Растровые алгоритмы

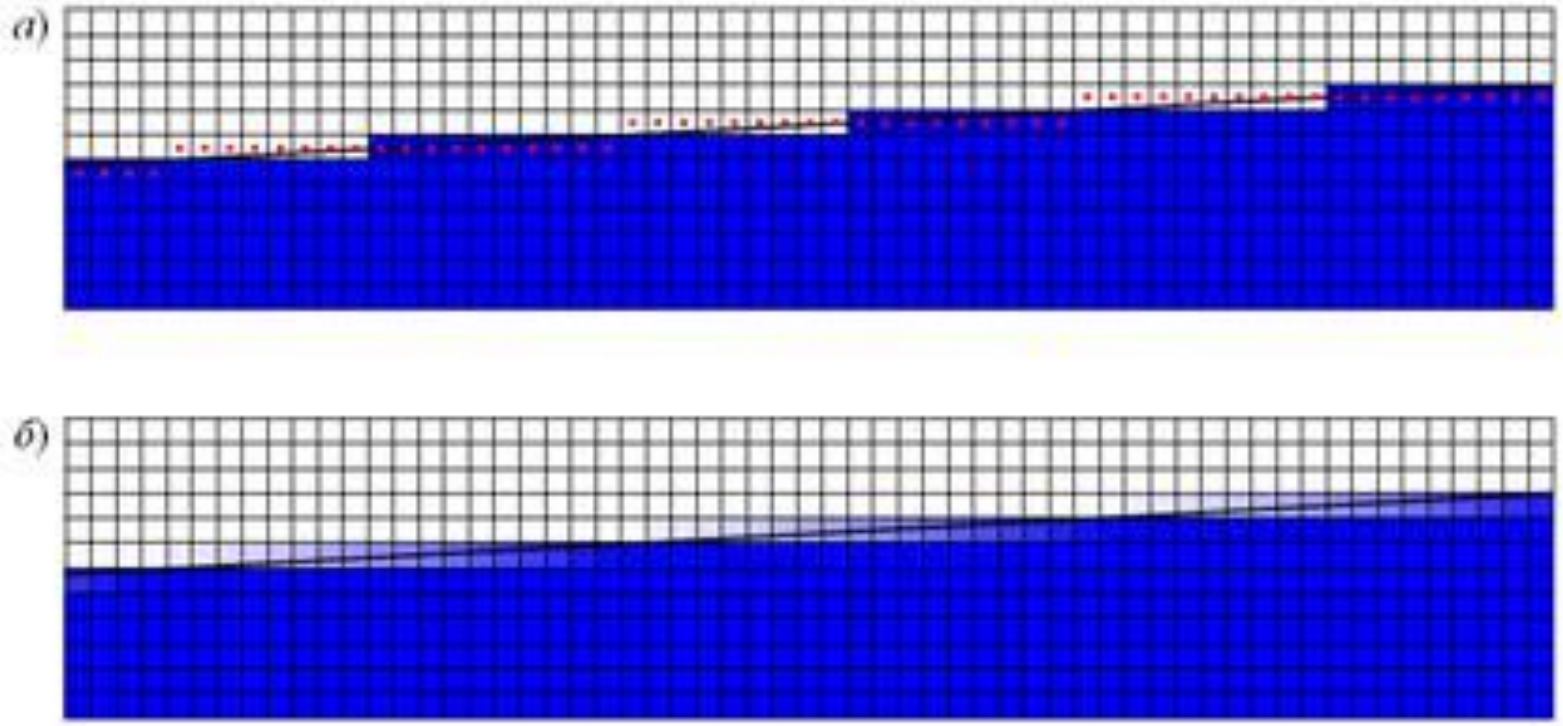
Лекция 4

Лектор – доцент каф. ВТ ИК ТПУ
Болотова Юлия Александровна


Растровые алгоритмы
Алгоритмы растеризации
Алгоритмы обработки растровых изображений
Алгоритмы перевода графических примитивов в растровую форму
Алгоритмы заполнения областей и многоугольников
Регулировка яркости и контрастности
Масштабирование изображений
Геометрические преобразования
Алгоритмы фильтрации

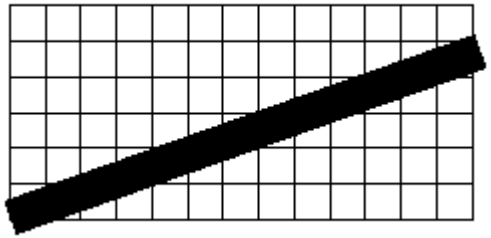
Методы устранение ступенчатости

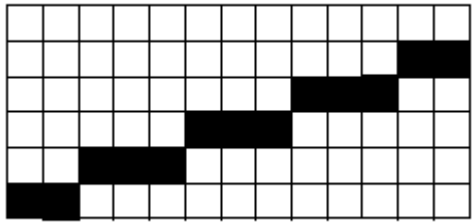
Крыша, фонари, ограждения на мосту


а)
б)

Почему возникает ступенчатость?

Главная причина - "сетка" пикселей на компьютерном мониторе.





2 способа решения:

1. увеличение частоты выборки;
2. использование полутонов.

I. Увеличение частоты выборки

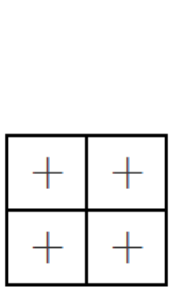
Но! Как это сделать, если разрешение отображающего устройства конечно?

Можно вычислять изображение с более высокой частотой растра, а отображать его с более низкой: (*После расчета изображения с высоким разрешением, процессор уменьшает размер картинки до разрешения дисплея, причем эта операция производится с соответствующей фильтрацией*)

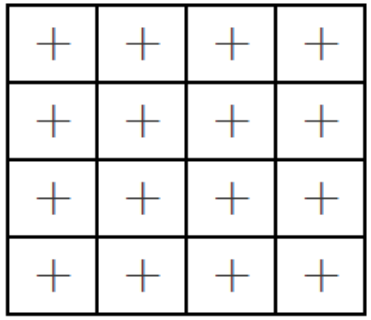
Увеличение частоты выборки

$$F_{x,y} = \frac{1}{K} \sum_{i=i_{min}}^{i_{max}} \sum_{j=j_{min}}^{j_{max}} P_{x+j,y+i} \cdot M_{i-i_{min},j-j_{max}},$$

Среднее значение



$P_{1,1}$



$F_{1,1}$

1	1	1
1	1	1
1	1	1

K=1/9

Взвешенное значение

1	2	1
2	3	2
1	2	1

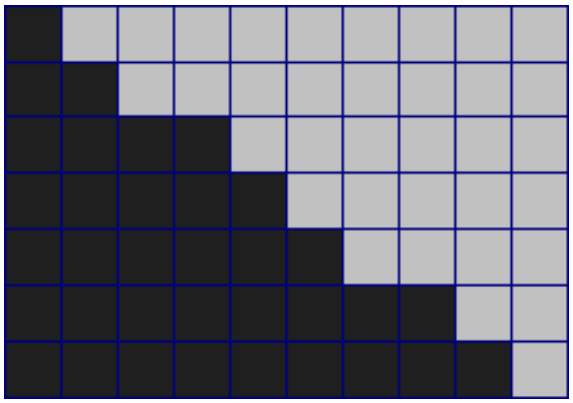
K=1/15

1	2	3	4	3	2	1
2	4	6	8	6	4	2
3	6	9	12	9	6	3
4	8	12	16	12	8	4
3	6	9	12	9	6	3
2	4	6	8	6	4	2
1	2	3	4	3	2	1

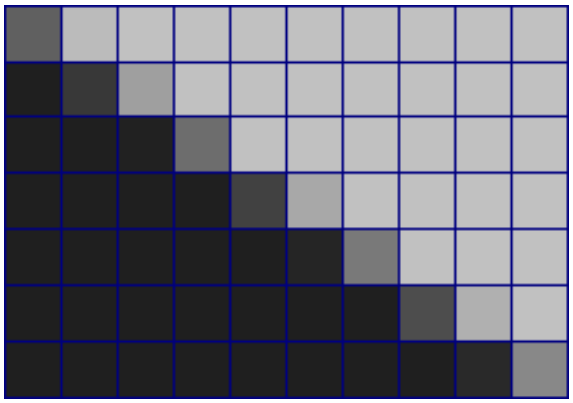
$F_{x,y}$ – результирующее значение;
K – нормировочный коэффициент.

Увеличение частоты выборки

Ступенчатая

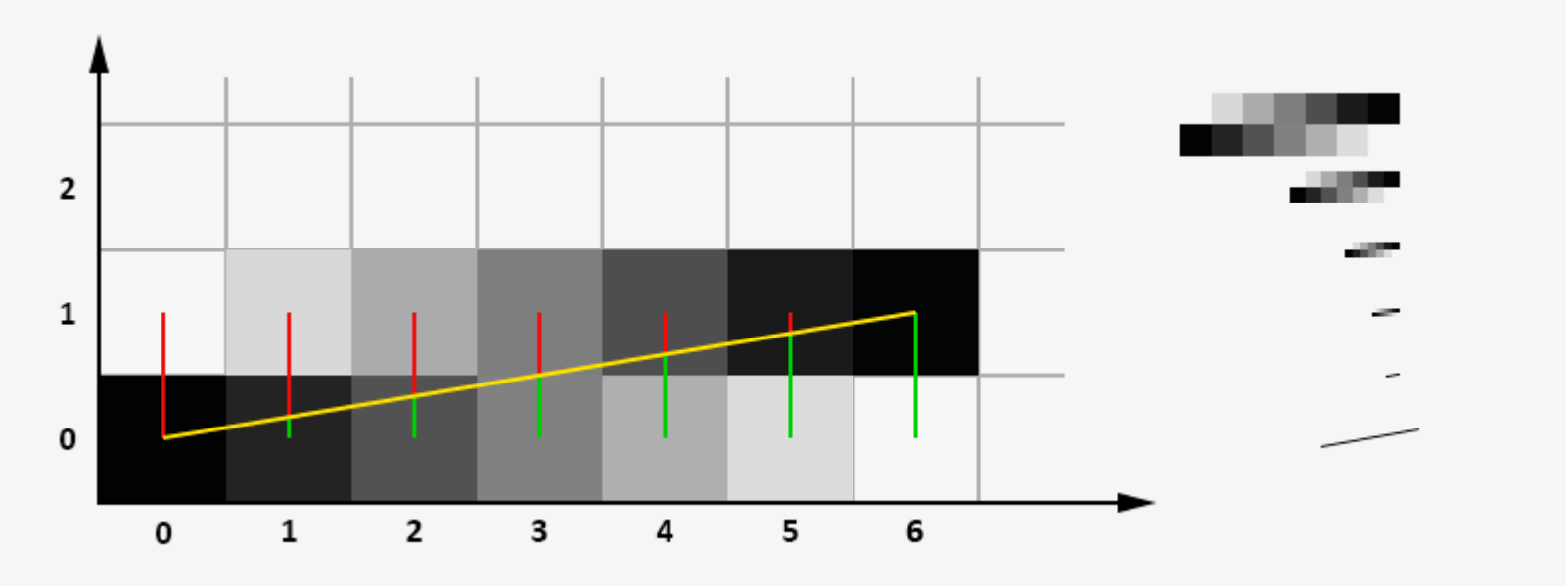


Сглаженная



II. Использование полутонов

- При построении прямой на каждом шаге ведётся расчет для двух ближайших к прямой пикселей. 100% интенсивности делится между пикселями, которые ограничивают векторную линию с двух сторон.

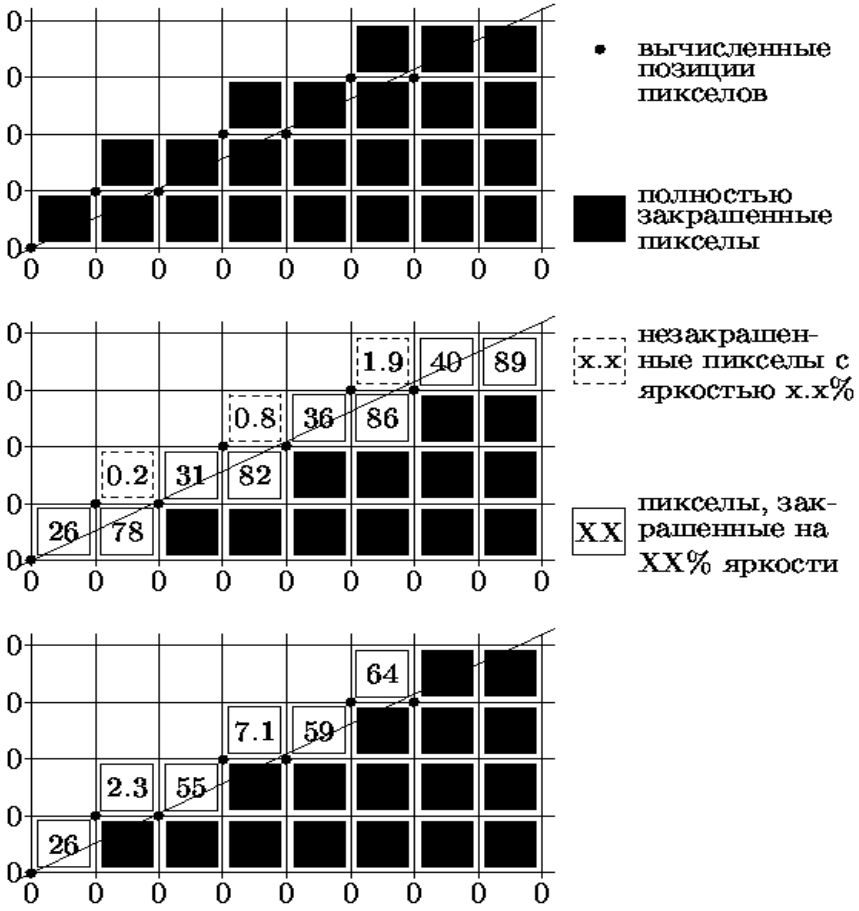


Растеризация линий.

Модификация алгоритма Брезенхэма со сглаживанием границы

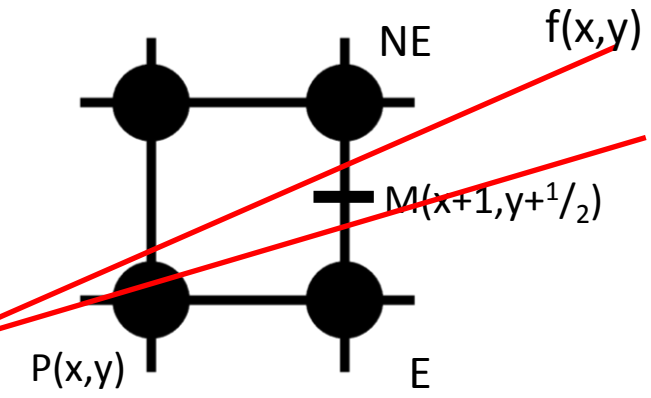
Модификация растеризации линии по Брезенхэйму с целью сглаживания границы:

- Полная закраска
- Неполная закраска 8-связный вариант
- Неполная закраска, 4-связный вариант



Для оценки процента закраски (p) может быть использована величина функции ошибки:
p=100·(d-err)/d

$$f(x,y) = dy \cdot x - dx \cdot y - (x1 \cdot dy - y1 \cdot dx)$$



$$f(M) = f(x+1, y+1/2) =$$

$$dy \cdot (x+1) - dx \cdot (y+1/2) - C =$$

$$dy \cdot x + dy - dx \cdot y - dx/2 - C$$

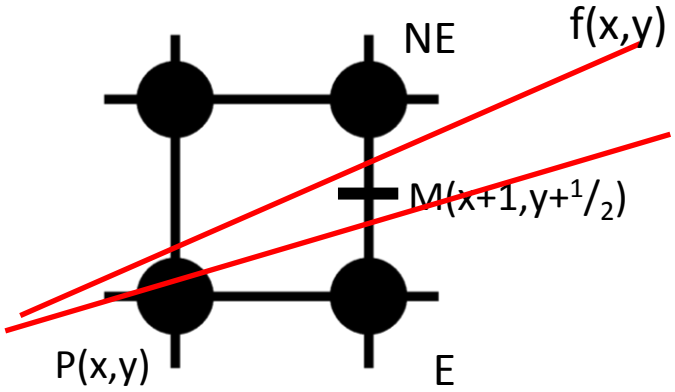
$$f(E) = f(x+1, y) = dy \cdot (x+1) - dx \cdot (y) - C =$$

$$dy \cdot x + dy - dx \cdot y - C = f(M) + dx/2$$

$$f(NE) = f(x+1, y+1) = dy \cdot (x+1) - dx \cdot (y+1) - C =$$

$$dy \cdot x + dy - dx \cdot y - dx - C = f(M) - dx/2$$

Расчет степени закраски



$$f(E) = f(M) + dx/2$$
$$f(NE) = f = f(M) - dx/2$$
$$delta = f(E) - f(NE) = dx$$

Составляем пропорцию :

$$delta = dx \; - \; 100 \; \% \quad \textit{закраски} \;\; ,$$
$$f(E) = 100 \; [f(M) + dx/2]/dx \;\; ,$$
$$f(NE) = 100 \; [f(M) - dx/2]/dx \;\; .$$

Модификация алгоритма Брезенхэйма со сглаживанием грани

```
int x, y, dx, dy, incrE, incrNE, e, color;
dx = x2 - x1;
dy = y2 - y1;
e = 2 * dy - dx;
incrE = 2 * dy;
incrNE = 2 * dy - 2 * dx;
color = 100;
x = x1; y = y1;
SetPixel(x, y, color);
int count = dx;
while (count > 0)
{
    count = count - 1;
    if (e > 0)
    {
        y = y + 1;
        e = e + incrNE;
        color = 50 * (e / dx - 1);
    }
    else
    {
        e = e + incrE;
        color = 50 * (e / dx + 1);
    }
    x = x + 1;
    SetPixel(x, y, color);
}
```

```
int x, y, dx, dy, incrE, incrNE, e, color, mult;

dx = x2 - x1;
dy = y2 - y1;
e = 2 * dy - dx;
incrE = 2 * dy;
incrNE = 2 * dy - 2 * dx;
color = 100;
mult = 50/dx;

x = x1; y = y;
SetPixel(x, y, color);
int count = dx;
while (count > 0)
{
    count = count - 1;

    if (e > 0)
    {
        y = y + 1;
        e = e + incrNE;
        color = mult*(e - dx);
    }
    else
    {
        e = e + incrE;
        color = mult * (e + dx);
    }
    x = x + 1;
    SetPixel(x, y, color);
}
```

Простейшие методы обработки изображений


Растровые алгоритмы
Алгоритмы растеризации
Алгоритмы обработки растровых изображений
Алгоритмы перевода графических примитивов в растровую форму
Алгоритмы заполнения областей и многоугольников
Регулировка яркости и контрастности
Масштабирование изображений
Геометрические преобразования
Алгоритмы фильтрации

Яркость и контраст

Являются субъективными характеристиками изображения, воспринимаемыми человеком.

- **Яркость** определяет, насколько сильно цвета пикселей отличается от черного цвета. (*мат. ожидание значений выборки*)
- **Контраст** определяет , насколько большой разброс имеют цвета пикселей изображения. (*дисперсия значений выборки*)

Гистограмма

- Статистическая характеристика, показывающая количество пикселей изображения (*N*), обладающих определенным уровнем яркости (*I*).

Изменение яркости и контраста

- Полутоновое изображение: $I_{x,y} = f_{x,y}(r, g, b)$
- Цветное изображение: $IR_{x,y} = f_{x,y}(r)$;
  $IG_{x,y} = f_{x,y}(g)$;
  $IB_{x,y} = f_{x,y}(b)$;

Изменение яркости и контраста связано с изменением интенсивности:

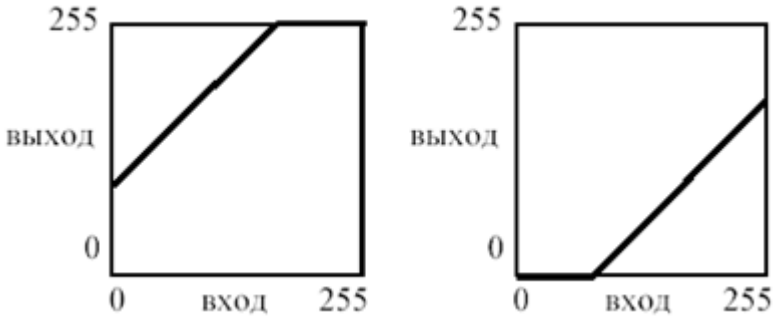
$$\boldsymbol{I_{вых} = k * I_{вх} + b}, \textit{ где}$$

b - параметр, влияющий на яркость

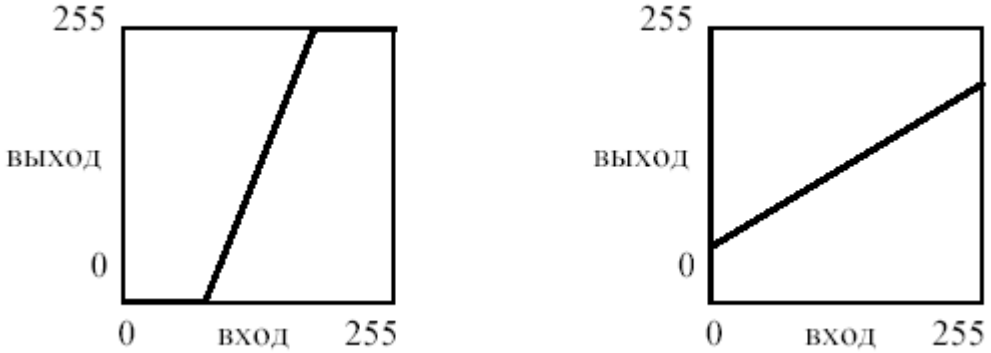
k - параметр, влияющий на контраст

Изменение яркости и контраста

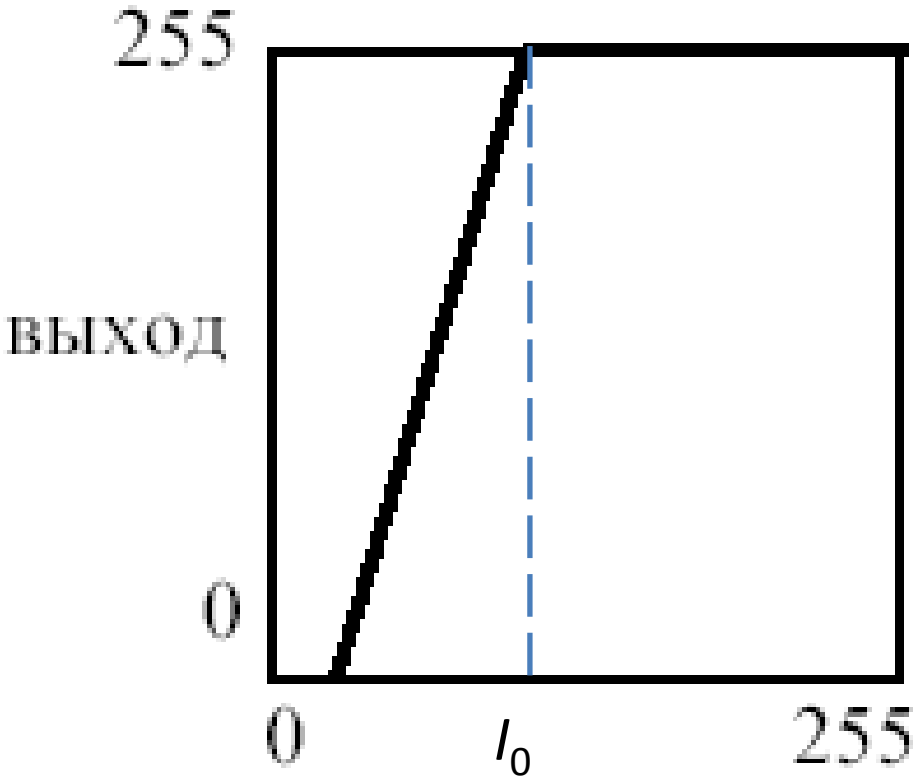
- Изменение яркости



- Изменение контраста



Что происходит, когда прямая становится параллельной оси I_0?



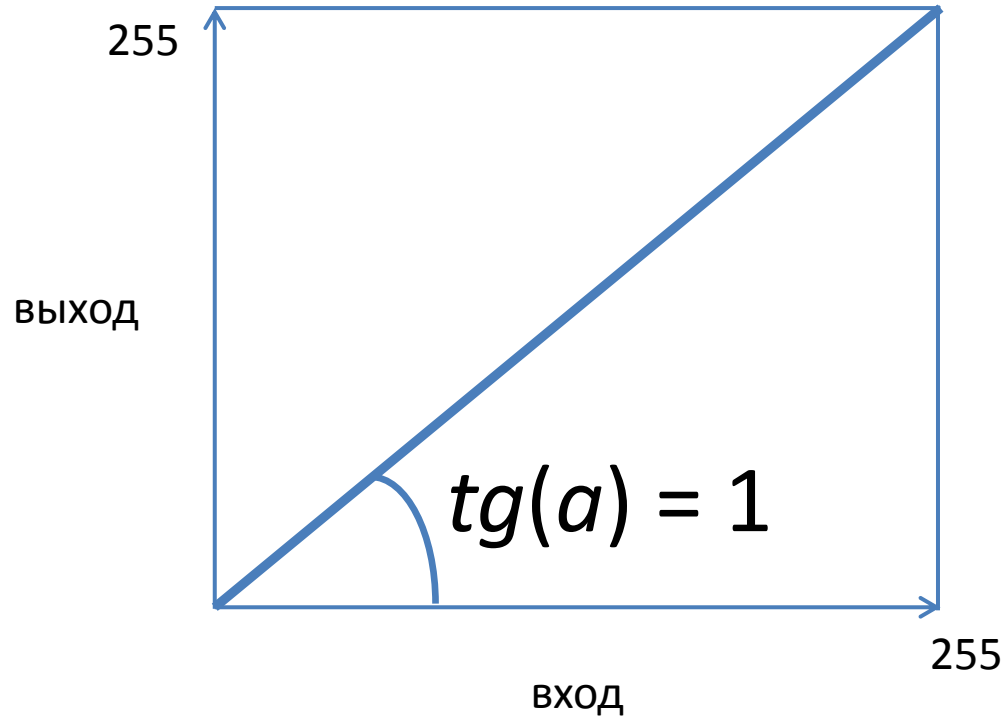
Арифметика с насыщением

При возникновении переполнений или заёмов фиксируется наибольшее представимое или наименьшее представимое значение соответственно.

Если $I_{вых} > 255$, то $I_{вых} = 255$

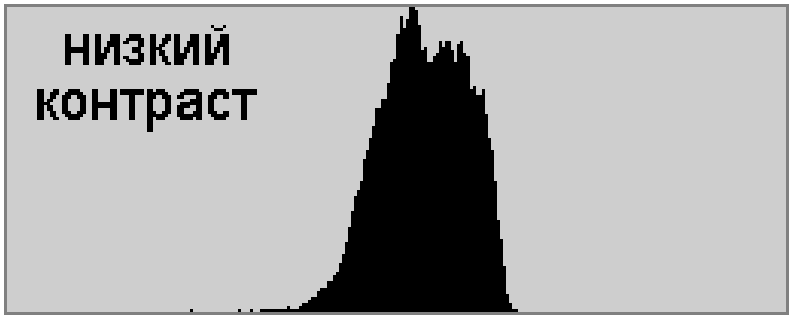
Если $I_{вых} < 0$, то $I_{вых} = 0$

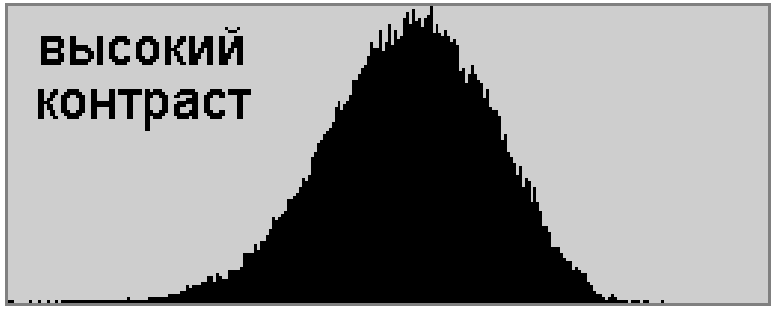
Что происходит в этом случае?



Изменение контраста

Увеличение контраста - растяжение гистограммы





Изменение яркости

Применение изменений яркости и контраста

Преобразования яркости и контраста могут быть применены к **полутоновому** изображению, **по-отдельности** и **совместно** к компонентам r, g, b цветного изображения.


Растровые алгоритмы
Алгоритмы растеризации
Алгоритмы обработки растровых изображений
Алгоритмы перевода графических примитивов в растровую форму
Алгоритмы заполнения областей и многоугольников
Регулировка яркости и контрастности
Масштабирование изображений
Геометрические преобразования
Алгоритмы фильтрации

Масштабирование изображений

- Позволяет сжать/растянуть изображение по горизонтали и/или вертикали.

При этом задаются масштабные коэффициенты, насколько нужно сжать/растянуть изображение.

Как определить цвета при изменении размеров?

Определение цвета пикселя

- **Метод ближайшего соседа.** Цвет пикселя принимается равным цвету ближайшего к нему пикселя в исходном изображении.
- **Использование интерполяции.** Цвет пикселя вычисляется как значение некоторой интерполирующей функции от цветов соседних пикселей в исходном изображении.

Метод ближайшего соседа

$$C_{new}[i][j] = C_{old}[k_1 * i][k_2 * j],$$

$$i = 0..H_{old} - 1, \qquad j = W_{old} - 1,$$

$$k_1 = H_{old}/H_{new}, \qquad k_2 = W_{old}/W_{new}$$

W – ширина изображения в пикселях;

H – высота изображения в пикселях;

K_1, k_2 – масштабные коэффициенты

Метод ближайшего соседа

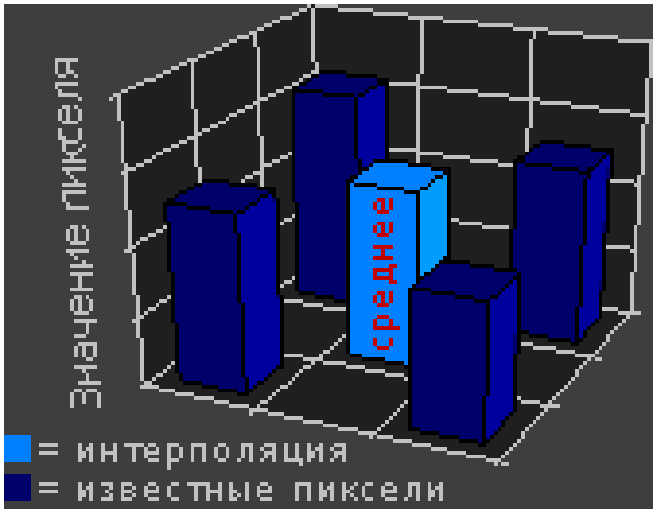
Метод прост, но не всегда дает приемлемое качество.

1. Блочная структура изображения при увеличении.

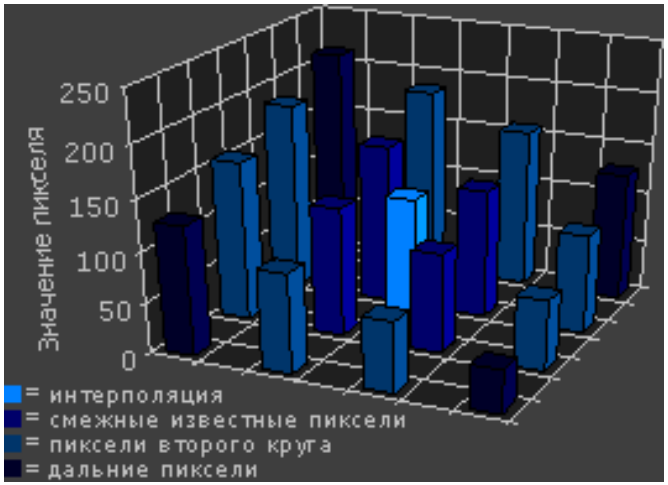
2. Исчезновение информативных пикселей при уменьшении.

Интерполяция

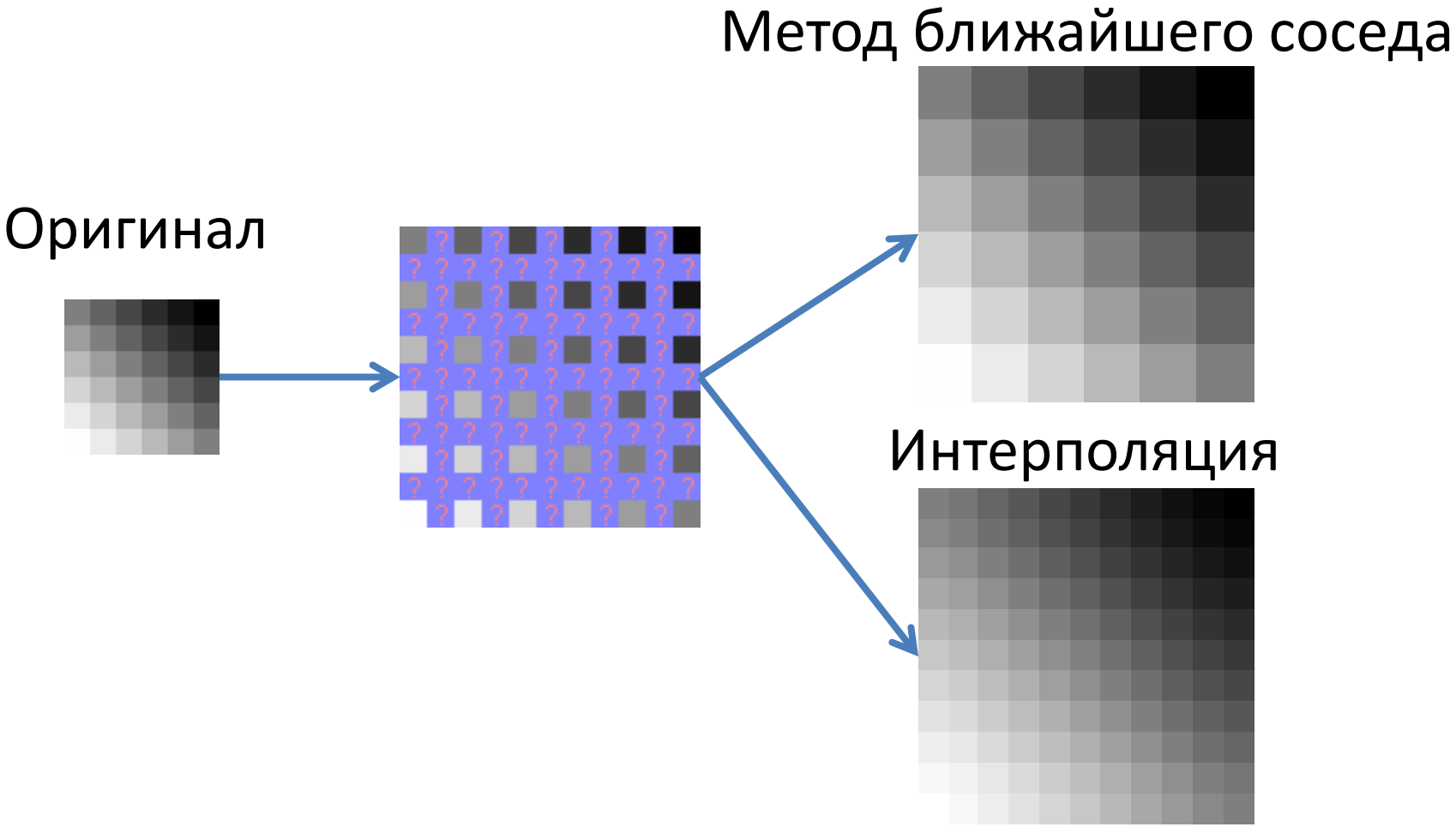
Позволяет получить изображение с более высокой точностью.



- Билинейная интерполяция – взвешенная сумма ближайших четырех пикселей исходного изображения.
- Бикубическая интерполяция - значение функции в искомой точке вычисляется через ее значения в 16 соседних точках.



Пример масштабирования изображения



Ближ. сосед

Билинейная

Бикубическая


Растровые алгоритмы
Алгоритмы растеризации
Алгоритмы обработки растровых изображений
Алгоритмы перевода графических примитивов в растровую форму
Алгоритмы заполнения областей и многоугольников
Регулировка яркости и контрастности
Масштабирование изображений
Геометрические преобразования
Алгоритмы фильтрации

Преобразование поворота

Позволяет поворачивать изображение на заданный угол.

Возможны 2 варианта поворота:

1. Области изображения, вышедшие за его границы отсекаются, а незаполненные части заполняются каким-либо цветом.
2. Рассчитывает новый размер изображений на основе угла поворота таким образом, чтобы повернутое изображение целиком поместилось в новые размеры.

Поворот растрового изображения

$$C_{new}[i][j] = \begin{cases} C_{old}[a][b], a \in [0, H_{old} - 1] \wedge b \in [0, W_{old} - 1]; \\ C, a \notin [0, H_{old} - 1] \vee b \notin [0, W_{old} - 1]; \end{cases}$$

$$a = \left| i \cdot \sin(\varphi) + \frac{H_{new}}{2} \right|; \; b = \left| j \cdot \cos(\varphi) + \frac{W_{new}}{2} \right|;$$

$$i = \overline{0, H_{old} - 1}, \; j = \overline{0, W_{old} - 1}.$$


Растровые алгоритмы
Алгоритмы растеризации
Алгоритмы обработки растровых изображений
Алгоритмы перевода графических примитивов в растровую форму
Алгоритмы заполнения областей и многоугольников
Регулировка яркости и контрастности
Масштабирование изображений
Геометрические преобразования
Алгоритмы фильтрации

Цифровые фильтры

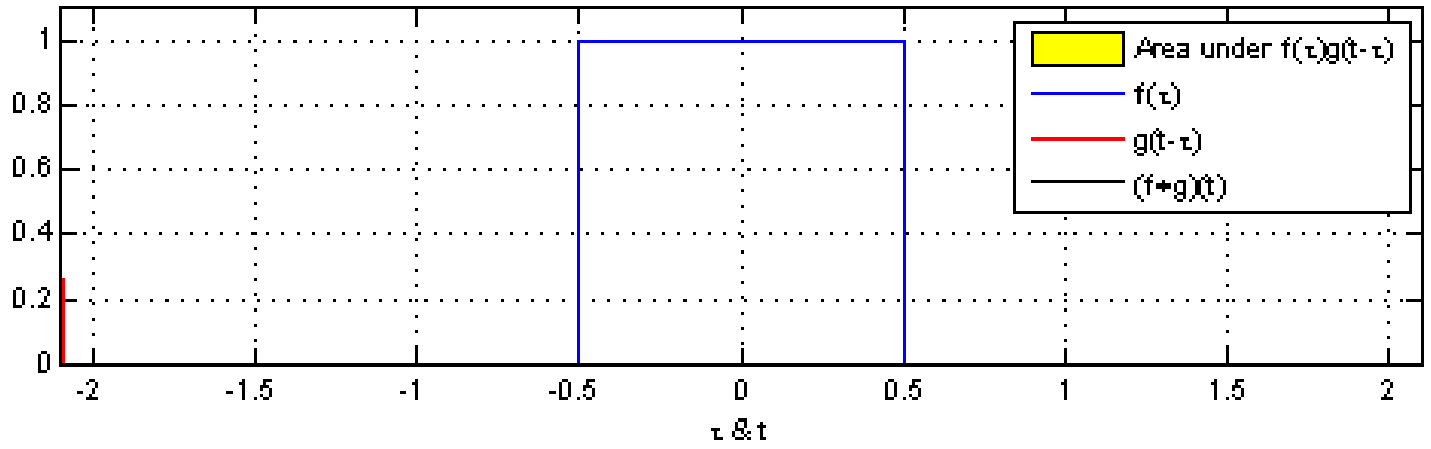
- применяются при обработке изображений;
- позволяют накладывать на изображение различные эффекты: **размытие**, **увеличение резкости**, **удаление шумов** и т.д.

Работа любых фильтров основана на понятии свёртки.

Свёртка

Свёртка фу́нкций — операция в функциональном анализе, показывающая «схожесть» одной функции с отражённой и сдвинутой копией другой.

$$(f * g)(x) \stackrel{\text{def}}{=} \int_{\mathbf{R}^d} f(y)\, g(x-y)\, dy = \int_{\mathbf{R}^d} f(x-y)\, g(y)\, dy.$$



Линейные фильтры

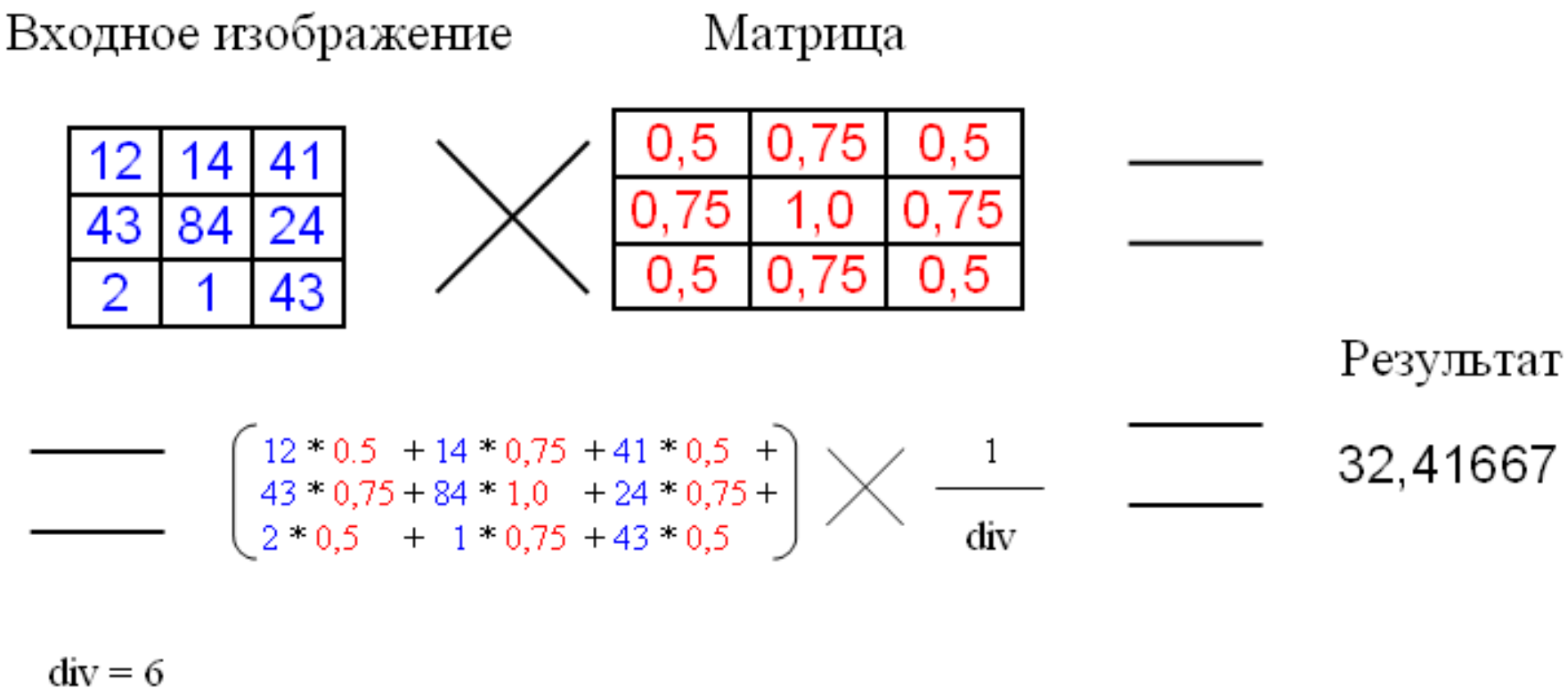
Пусть задано исходное полутоновое изображение ***A***, обозначим интенсивности его пикселей ***A(x, y)***.

Линейный фильтр определяется вещественнозначной функцией ***F***, заданной на растре. Данная функция называется **ядром фильтра**, а сама фильтрация производится при помощи операции дискретной свертки (взвешенного суммирования):

$$B(x,y) = \sum_i \sum_j F(i,j) \cdot A(x+i, y+j).$$

Результатом служит изображение ***B***.

Пример наложения фильтра



div = 6

Свёртка

- Изображение представляет собой функцию яркости $I = f(x, y)$;
- Вторая функция называется **ядром свертки** и представляет собой матрицу коэффициентов $g=f(x, y)$:

$-k/8$	$-k/8$	$-k/8$
$-k/8$	$k+1$	$-k/8$
$-k/8$	$-k/8$	$-k/8$

Увеличение резкости

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

Размытие изображения

Характеристики цифровых фильтров

- **Размер фильтра.** Чем больше размер фильтра, тем большее число соседних пикселей «влияют» на значение результирующего пикселя.
- **Импульсная характеристика фильтра.** Представляет собой изображение, которое получается в результате обработки черного изображения, в центре которого располагается белая точка.

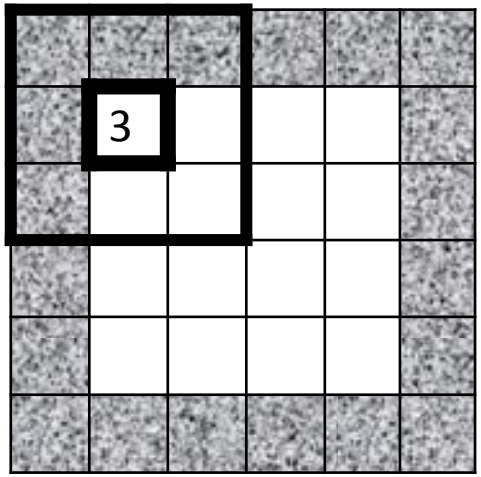
Применение цифровых фильтров

- Увеличение резкости.
- Удаление шумов.
- Выделение границ объектов.

Условия на границе

1	5	7	3	8	5
1	1	5	2	2	2
1	1	5	2	2	2
5	5	5	1	1	1
7	7	7	1	1	1
7	7	7	1	1	1

Исходное изображение



Результирующее изображение

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Ядро фильтра

Условия на границе. Возможные решения

1. Не проводить фильтрацию для таких пикселей, обрезав изображение по краям или закрасив их, к примеру, черным цветом.

2. Не включать соответствующий пиксель в суммирование, распределив его вес ***F(i, j)*** равномерно среди других пикселей окрестности ***N(x, y)***.

3. Доопределить значения пикселей за границами изображения при помощи экстраполяции.

4. Доопределить значения пикселей за границами изображения, при помощи зеркального отражения.

Сглаживающие фильтры

- Сглаживающие фильтры действуют на изображение аналогично мутному стеклу: изображение становится нерезким, размытым. Простейший ***прямоугольный сглаживающий фильтр*** радиуса ***r*** задается при помощи матрицы размера ***(2r + 1) × (2r + 1)***, все значения которой равны

$$\frac{1}{(2r+1)^2},$$

- а сумма по всем элементам матрицы равна, таким образом, единице. При фильтрации с данным ядром значение пикселя заменяется на усредненное значение пикселей в квадрате со стороной 2r+1 вокруг него.

Примеры сглаживающих фильтров

Гауссовский фильтр

- Гауссовский фильтр – это линейный фильтр имеющий следующее ядро:

$$F_{gauss}(i,j) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{i^2+j^2}{2\sigma^2}\right).$$

- Где σ^2 – дисперсия случайной велечины
- Гауссовский фильтр имеет ненулевое ядро бесконечного размера. Однако ядро фильтра очень быстро убывает к нулю при удалении от точки (0, 0), и потому на практике можно ограничиться сверткой с окном небольшого размера вокруг (0, 0) (например, взяв радиус окна равным 3σ).

Контрастоповышающие фильтры

- Ядро контрастоповышающего фильтра имеет значение, большее 1, в точке (0, 0), при общей сумме всех значений, равной 1.

$$M_1^{contr} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad M_2^{contr} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

- Характерным артефактом линейной контрастоповышающей фильтрации являются заметные светлые и менее заметные темные ореолы вокруг границ.

Пример

эффект от применения фильтра с ядром

$$M_2^{contr}$$

Разностные фильтры

- Разностные фильтры называют **фильтрами, находящими границы.**
- Фильтры Прюита (Prewitt) и Собеля (Sobel):

$$M_1^{prewitt} = \frac{1}{3}\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}. \quad M_1^{sobel} = \frac{1}{4}\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}.$$

Пример

Нелинейные фильтры

- Одним из примеров нелинейного фильтра являтеся пороговая фильтрация. Результатом пороговой фильтрации служит бинарное изображение, определяемое следующим образом:

$$B(x,y) = \begin{cases} 1, & \text{если} \quad A(x,y) > \gamma \\ 0, & \text{иначе} \end{cases}.$$