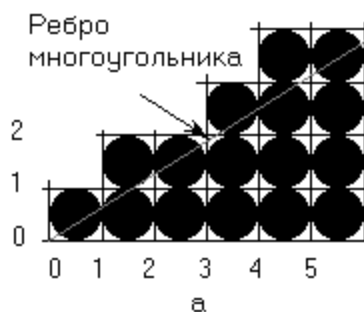
Второй метод устранения искажений изображения состоит в том, чтобы трактовать пиксел не как точку, а как конечную область.

В алгоритмах разложения отрезка в растр и заполнения многоугольника, обсуждавшихся ранее, интенсивность или цвет пиксела определялись интенсивностью или цветом единственной точки внутри области пиксела. В этих методах предполагалось, что пиксел является скорее математической точкой, нежели конечной областью.

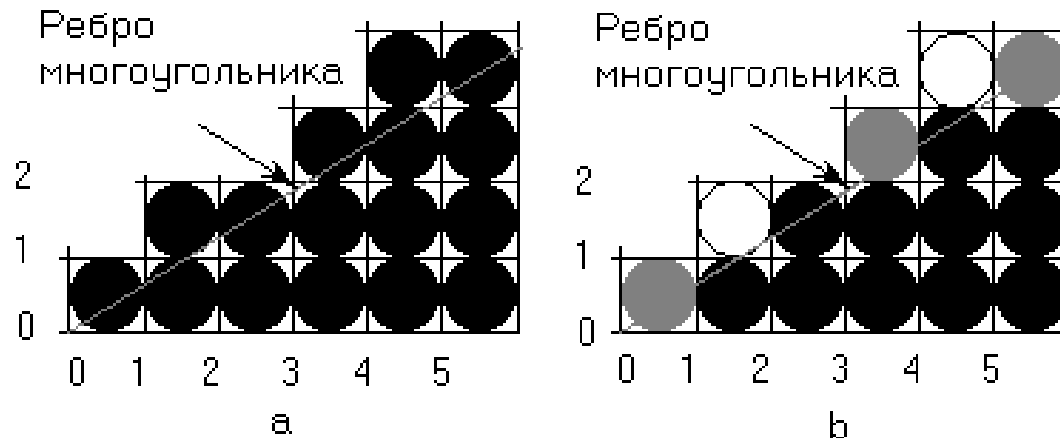
В методах растровой развертки сплошной области определение местоположения пиксела (внутри или вне многоугольника) основывалось на положении центра пиксела. Если центр находился внутри, то активировался весь пиксел. Если центр находился вне, то игнорировалась вся область пиксела. Этот метод необходим для простых двухуровневых изображений, т. е. черных или белых, цвета многоугольника или цвета фона. В результате получается характерное ступенчатое или зазубренное ребро многоугольника или отрезок. Как обсуждалось в предыдущем разделе, основной причиной лестничного эффекта является то, что дискретность отрезка или ребра недостаточна для того, чтобы соответствовать дискретным пикселям экрана дисплея.

При наличии полутонов серого или оттенков цвета, внешний вид ребра или отрезка может быть улучшен размыванием краев.

Простой эвристический метод состоит в том, чтобы устанавливать интенсивность пиксела на ребре пропорционально площади части пиксела, находящейся внутри многоугольника. Эта простая форма устранения ступенчатости иллюстрируется на рис. 2.19, на котором изображено одно ребро многоугольника с тангенсом угла наклона $5/8$.



Внутренняя часть многоугольника расположена справа. На рис. а представлено ребро, разложенное в растр с помощью стандартного алгоритма Брезенхема с использованием только двух уровней интенсивности. Изображение имеет характерный зазубренный или ступенчатый вид.



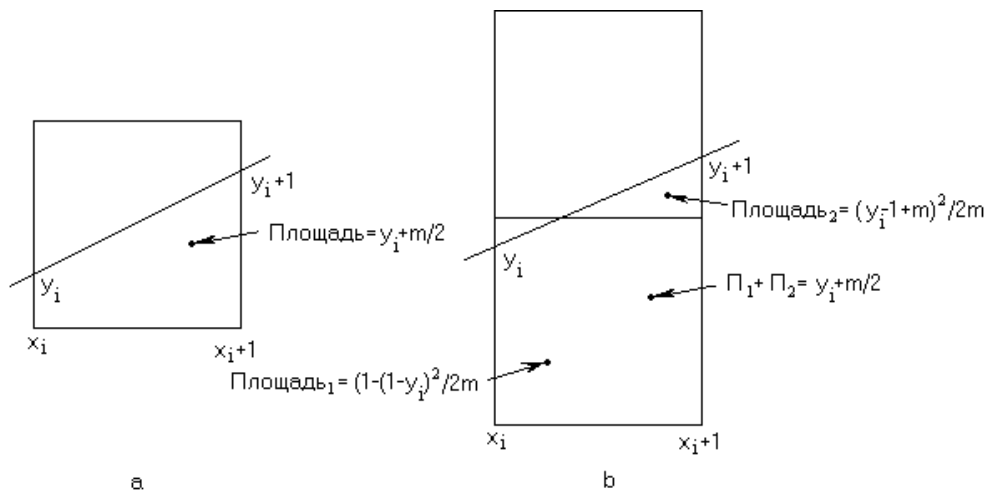
На рис. б для выбора одного из восьми (от 0 до 7) уровней интенсивности используется площадь части пиксела, находящейся внутри многоугольника.

Так как эта площадь для некоторых пикселов меньше одной восьмой, то некоторые пикселы, полностью черные на рис. а, окрашены в белый цвет на рис. б.

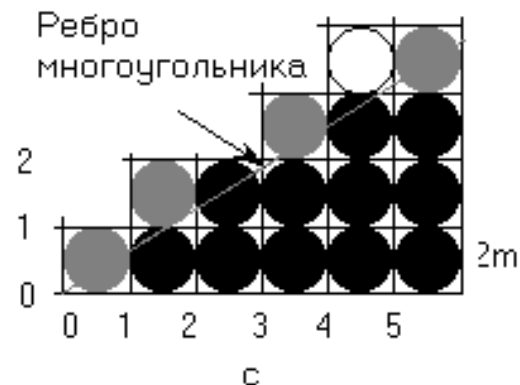
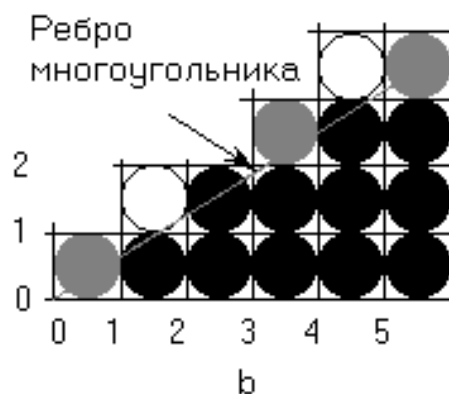
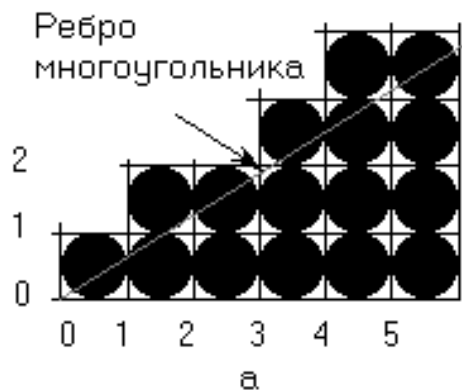
В результате простой модификации алгоритма Брезенхема можно получить аппроксимацию площади части пиксела, находящейся внутри многоугольника.

Эту аппроксимацию можно использовать для модуляции интенсивности. Как показано на рис., при пересечении пиксела и отрезка с тангенсом угла наклона m ($0 \leq m \leq 1$) может быть задействован либо один, либо два пиксела.

Если пересекается только один пиксел (а), то площадь правее и ниже отрезка равна $y_i + m/2$. Если же надо рассмотреть два пиксела (б), то площадь нижнего пиксела составляет $1 - (1 - y_i)^2/2m$, а верхнего - $(y_i - 1 + m)^2/2m$.



Для отрезка в первом октанте с тангенсом угла наклона $0 \leq m \leq 1$ площадь верхнего пиксела может быть достаточно мала для того, чтобы ее можно было проигнорировать в вышеописанном эвристическом методе, например, для пиксела (1, 1) на рис.б. Однако объединение этой площади с площадью нижнего пиксела позволит более реалистично представить ребро многоугольника. Суммарная площадь для двух пикселов равная, $y_i + m/2$.



Если к ошибке в исходном алгоритме Брезенхема добавить величину $w = 1 - m$, т. е. ввести преобразование $\bar{e} = e + w$, то $0 \leq \bar{e} \leq 1$.

Теперь ошибка \bar{e} - это мера площади той части пиксела, которая находится внутри многоугольника, т. е. $y_i + m/2$.

В связи с этими модификациями начальное значение ошибки равно $1/2$ поэтому для первого пиксела алгоритм всегда будет выдавать значение интенсивности, равное половине максимальной.

Более реалистичное значение для первого пиксела дает перемещение оператора активирования пиксела на другое место. Более того, можно получить непосредственно значение интенсивности, а не десятичную дробь от ее максимума с помощью умножения на максимальное число доступных уровней интенсивности I следующих величин: тангенса угла наклона (m), весового коэффициента (w) и ошибки \bar{e} .

Модифицированный алгоритм Брезенхема выглядит следующим образом:

Модифицированный алгоритм Брезенхема с устранением ступенчатости для первого квадранта

```
x = x1
y = y1
Δx = x2 - x1
Δy = y2 - y1
m = (I · Δy)/Δx
w = I - m
ē = 1/2
Plot (x, y, m/2)
while (x < x2)
  If ē < w then
    x = x + 1
    ē = ē + m
  else
    x = x + 1
    y = y + 1
    ē = ē - w
  end if
  Plot (x, y, ē)
end while
finish
```

/ инициализация
переменных

отрезок проводится из (x_1, y_1) в (x_2, y_2)
 I - число доступных уровней
интенсивности
переменные целого типа

Интенсивность для первого пиксела предполагает, что отрезок начинается с адреса пиксела. Результаты для отрезка с тангенсом угла наклона $m = 5/8$ и восемью уровнями интенсивности представлены на рис. 2.19,с. Распространить работу алгоритма на другие октанты можно тем же способом, что и для основного алгоритма Брезенхема.