Фильтрация изображений

Что такое фильтрация изображений?

- Фильтрация изображений – это процесс вычленения сигнала или структуры изображения путем подавления нежелательных или неинтересующих вариаций на изображении.

- Главный механизм – изменение значений пикселов на изображении с помощью некоторой функции от значений пикселов в локальный окрестности.

Фрагмент изображения

Преобразованный фрагмент изображения

Что такое фильтрация изображений?

- Простейший случай: **линейная фильтрация** – замена каждого пиксела линейной комбинацией его соседей.
- Коэффициенты линейной комбинации называют «маской фильтра» («ядром фильтра», «ядром свертки»).

10	5	3
4	5	1
1	1	7

Фрагмент изображения

0	0	0
0	0.5	0
0	1	0.5

Ядро фильтра

	7	

Преобразованный фрагмент изображения

Свертка

- Действие фильтра g на сигнал f можно записать в виде свертки $f \otimes g$.
- В случае непрерывных сигналов f и g:

$$h(x) = f(x) \quad g(x) = \overset{+}{\underset{-}{}} f(x-t)\, g(t)\, dt$$

- В дискретном случае f и g – сигналы, представленные конечным числом отсчетов: $f = \left\{f_k\right\}_{k=K_0}^{K}, g = \left\{g_l\right\}_{l=L_0}^{L}$

$$h_k = f \quad g = \overset{L}{\underset{l=L_0}{}} f_{k-l}\, g_l$$

- Содержательный смысл: осреднение f с весами g.

Свертка для изображений

- Роль конечного дискретного двумерного сигнала f играет изображение I.

- Ядро двумерного линейного фильтра g с конечным носителем представляет собой матрицу.

- Результат фильтрации – результат двумерной свертки:

$$I^{*}[i,j]=I \quad g = \underset{k,l}{} I[i-k,j-l]g[k,l]$$

Виды фильтров

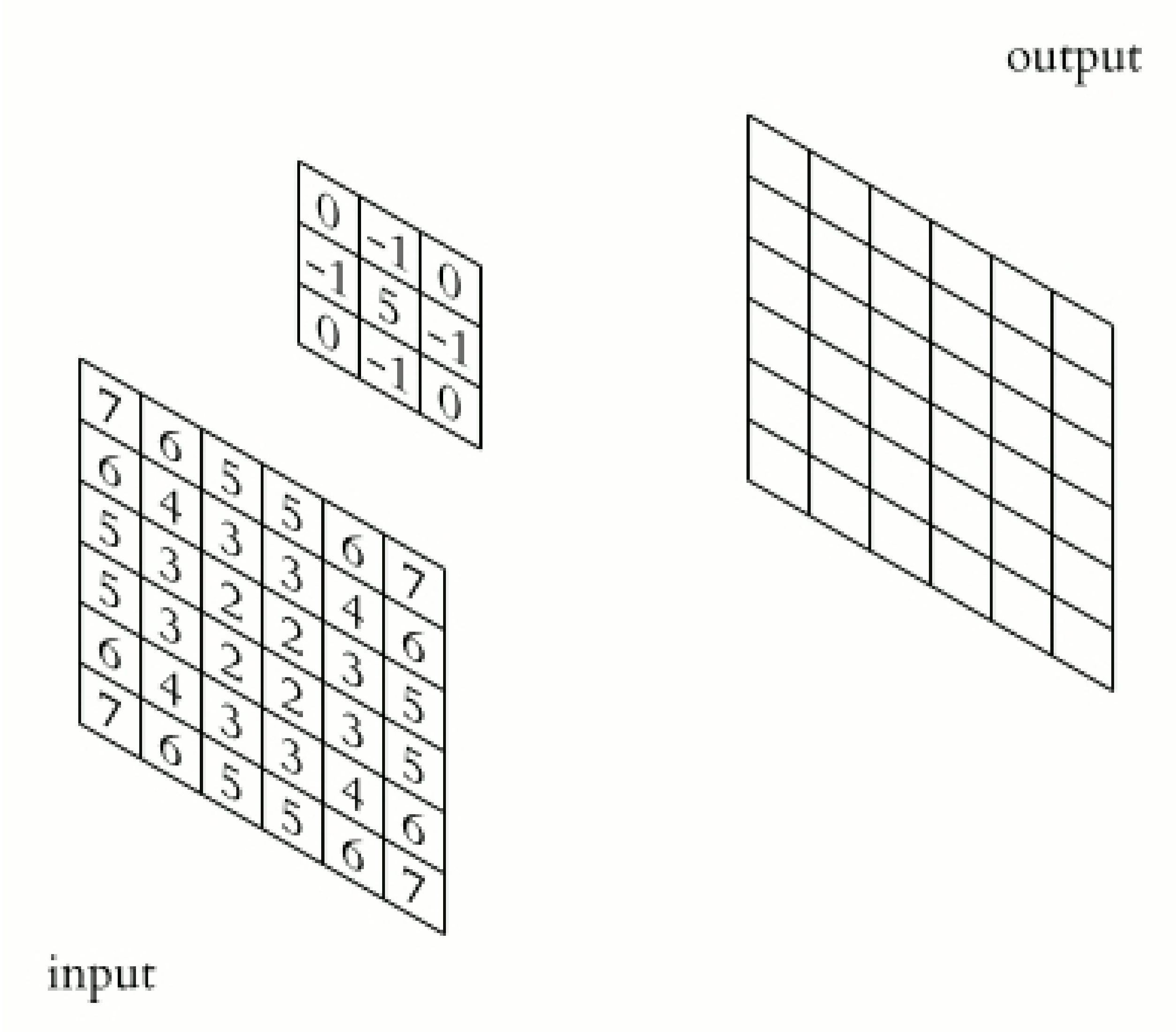
Фильтры подразделяют

- По виду функции, связывающей окрестность пиксела с откликом фильтра, на:
  - Линейные
  - Нелинейные
- По способу записи отклика фильтра в исходное или результирующее изображение на:
  - Рекурсивные
  - Нерекурсивные
- По независимости фильтрующей функции от координат на изображении на:
  - Стационарные
  - Нестационарные

Схемы перемещения маски фильтра

- **P–схема**: ни один элемент маски не выходит за пределы изображения.
- **S–схема**: центральный элемент маски не может выходить за пределы изображения (остальные элементы окна могут)
  - –Усекается окно
  - –Расширяется изображение
- **T–схема**: изображение «сворачивается» в тор так, чтобы его правый край примыкал к левому со сдвигом в один пиксел (конец второй строки к началу первой), а верхний край – к нижнему (последний столбец к первому).

Схемы перемещения маски фильтра



Фильтрация и зашумленные изображения

- Часто изображение содержит шум, затрудняющий дальнейшую обработку
  - Речь идет о **простом** шуме: флуктуации интенсивности; помехи, вносимые камерой; нежелательные эффекты квантования; эффекты, обусловленные конечной точностью компьютера.
  - Речь не идет о сложных эффектах: тени; посторонние объекты и т.п.

Распространенные типы шумов

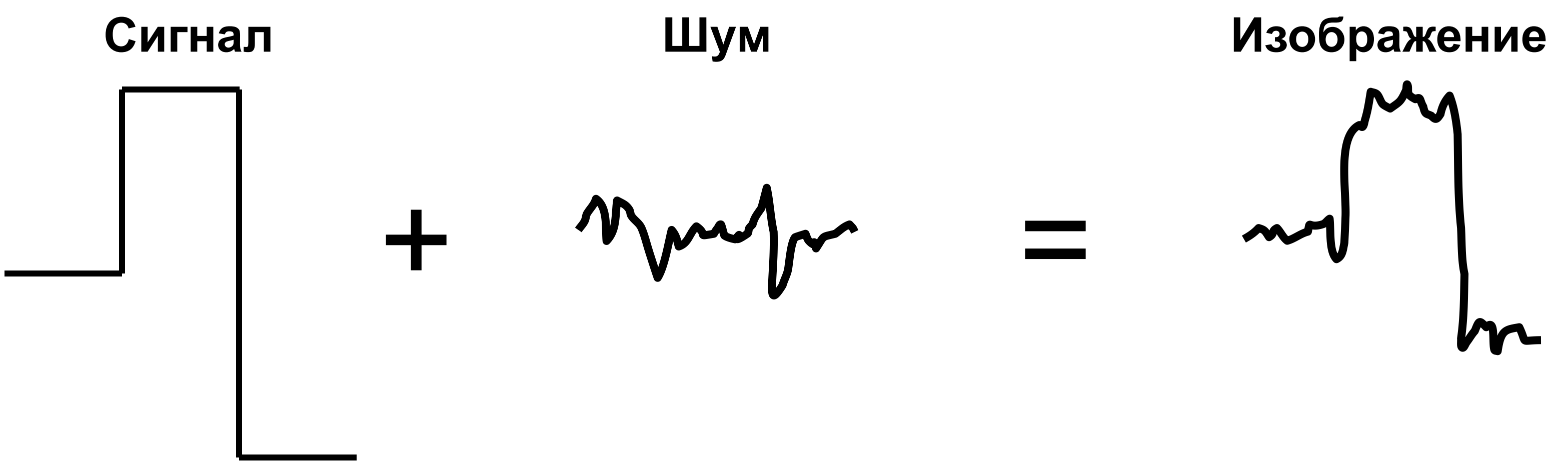
- Импульсный шум («соль и перец»)
  - Единичные светлые пикселы в темных областях изображения и темных пикселы в светлых областях изображения.
  - Обычно является результатом ошибочной классификации в виду вариации свойств объектов или освещения.

Распространенные типы шумов

- Гауссов шум
  - Нормально распределенные отклонения интенсивности.
  - Является результатом неточных измерений (или квантования) интенсивности.
- Белый шум
  - Равномерно распределенные отклонения интенсивности.
  - Является результатом неточных измерений (или квантования) интенсивности.

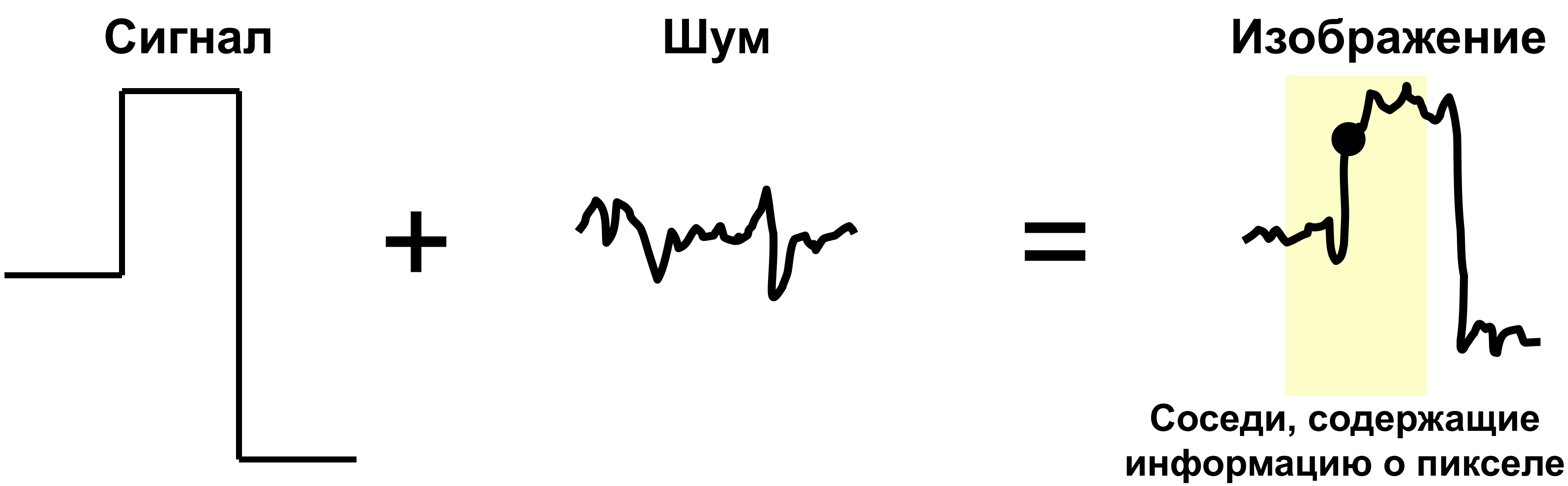
Модель шума

- Предполагается, что шум воздействует на сигнал аддитивно: $I = S + N$ (изображение = сигнал + шум).
- Сигнал — детерминированная величина
- Шум – случайная величина (с нулевым математическим ожиданием).
- Шум не зависит от сигнала.



Шумоподавляющие фильтры

- Соседи пиксела содержат информацию о его интенсивности.
- Более близкие точки содержат больше информации о сигнале, чем более удаленные.
- Усредняющие фильтры подавляют шумовые эффекты.



Усредняющий фильтр

- Ядро – матрица из положительных весов, в сумме дающих 1.
- Заменяет каждый пиксел средним значением интенсивности в его окрестности.
- Если все веса равны, то фильтр называется box–фильтром.

$$g$$

$$1/9 \cdot \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Свойства сглаживающих масок

- Элементы сглаживающих масок положительны, их сумма равна единице, поскольку значение на выходе должно совпадать со значением на входе для областей постоянной интенсивности.
- Степень сглаживания и подавления шума пропорциональна размеру маски.
- Резкие перепады интенсивности сглаживаются пропорцинально размеру маски.

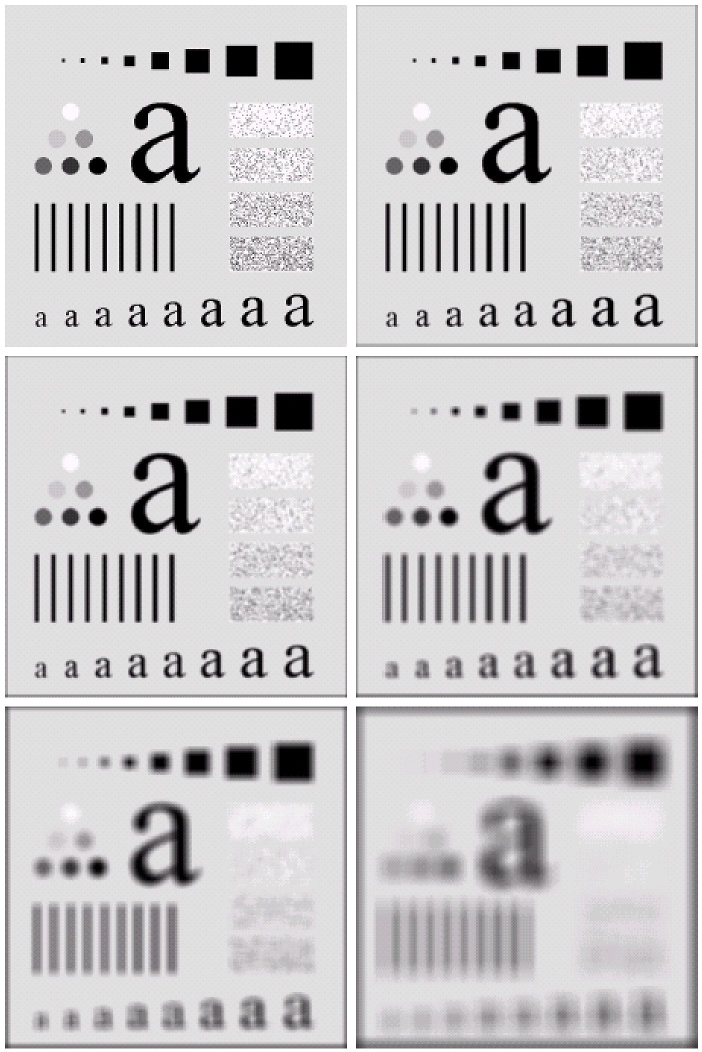
Пример: сглаживание усреднением

Пример: подавление шума усреднением

Использован box–фильтр с маской 5x5

Пример: сглаживание усредняющими фильтрами

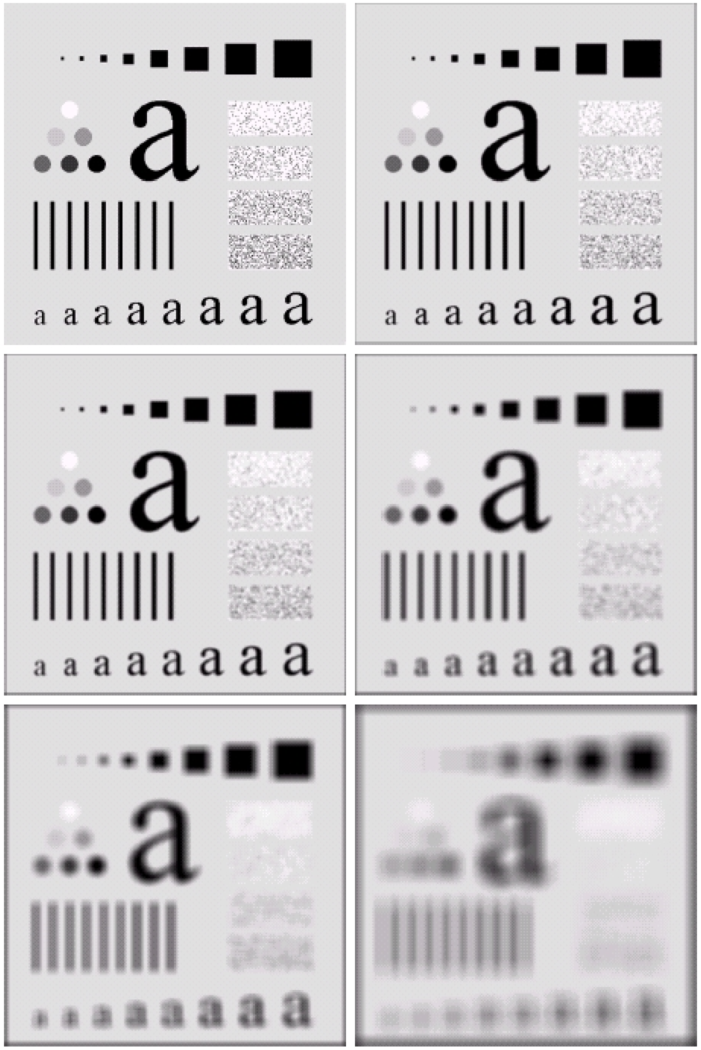
(a) Исходное изображение размера 500x500. (b)–(f) Результаты сглаживания усредняющим фильтром с масками nxn, n=3, 5, 9, 15 и 35 соответственно. Длины сторон черных квадратов в верхней части равны 3, 5, 9, 15, 35, 45 и 55 пикселов соответственно; расстояние между квадратами равно 25 пикселам.



(a) (b)
(c) (d)
(e) (f)

Пример: сглаживание усредняющими фильтрами

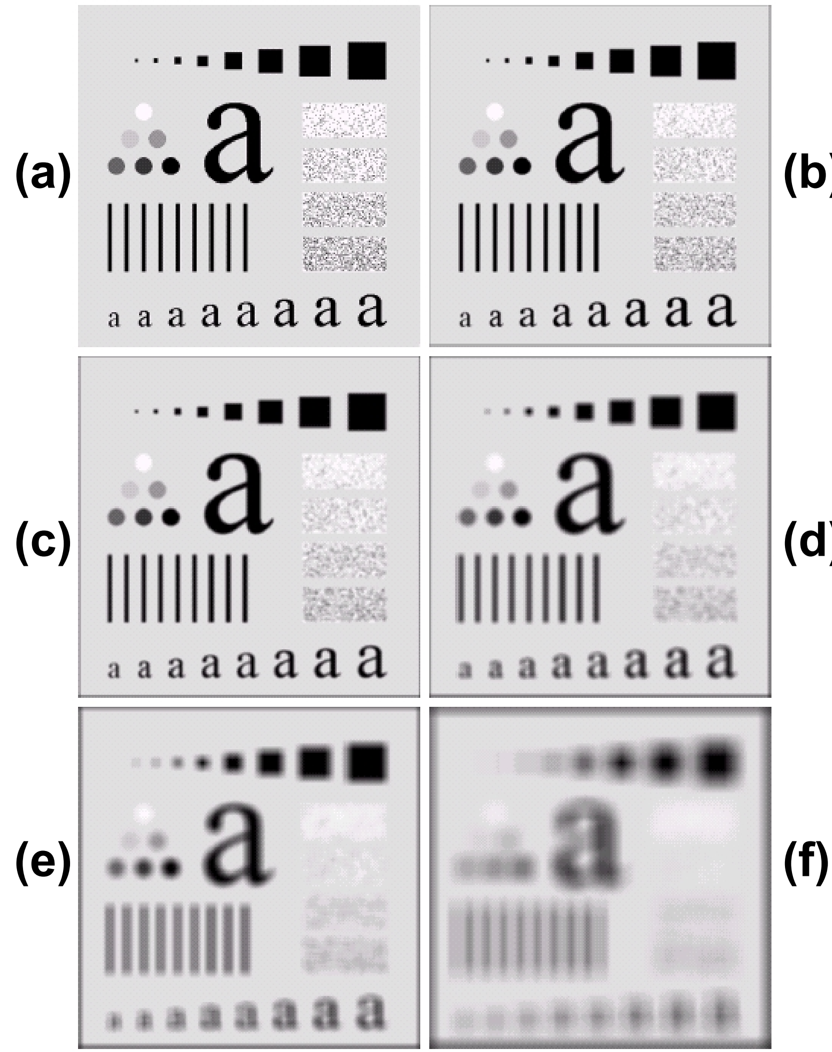
Буквы в нижней части набраны шрифтами от 10 до 24 пунктов с шагом в 2 пункта; большая буква наверху имеет размер 60 пунктов. Вертикальные прямоугольники имеют размеры 5x100 пикселов, расстояние между ними –20 пикселов.



(a) (b) (c) (d) (e) (f)

Пример: сглаживание усредняющими фильтрами

Диаметр кругов – 25 пикселов, расстояние между ними – 15 пикселов; интенсивность кругов меняется от 0% до 100% с шагом в 20%. Интенсивность фона изображения – 10%. Зашумленные прямоугольники имеют размеры 50x120 пикселов.



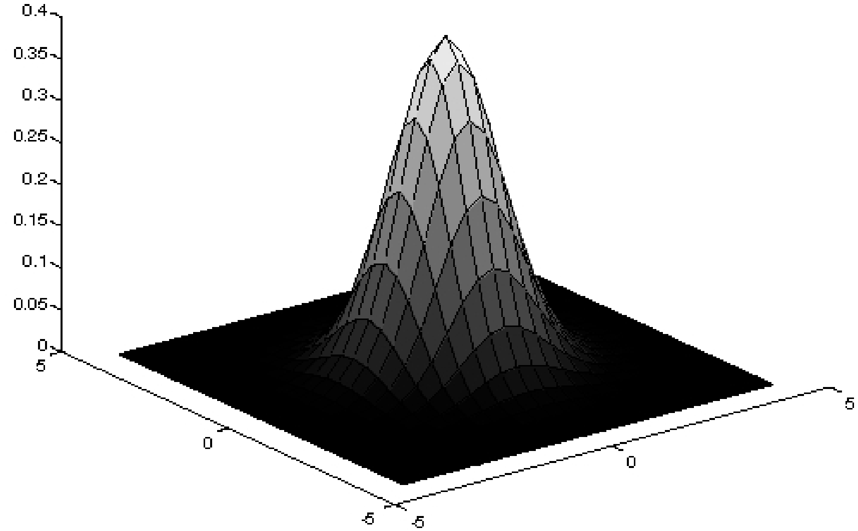
Сглаживание и сигнал

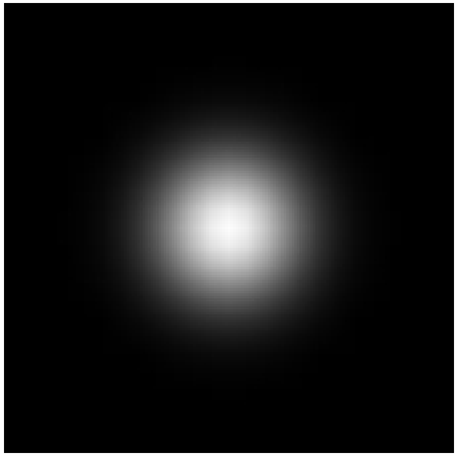
- Помимо подавления шума сглаживание делает более гладким и сигнал тоже.
- Сглаживание удаляет детали.
- Плюсы и минусы:
  – Минус: нельзя удалить шум без размывания форм объектов (границ).
  – Плюс: выделяется крупномасштабная структура изображения.

Гауссово усреднение

- Веса ближних пикселов больше весов дальних.
- Ядро симметрично относительно начала координат.
- Ядро пропорционально функции

$$g(x, y) = \exp \; - \; \frac{x^2 + y^2}{2\sigma^2}$$



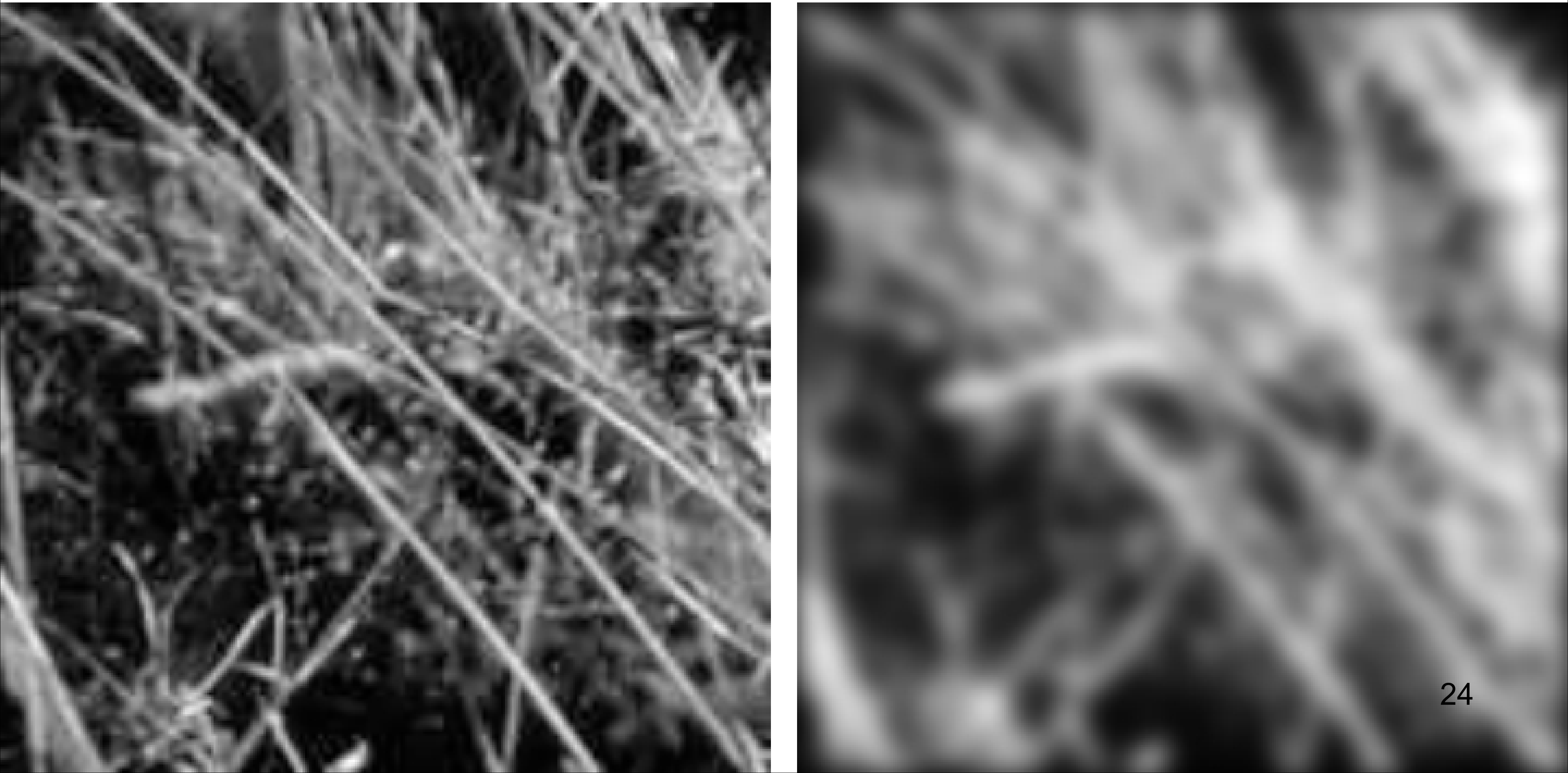


Примеры масок гауссовых фильтров

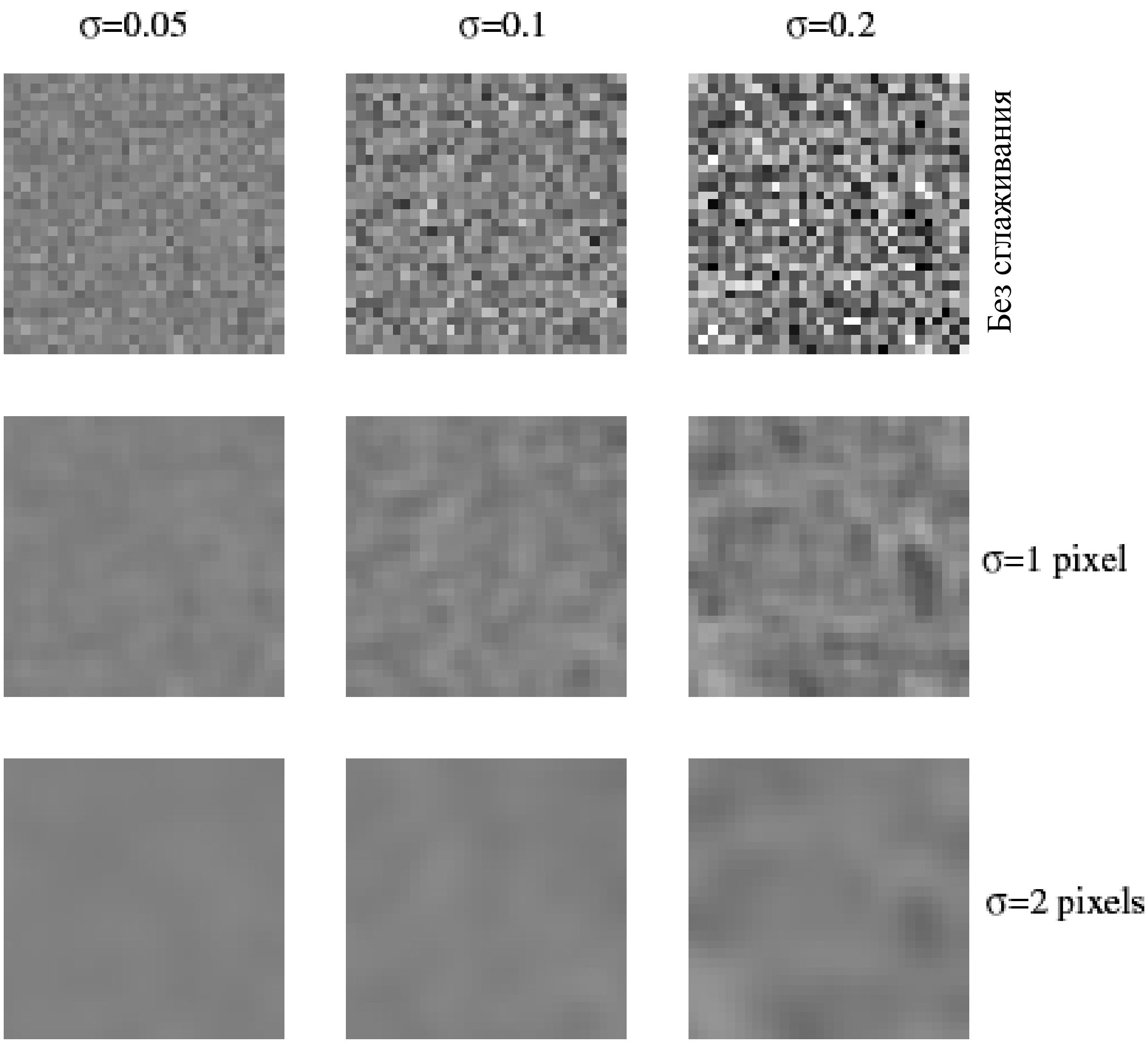
$$g_{3x3} = 1/16 \cdot \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

$$g_{7x7} = 1/1098 \cdot \begin{array}{|c|c|c|c|c|c|c|}\hline 1 & 3 & 7 & 9 & 7 & 3 & 1 \\ \hline 3 & 12 & 26 & 33 & 26 & 12 & 3 \\ \hline 7 & 26 & 55 & 70 & 55 & 26 & 7 \\ \hline 9 & 33 & 70 & 90 & 70 & 33 & 9 \\ \hline 7 & 26 & 55 & 70 & 55 & 26 & 7 \\ \hline 3 & 12 & 26 & 33 & 26 & 12 & 3 \\ \hline 1 & 3 & 7 & 9 & 7 & 3 & 1 \\ \hline \end{array}$$

Пример: сглаживание гауссовым усреднением



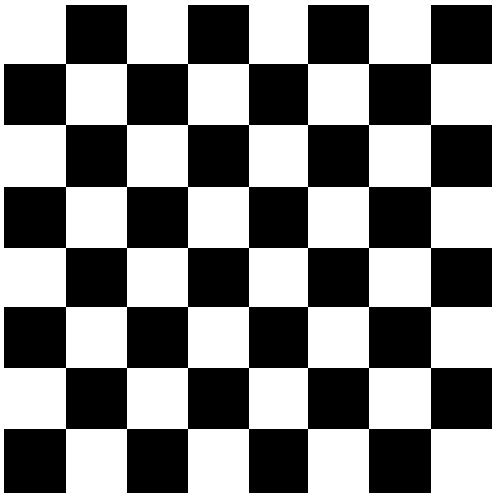
Эффекты сглаживания

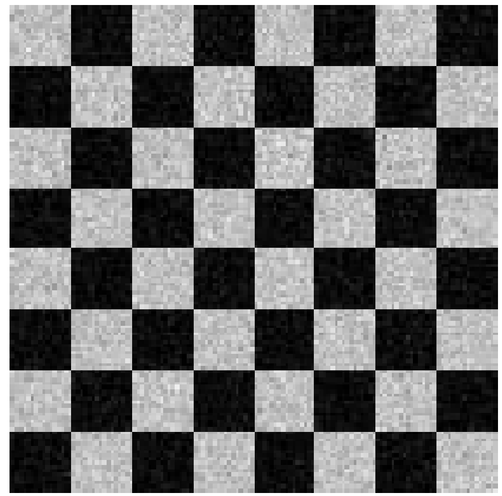


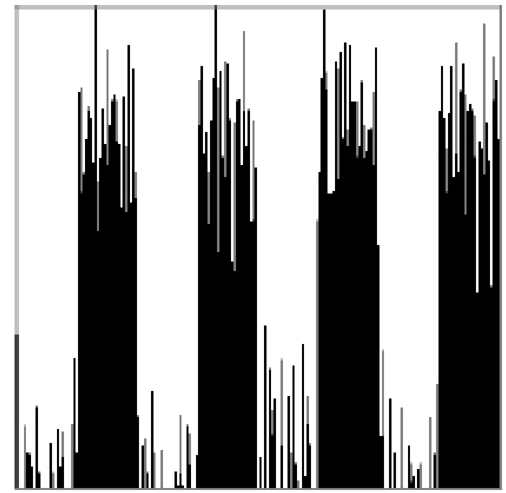
Изображения в каждой строке сглажены гауссовым фильтром с одним и тем же размером ядра.

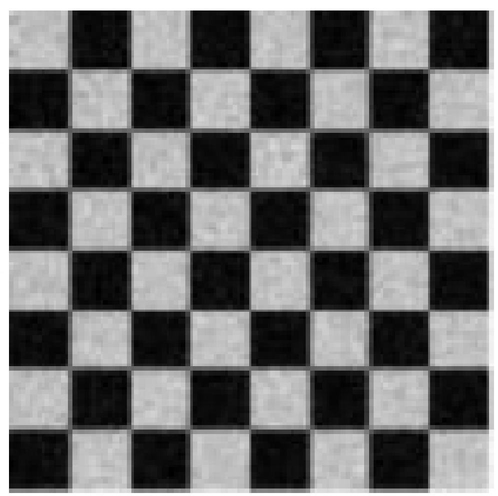
Изображения в каждом столбце являются реализациями гауссова шума с одной и той же дисперсией

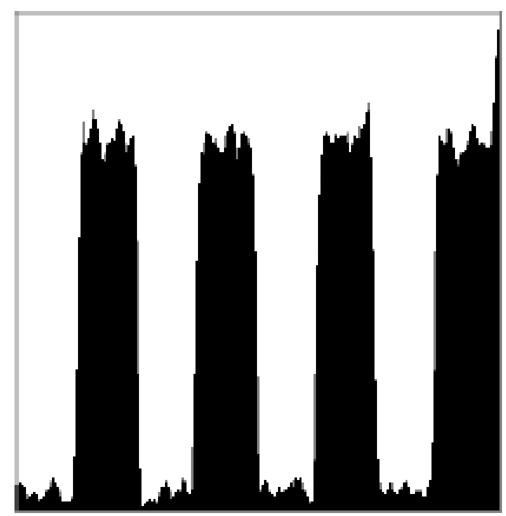
Изображение с гауссовым шумом











Пример: подавление шума гауссовым сглаживанием

Эффективная реализация линейных фильтров

- Большинство фильтров являются сепарабельными, в том числе box–фильтр и гауссов фильтр:

$$\begin{matrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{matrix} = \begin{matrix} 1/3 \\ 1/3 \\ 1/3 \end{matrix} \begin{pmatrix} 1/3 & 1/3 & 1/3 \end{pmatrix} \qquad \begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix} = \begin{matrix} 1 \\ 2 \\ 1 \end{matrix} \begin{pmatrix} 1 & 2 & 1 \end{pmatrix}$$

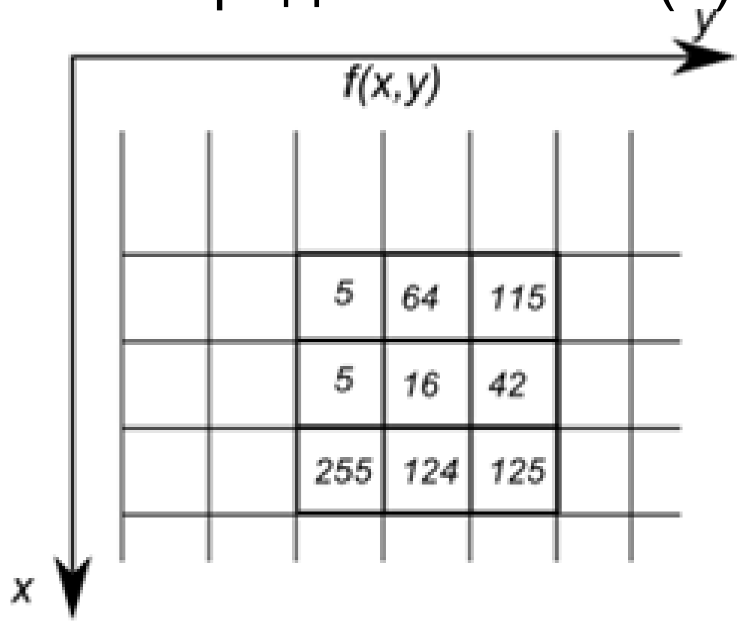
- Таким образом, фильтрация может быть осуществлена в два этапа:
  - Одномерная фильтрация столбцов;
  - Одномерная фильтрация строк.

Медианный фильтр

Из исходного изображения выделим область размером 3x3 с 9 значениями яркостей пикселов. Получим упорядоченную по возрастанию последовательность

a= [5, 5, 16, 42, 64, 115, 124, 125, 255],

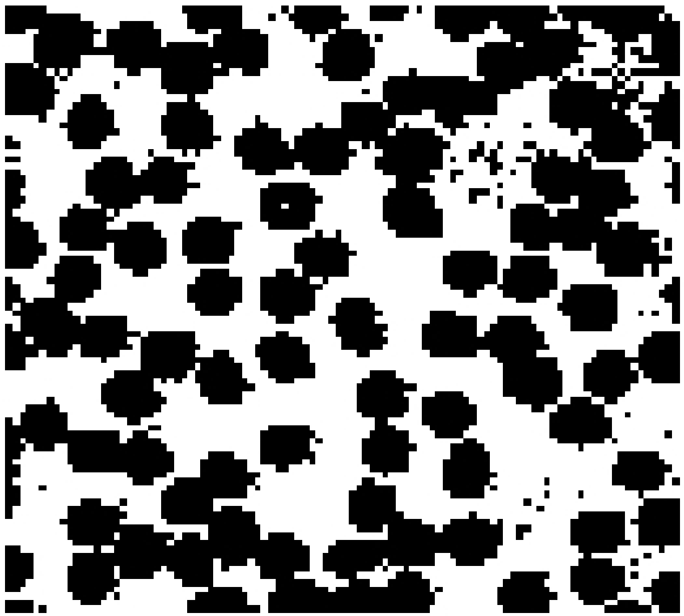
выберем из данной последовательности элемент, расположенный по середине median(a)=64.

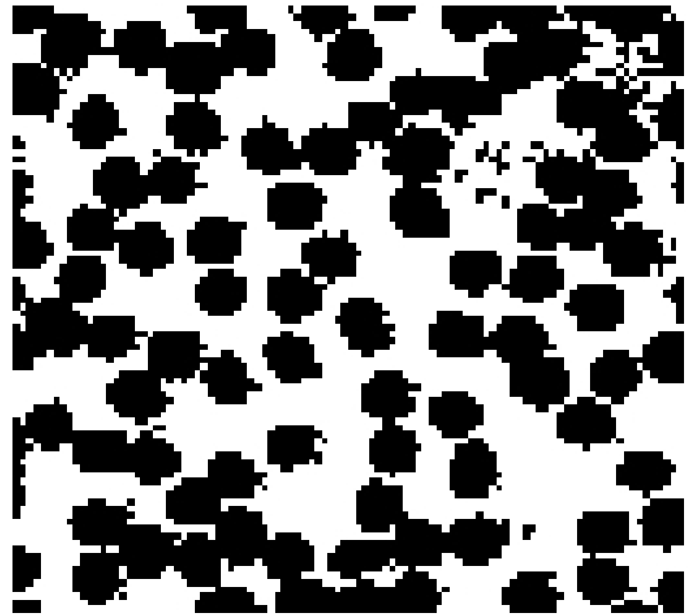


Медианный фильтр

- Усредняющий фильтр обычно работает, если области изображения более или менее однородны.
- **Медианный фильтр** заменяет центральный пиксел в маске медианой пикселов из некоторой окрестности.
  - Работает для многих типов шума.
  - Сохраняет границы в отличие от усредняющего фильтра.
  - Вычислительно сложен (из-за сортировки).
- Медианный фильтр – нелинейный!

Медианный фильтр





1	1	1
1	0	1
1	1	1

⇒

1	1	1
1	1	1
1	1	1

;

0	0	0
0	1	0
0	0	0

⇒

0	0	0
0	0	0
0	0	0

X	X	X
X	L	X
X	X	X

⇒

X	X	X
X	X	X
X	X	X

;

	X	
X	L	X
	X	

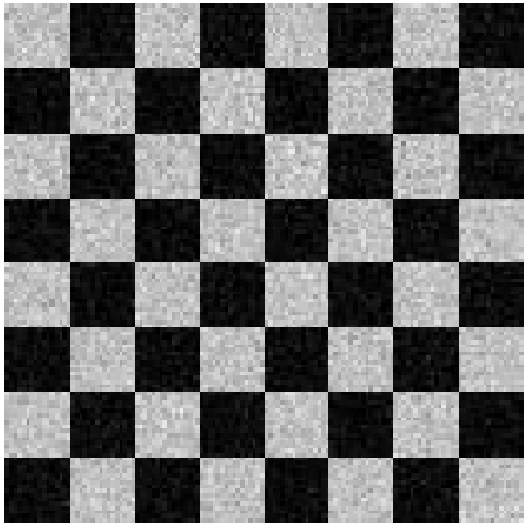
⇒

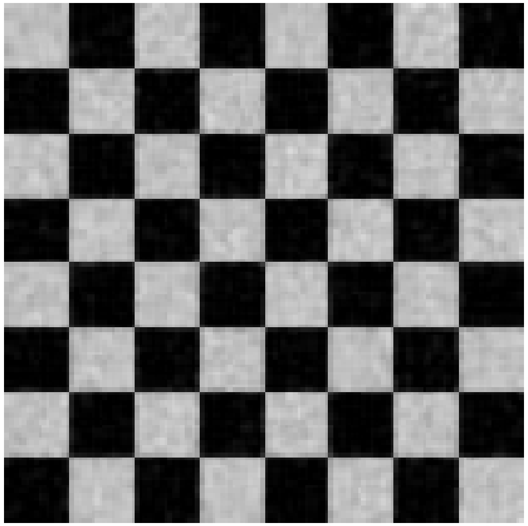
	X	
X	X	X
	X	

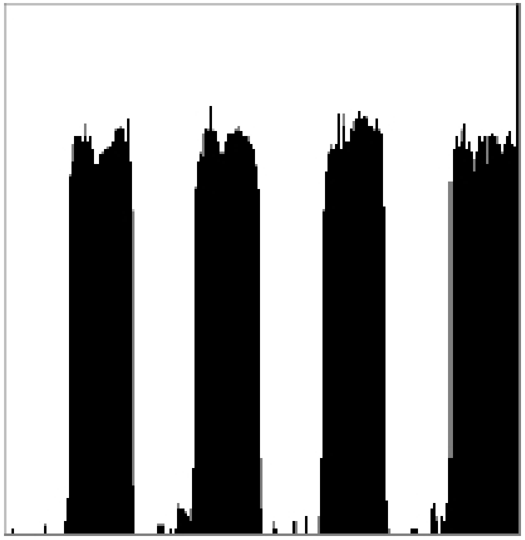
Пример: устранение импульсного шума медианной фильтрацией

Пример: устранение гауссова шума медианной фильтрацией

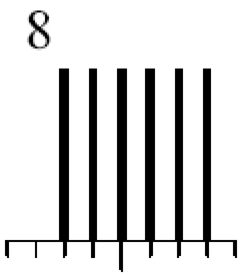
Использован медианный фильтр с маской 5x5



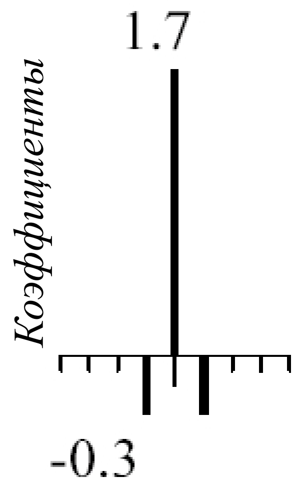




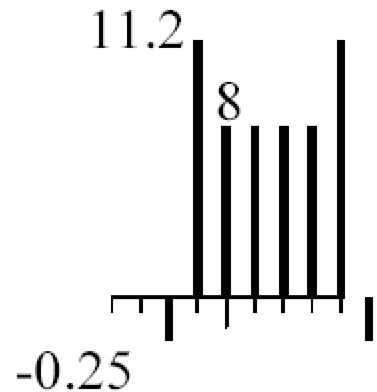
Пример действия фильтра увеличения резкости



Исходное изображение

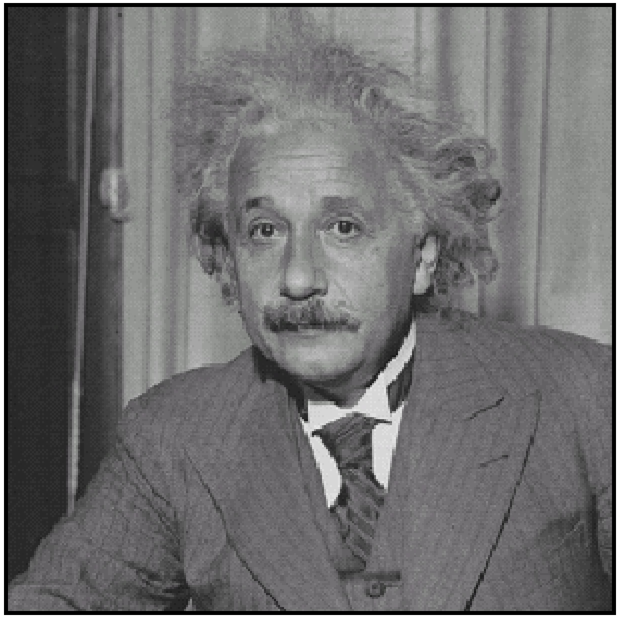


Ядро фильтра

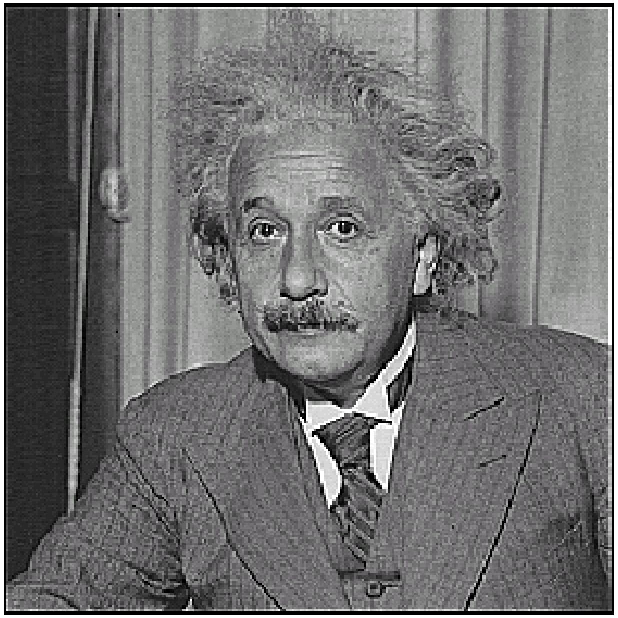


Результирующее изображение (перепады подчеркнуты; области постоянной интенсивности неизменны)

Пример действия фильтра увеличения резкости



Исходное изображение



Результирующее изображение

Линейные фильтры, подчеркивающие линии определенного направления

$$\frac{1}{16} \cdot \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Подчеркиваются четырехсвязные элементы исходного изображения, т.е. горизонтальные и вертикальные линии.

$$\frac{1}{16} \cdot \begin{array}{|c|c|c|}\hline 2 & 1 & 2 \\ \hline 1 & 4 & 1 \\ \hline 2 & 1 & 2 \\ \hline \end{array}$$

Подчеркиваются восьмисвязные элементы, не являющиеся четырехсвязными элементами исходного изображения.

Линейные фильтры, подчеркивающие контуры

Общий вид (A ≥ 1)

0	–1	0
–1	A+4	–1
0	–1	0

–1	–1	–1
–1	A+8	–1
–1	–1	–1

Примеры

0	–1	0
–1	5	–1
0	–1	0

–1	–1	–1
–1	9	–1
–1	–1	–1

- Подчеркиваются контуры независимо от их направления.
- Сумма весов фильтра равна **единице**.

Линейные фильтры, выделяющие контуры

0	–1	0
–1	4	–1
0	–1	0

–1	–1	–1
–1	8	–1
–1	–1	–1

- Выделяются контуры независимо от их направления.
- Сумма весов фильтра равна **нулю**.
- Фильтры выделения контуров – дискретные **дифференциальные операторы.**

Линейные фильтры, выделяющие контуры

Методы выделения контуров 1 порядка

1. Вычислить градиент изображения
2. Выполнить подавление немаксимальных пикселов
3. Произвести пороговую бинаризацию

Методы выделения контуров 2 порядка

- Найти пикселы изображения, в которых вторая производная изображения (лапласиан) обращается в нуль.

Линейные фильтры, выделяющие контуры

Методы выделения контуров 1 порядка

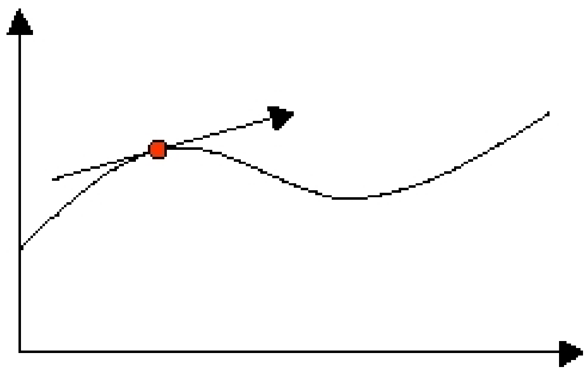
1. Вычислить градиент изображения
2. Выполнить подавление немаксимальных пикселов
3. Произвести пороговую бинаризацию

Методы выделения контуров 2 порядка

- Найти пикселы изображения, в которых вторая производная изображения (лапласиан) обращается в нуль.

Градиент

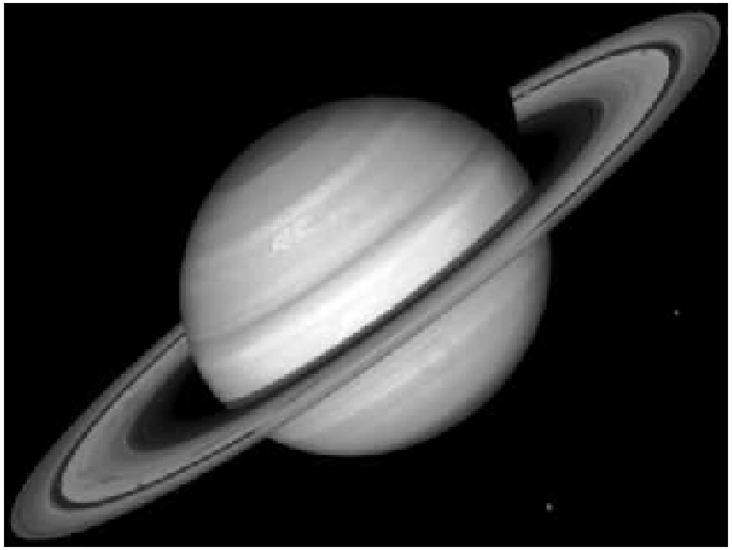
- Градиент несет информацию об изменении (о скорости изменения) функции
- В случае функции одной переменной градиент – тангенс угла наклона касательной.

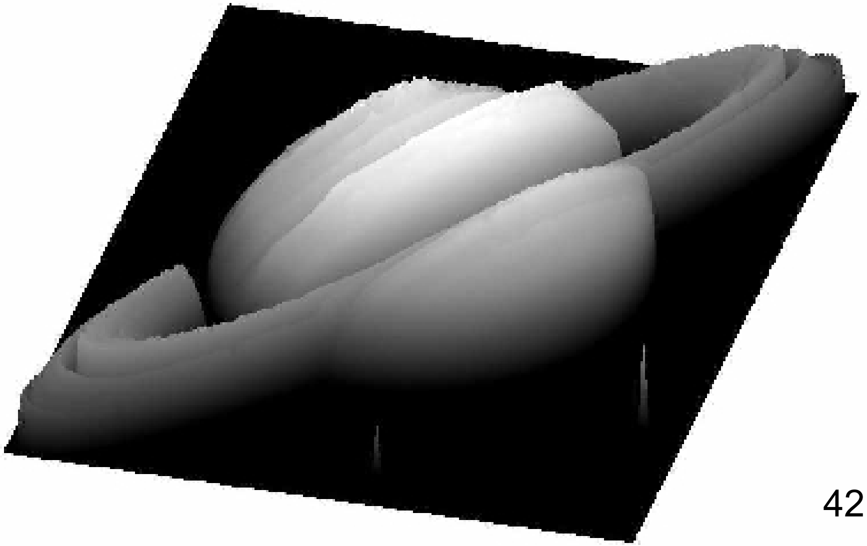


- Что такое градиент изображения?

Градиент изображения

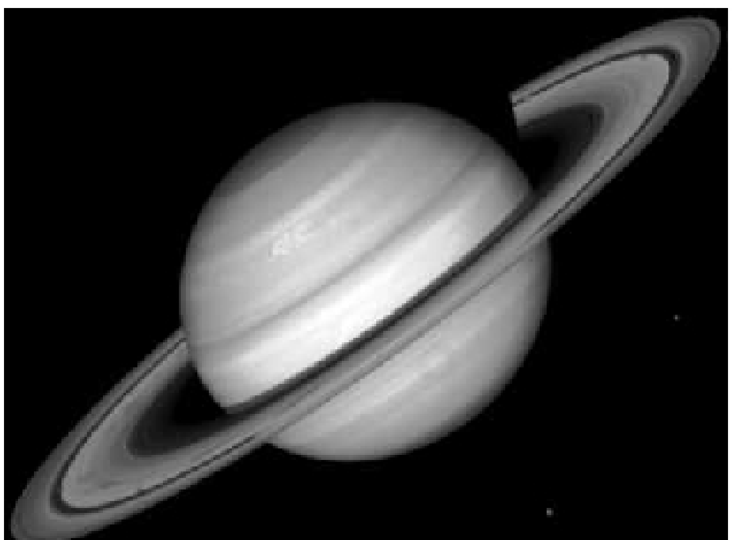
- Изображение – функция двух переменных (трехмерная поверхность)
- Интенсивность пикселов – значение функции («высота», z–координата)

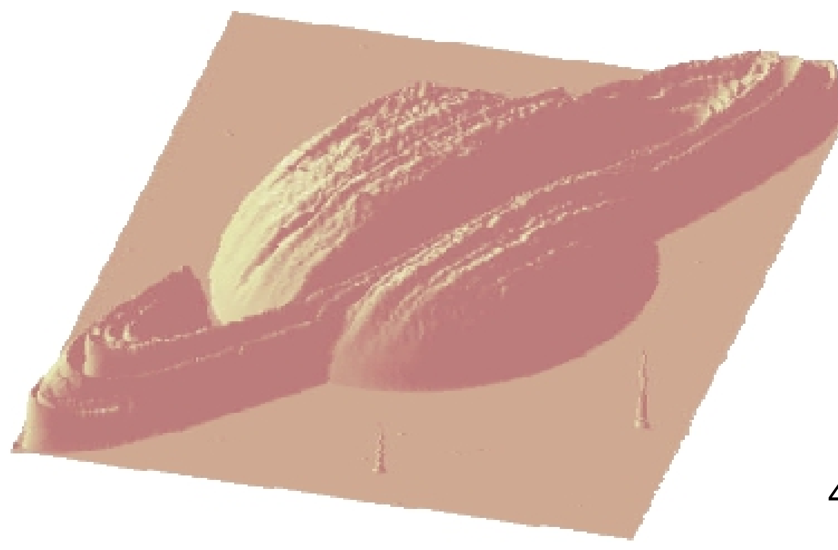




Градиент изображения

- Градиент в точке p – вектор, указывающий направление максимального роста функции от темных значений к светлым
- Градиент изображения ∇I – *векторное поле,* где каждый элемент $\nabla I(p)$ – вектор направления к более светлым значениям





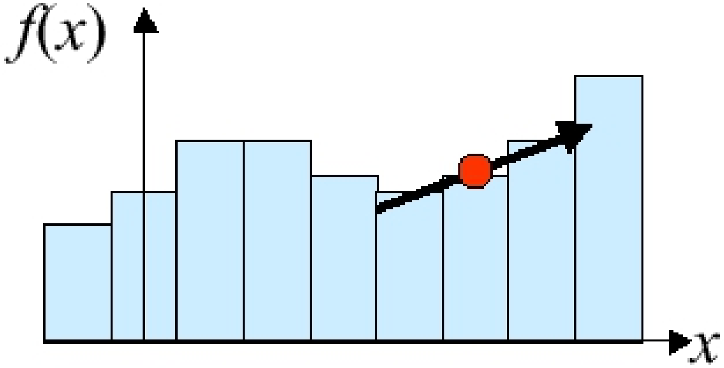
Компоненты градиента

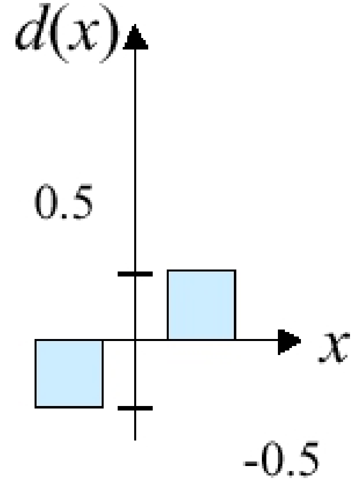
- Компоненты вектора градиента $\nabla \boldsymbol{I}$ – частные производные в горизонатльном (по переменной x) и вертикальном (по переменной y) направлениях:

$$I(x, y) = {\frac{\partial I}{\partial x} \quad \frac{\partial I}{\partial y}}^{T}$$

- В качестве базисных направлений могут быть выбраны произвольные, однако удобнее выбирать их совпадающими с направлениями x и y

Численное дифференцирование





$$\mathbf{d} = \begin{bmatrix} -0.5 \\ 0 \\ 0.5 \end{bmatrix}$$

Ядро одномерного дифференциального фильтра

- Производную дискретной функции одной переменной можно численно оценить следующим образом:

$$\frac{df}{dx} \approx \frac{f(x+1) - f(x-1)}{2}$$

- Это равносильно свертке функции $f(x)$ с функцией $d(x)$ или действию дифференциального фильтра с ядром $\boldsymbol{d}$.

Дифференцирование изображения

- Производная изображения вычисляется путем фильтрации изображения дифференциальным фильтром.
- Один из возможных дифференциальных фильтров – фильтр Превитта 3x3:

-1	0	1
-1	0	1
-1	0	1

K

Пример дифференцирования изображения

- Как это работает?

-1	0	1
-1	0	1
-1	0	1

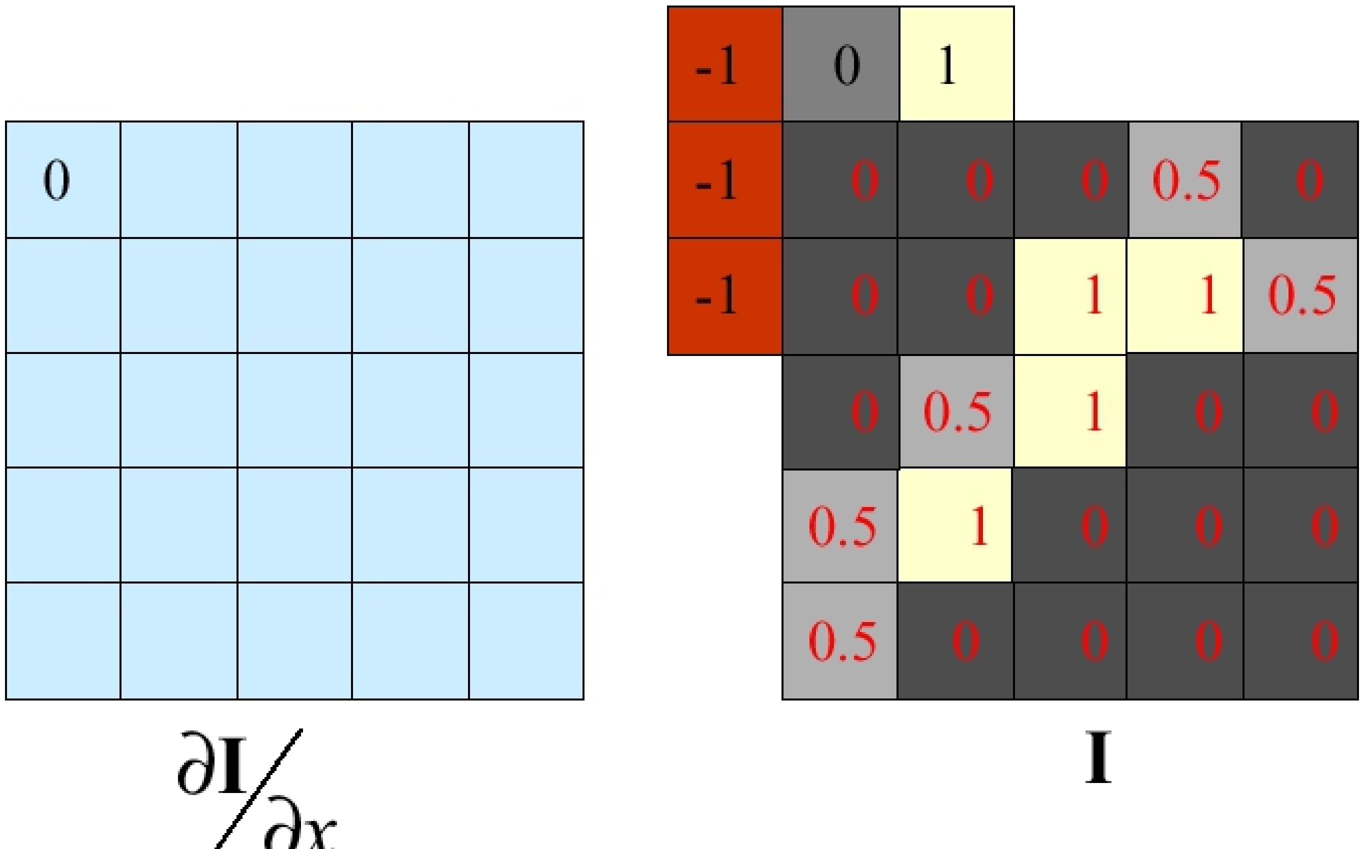
K

0	0	0	0.5	0
0	0	1	1	0.5
0	0.5	1	0	0
0.5	1	0	0	0
0.5	0	0	0	0

I

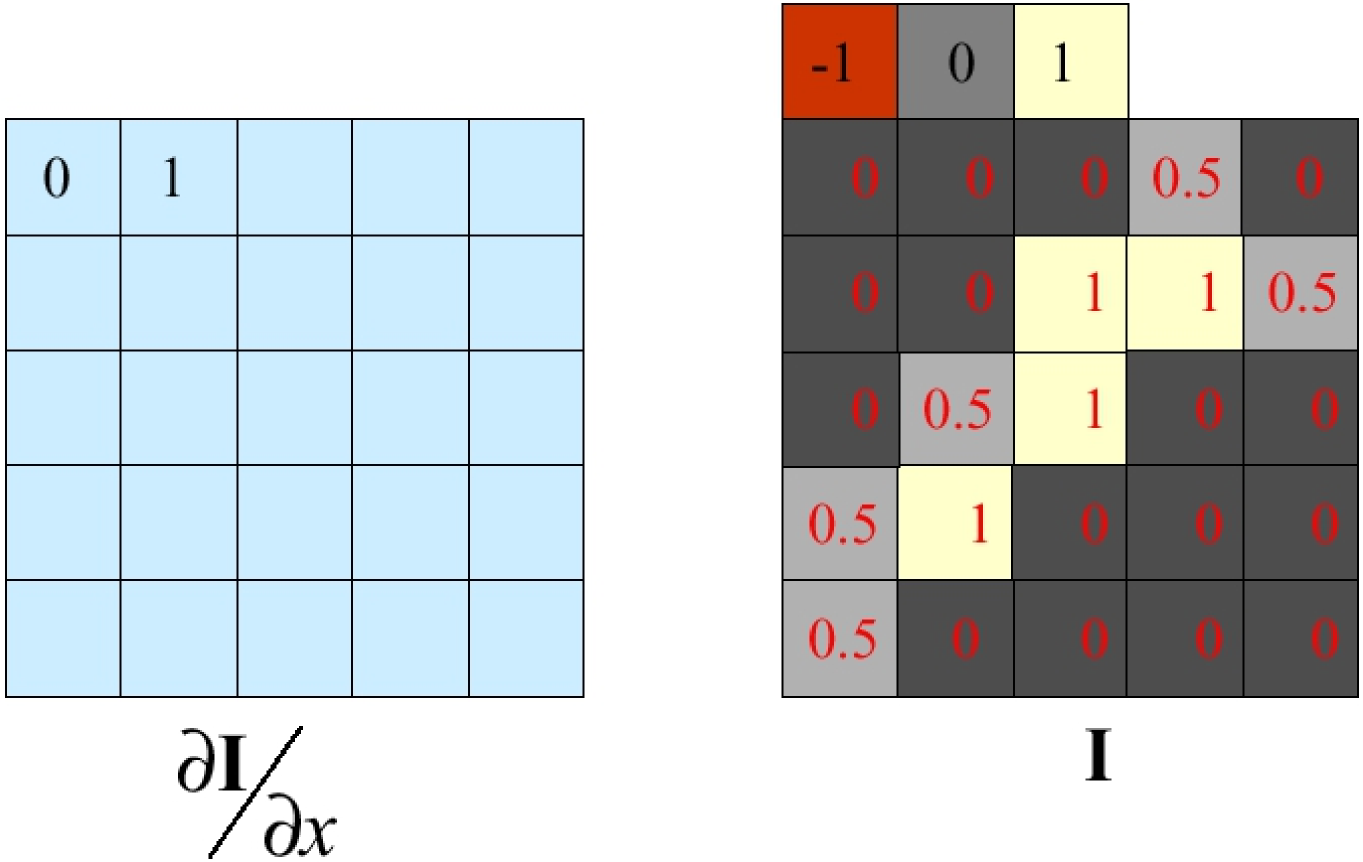
Пример дифференцирования изображения

- Хотим получить результат тех же размеров, что и изображение



Пример дифференцирования изображения

- Хотим получить результат тех же размеров, что и изображение



Пример дифференцирования изображения

- Хотим получить результат тех же размеров, что и изображение

-1	0	1

0	1	1.5		

$\partial \mathbf{I} / \partial x$

0	0	0	0.5	0
0	0	1	1	0.5
0	0.5	1	0	0
0.5	1	0	0	0
0.5	0	0	0	0

$\mathbf{I}$

Пример дифференцирования изображения

- Хотим получить результат тех же размеров, что и изображение

0	1	1.5	-0.5	

$\partial \mathbf{I} / \partial x$

-1	0	1

0	0	0	0.5	0
0	0	1	1	0.5
0	0.5	1	0	0
0.5	1	0	0	0
0.5	0	0	0	0

$\mathbf{I}$

Пример дифференцирования изображения

- Хотим получить результат тех же размеров, что и изображение

0	1	1.5	-0.5	-1.5

$\partial \mathbf{I} / \partial x$

			-1	0	1
0	0	0	0.5	0	1
0	0	1	1	0.5	1
0	0.5	1	0	0	
0.5	1	0	0	0	
0.5	0	0	0	0	

$\mathbf{I}$

Пример дифференцирования изображения

- Хотим получить результат тех же размеров, что и изображение

0	1	1.5	-0.5	-1.5
0.5				

$\partial \mathbf{I} / \partial x$

-1	0	0	0	0.5	0
-1	0	0	1	1	0.5
-1	0	0.5	1	0	0
	0.5	1	0	0	0
	0.5	0	0	0	0

$\mathbf{I}$

Пример дифференцирования изображения

- Хотим получить результат тех же размеров, что и изображение

0	1	1.5	-0.5	-1.5
0.5	2			

$\partial \mathbf{I} / \partial x$

0	0	0	0.5	0
0	0	1	1	0.5
0	0.5	1	0	0
0.5	1	0	0	0
0.5	0	0	0	0

$\mathbf{I}$

Пример дифференцирования изображения

- Хотим получить результат тех же размеров, что и изображение

0	1	1.5	-0.5	-1.5
0.5	2	1		

$\partial \mathbf{I} / \partial x$

0	0	0	0.5	0
0	0	1	1	0.5
0	0.5	1	0	0
0.5	1	0	0	0
0.5	0	0	0	0

$\mathbf{I}$

Пример дифференцирования изображения

- Фильтрация завершена
- Вообще говоря, значения отклика могут не укладываться в диапазон квантования.

0	1	1.5	-0.5	-1.5
0.5	2	1	-1.5	-1.5
1.5	1.5	-0.5	-1.5	-1
1.5	0	-1.5	-1	0
1	-1	-1	0	0

$$\partial \mathbf{I} / \partial x$$

0	0	0	0.5	0
0	0	1	1	0.5
0	0.5	1	0	0
0.5	1	0	0	0
0.5	0	0	0	0

$$\mathbf{I}$$

Пример дифференцирования изображения

- Представляем результат как изображение

0	1	1.5	-0.5	-1.5
0.5	2	1	-1.5	-1.5
1.5	1.5	-0.5	-1.5	-1
1.5	0	-1.5	-1	0
1	-1	-1	0	0

$\partial \mathbf{I} / \partial x$

0	0	0	0.5	0
0	0	1	1	0.5
0	0.5	1	0	0
0.5	1	0	0	0
0.5	0	0	0	0

$\mathbf{I}$

Пример дифференцирования изображения

- Вычислили

$$\partial I / \partial x = I \quad K$$

- Используем транспонированное ядро фильтра для вычисления

$$\partial I / \partial y = I \quad K^T$$

K

-1	0	1
-1	0	1
-1	0	1

K^{T}

1	1	1
0	0	0
-1	-1	-1

Пример дифференцирования изображения

- Используем ту же процедуру, что и при вычислении производной по x

0	1	2	2.5	1.5
0.5	1.5	1	0.5	-0.5
1.5	0.5	-1	-2.5	-1.5
0	-1	-1.5	-1	0
-1.5	-1.5	-1	0	0

$\partial \mathbf{I} / \partial y$

0	0	0	0.5	0
0	0	1	1	0.5
0	0.5	1	0	0
0.5	1	0	0	0
0.5	0	0	0	0

$\mathbf{I}$

Пример дифференцирования изображения

- Используем ту же процедуру, что и при вычислении производной по x

0	1	2	2.5	1.5
0.5	1.5	1	0.5	-0.5
1.5	0.5	-1		-1.5
0	-1	-1.5	-1	0
-1.5	-1.5	-1	0	0

$\partial \mathbf{I} / \partial y$

0	0	0	0.5	0
0	0	1	1	0.5
0	0.5	1	0	0
0.5	1	0	0	0
0.5	0	0	0	0

$\mathbf{I}$

Пример дифференцирования изображения

- Таким образом, мы вычислили градиент изображения $\boldsymbol{I}$ (используя $\boldsymbol{K}$ — оператор Превитта с размерами 3x3)

$$I = \left[\frac{\partial I}{\partial x} \; \frac{\partial I}{\partial y} \right]^T$$

Дифференциальные фильтры

- Существует много других дифференциальных фильтров. Наиболее известны из них:

Фильтр Превитта

–1	0	1
–1	0	1
–1	0	1

Фильтр Собеля

–1	0	1
–2	0	2
–1	0	1

Фильтр Робертса

0	1
–1	0

1	0
0	–1

y–составляющая не может быть получена транспонированием!

- Используются маски бóльших размеров
- Компоненты градиента могут быть измерены в двух любых ортогональных направлениях. Фильтр Робертса использует диагональные направления.

Свойства дифференцирующих фильтров

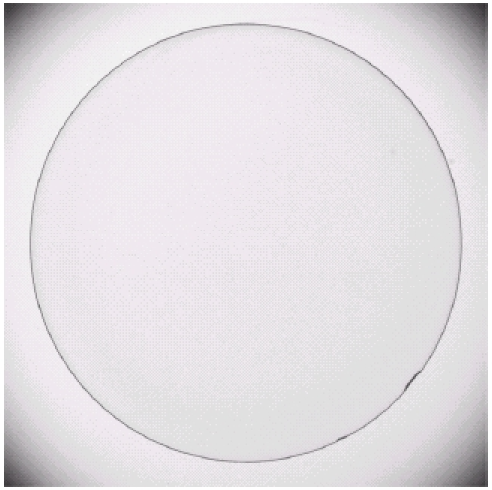
- Дифференцирующие фильтры содержат отрицательные элементы.
- Сумма весов дифференцирующего фильтра равна нулю, т.е. в областях постоянной интенсивности отклик фильтра равен нулю.
- Фильтры, вычисляющие первую производную, в точках больших перепадов интенсивности порождают большие по абсолютной величине значения откликов.
- Фильтры, вычисляющие вторую производную, в точках перепада интенсивности порождают нулевой отклик.

Замечания по использованию дифференциальных фильтров

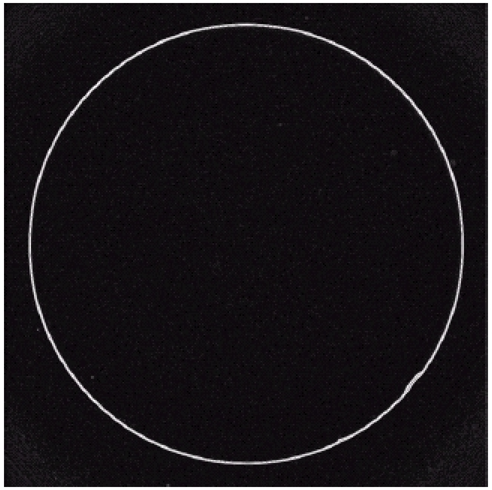
- Фильтр Робертса вычислительно более эффективен, благодаря маленькому количеству задействуемых соседей.
- Все локальные дифференциальные операторы «теряют» многие из границ.
- Вычисление квадратов и квадратных корней вычислительно слишком дорого. Можно заменять их вычислением абсолютных значений, максимумов и т.п.

Пример действия фильтра Собеля

Изображение контактной линзы с дефектами на краю (в районе 4–5 часов)



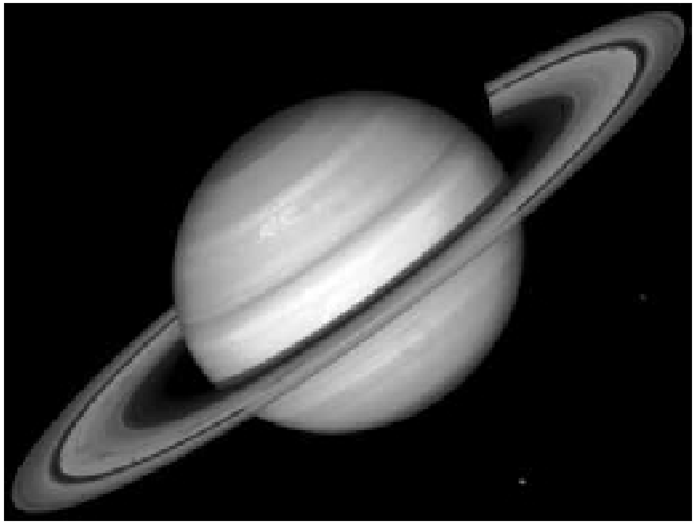
Результат действия фильтра Собеля



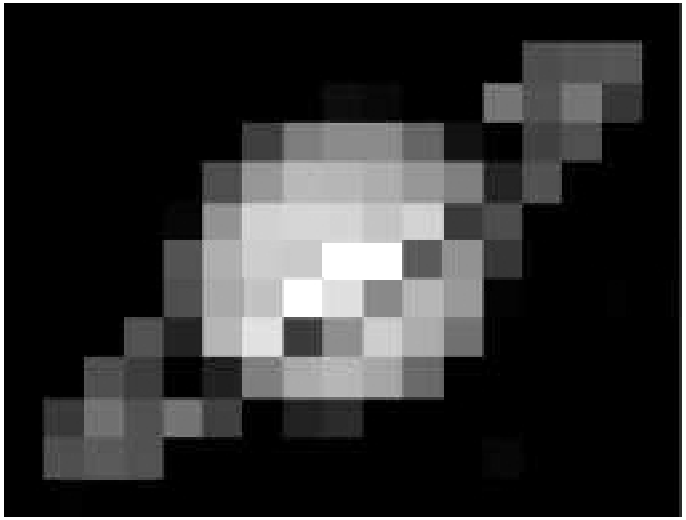
Еще о градиенте

- Рассмотрим изображение Сатурна передискретизированное с более крупным шагом

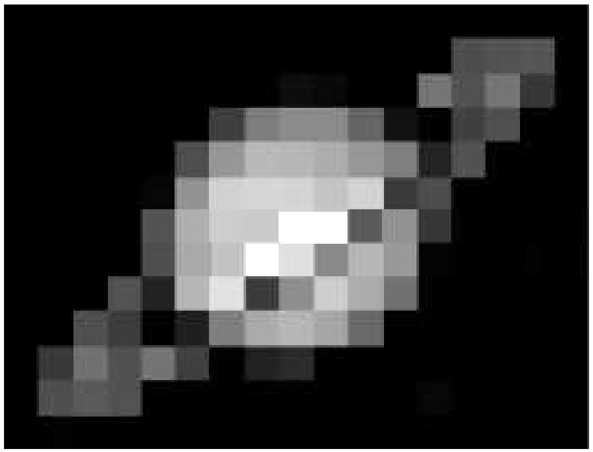
Исходное изображение

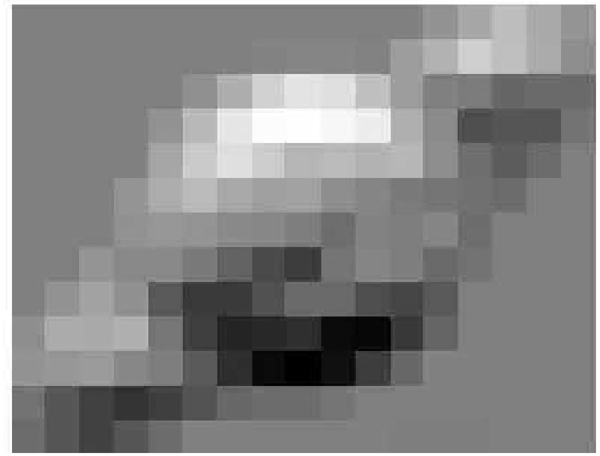


Передискретизированное изображение



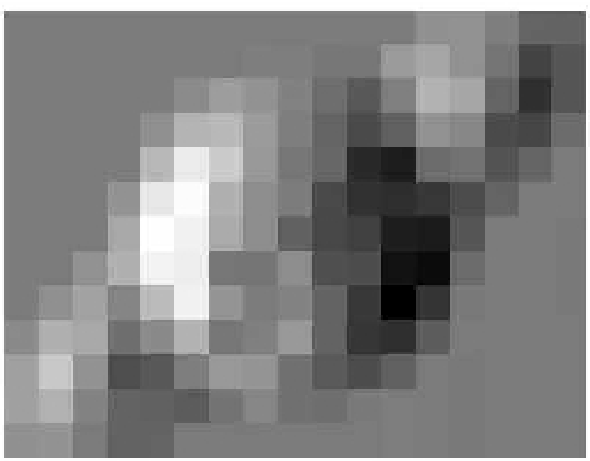
Еще о градиенте





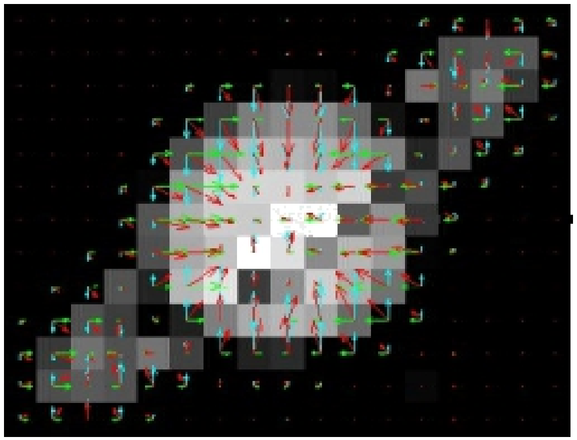
$\partial \mathbf{I} / \partial y$

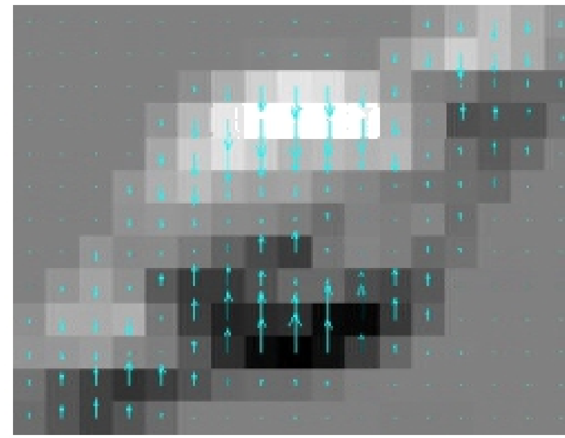
$\partial \mathbf{I} / \partial x$



- Определим градиент в x–направлении
- Определим градиент в y–направлении

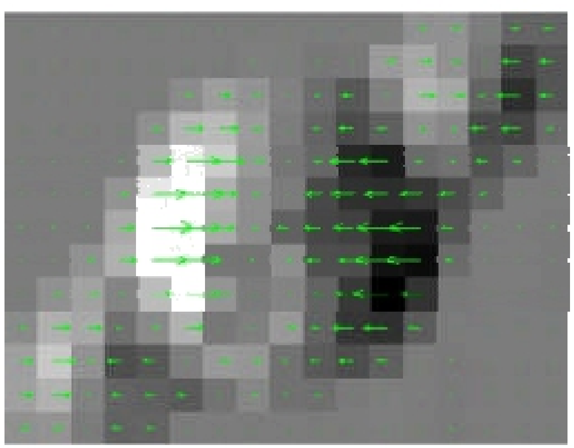
Еще о градиенте





$\partial \mathbf{I} / \partial y$

$\partial \mathbf{I} / \partial x$



- Каждая из компонент градиента может быть рассмотрена как вектор в соответствующем направлении
- В совокупности они определяют градиент (показан красным)

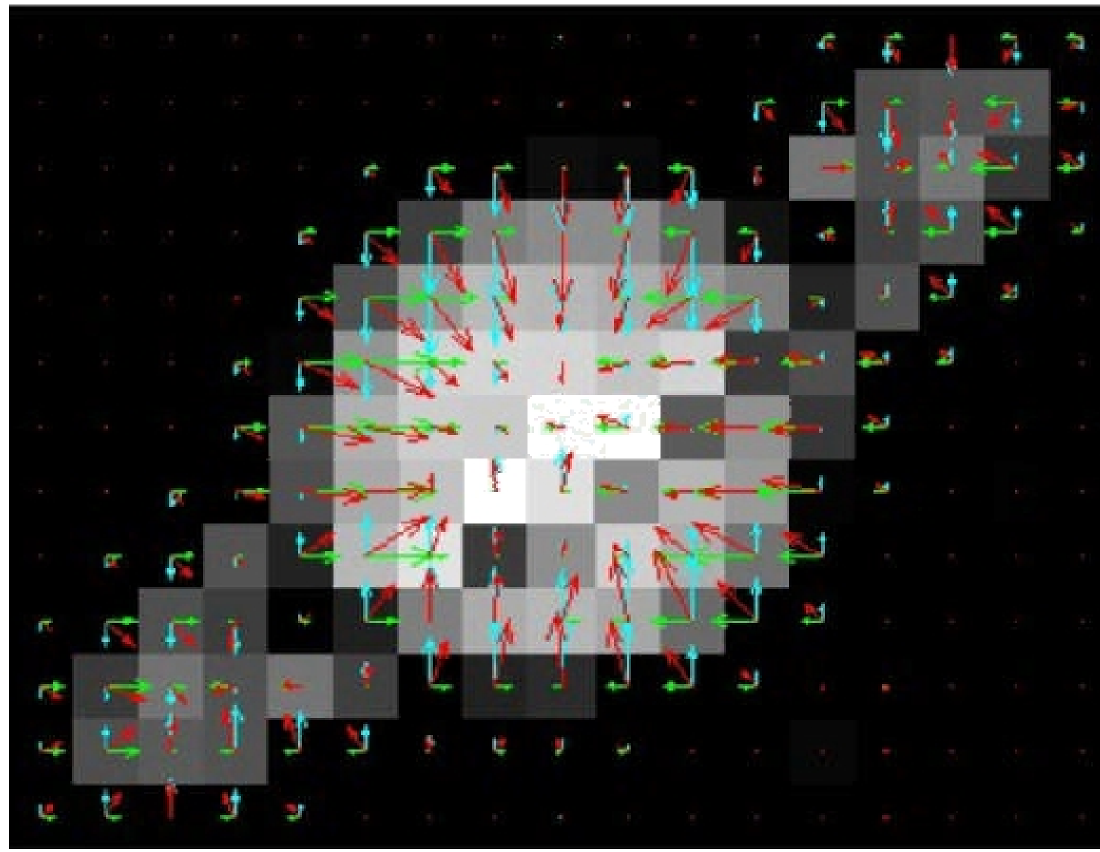
Еще о градиенте

- Каждый элемент векторного поля $\nabla \boldsymbol{I}$ имеет:
  - Модуль

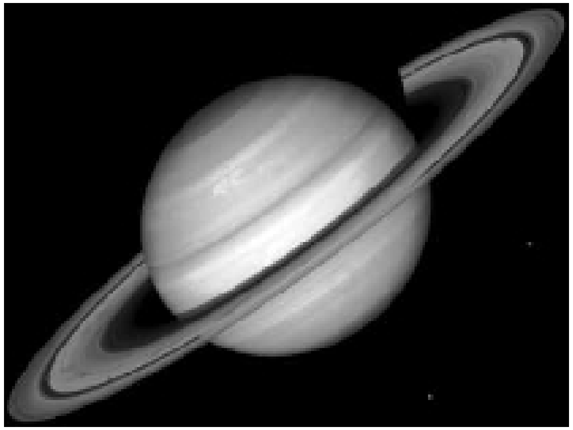
$$\| \quad I(p) \|$$

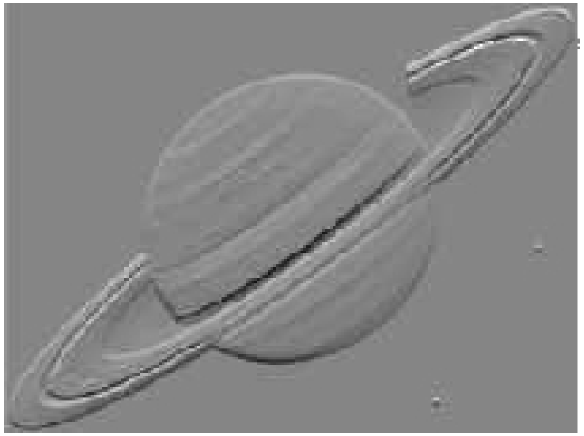
  - Ориентацию

$$I(p)$$

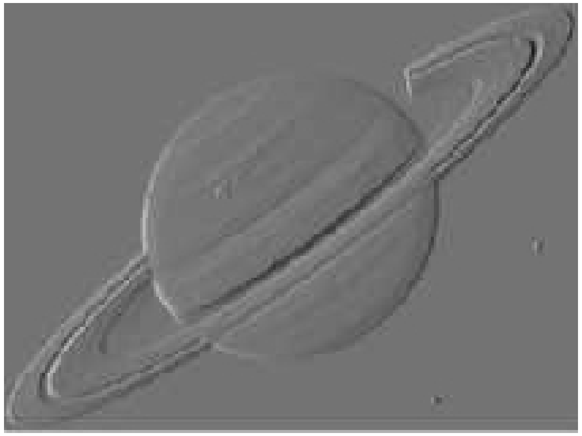


Градиент исходного изображения





$$\partial \mathbf{I} / \partial y$$

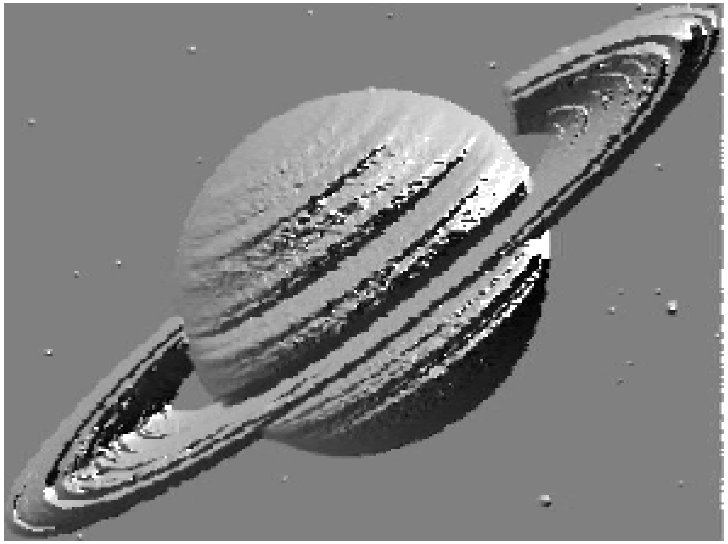


$$\partial \mathbf{I} / \partial x$$

Модуль и ориентация градиента

$$\nabla \mathbf{I} = \left({\partial \mathbf{I}}/{\partial x}, {\partial \mathbf{I}}/{\partial y} \right)^{\mathrm{T}}$$

$\|\nabla \mathbf{I}\|$



$\angle \nabla \mathbf{I}$

Модуль и ориентация градиента

$$\nabla \mathbf{I} = \left(\partial \mathbf{I} / \partial x, \partial \mathbf{I} / \partial y \right)^{\mathrm{T}}$$

$\|\nabla \mathbf{I}\|$

$\angle \nabla \mathbf{I}$

Модуль градиента и выделение контуров

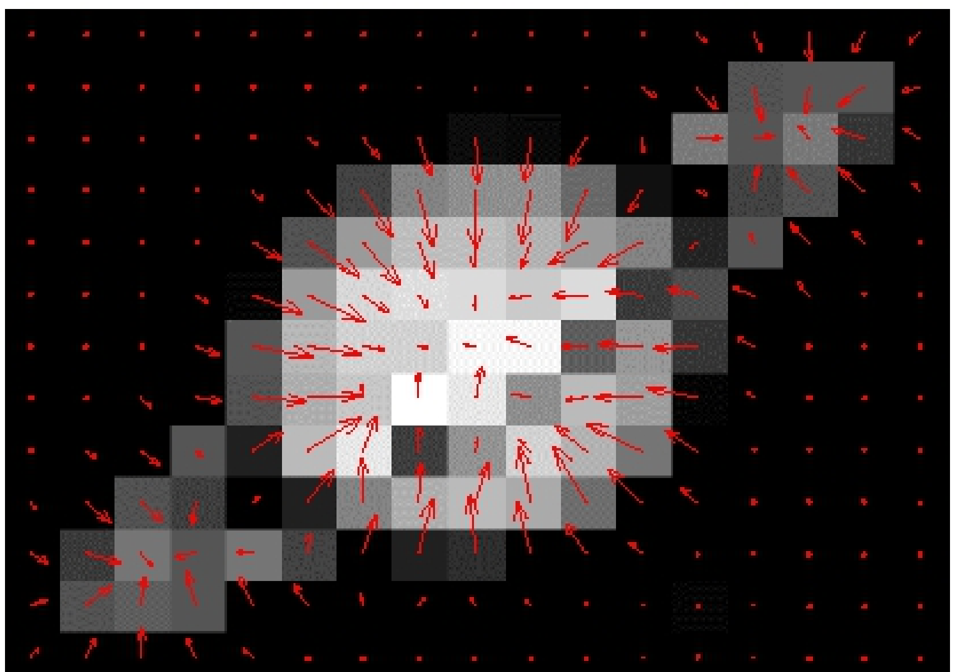
$\|\nabla \mathbf{I}\|$

- Модуль градиента служит хорошим индикатором местоположения контуров
- В чем состоят методы выделения контуров первого порядка:
  1. Вычислить градиент изображения Знаем!
  2. Выполнить подавление немаксимальных пикселов ?
  3. Пороговая бинаризация Легко!

Подавление немаксимальных пикселов

- Подавление немаксимальных пикселов состоит в обнулении тех пиков в $\| \quad I(p) \|$, которые не являются локальными максимумами в направлении $\quad I(p)$.

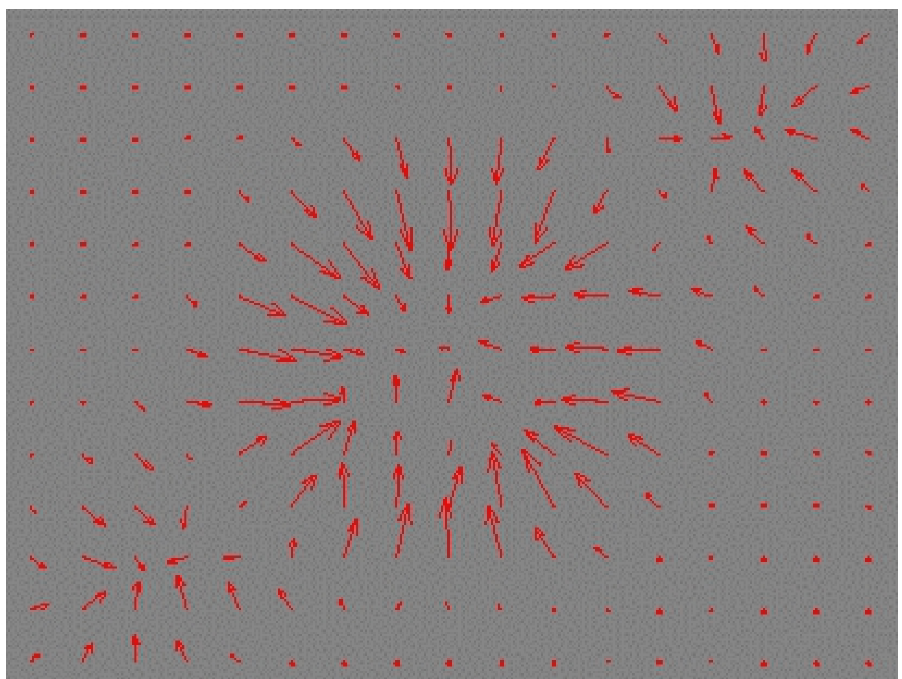
***I* и его градиентное поле**



Подавление немаксимальных пикселов

- Подавление немаксимальных пикселов состоит в обнулении тех пиков в $\| \quad I(p)\|$, которые не являются локальными максимумами в направлении $\quad I(p)$.

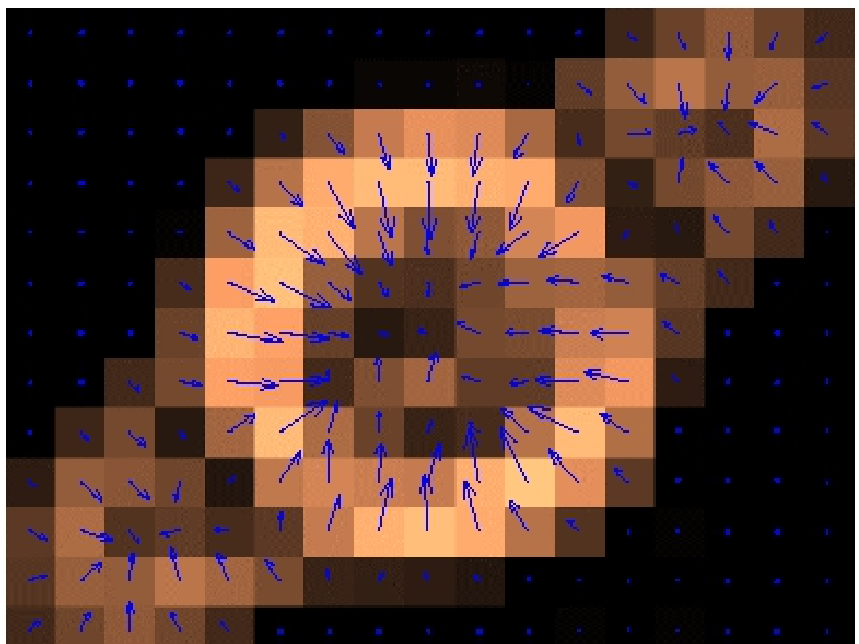
Градиентное поле ∇I



Подавление немаксимальных пикселов

- Подавление немаксимальных пикселов состоит в обнулении тех пиков в $\| \quad I(p)\|$, которые не являются локальными максимумами в направлении $\quad I(p)$.

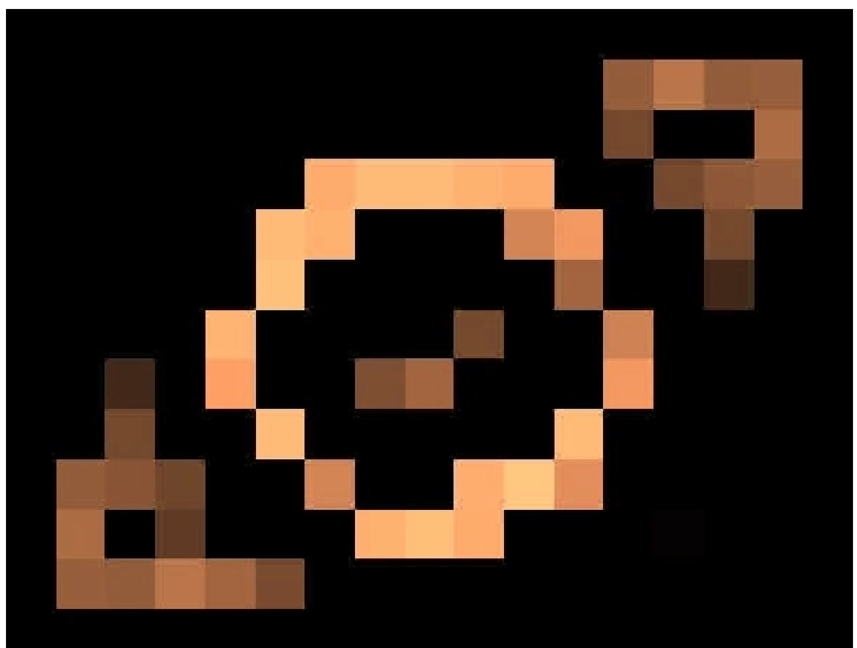
||∇*I*||



Подавление немаксимальных пикселов

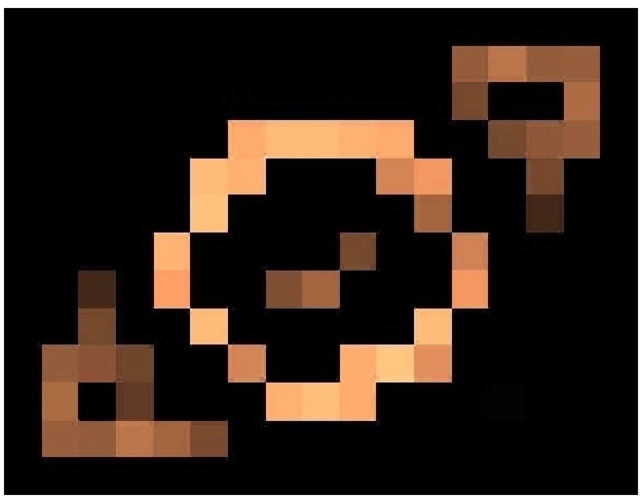
- Подавление немаксимальных пикселов состоит в обнулении тех пиков в $\| \; I(p)\|$, которые не являются локальными максимумами в направлении $\; I(p)$.

После подавления немаксимальных пикселов

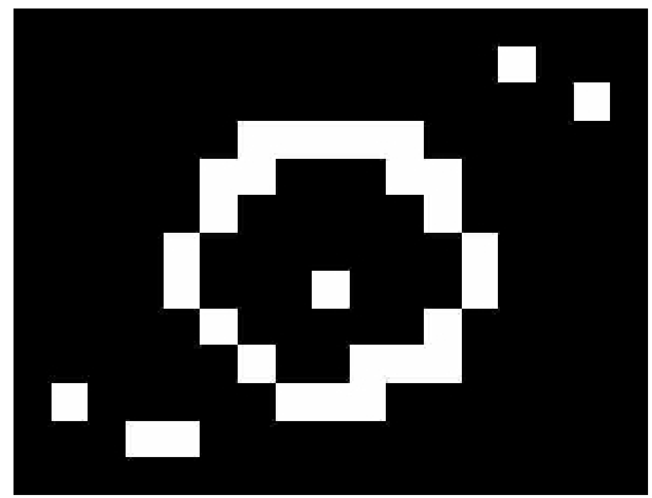


Пороговая бинаризация

- Бинаризация – последний шаг в методах выявления контуров 1 порядка.



После подавления немаксимальных пикселов



После пороговой бинаризации

Вычисление второй производной изображения

- Вторая производная есть производная первой производной.
- Для изображения I

$$\partial^2 I \big/ \partial x^2 = \partial \big/ \partial x \left(\partial I \big/ \partial x \right) = (I \quad K) \quad K = I \quad (K \quad K)$$

$$\partial^2 I \big/ \partial y^2 = \partial \big/ \partial y \quad \partial I \big/ \partial y \quad = (I \quad K^T) \quad K^T = I \quad (K^T \quad K^T)$$

- Ядро дифференциального фильтра второго порядка:

$$K \quad K$$

Дифференциальные операторы второго порядка

- На практике дифференциальные фильтры второго порядка используются редко.
- Исключение составляет оператор Лапласа (лапласиан):

$$
{}^{2}I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}
$$

Пример действия фильтра Лапласа

(a) Изображение северного полюса Луны

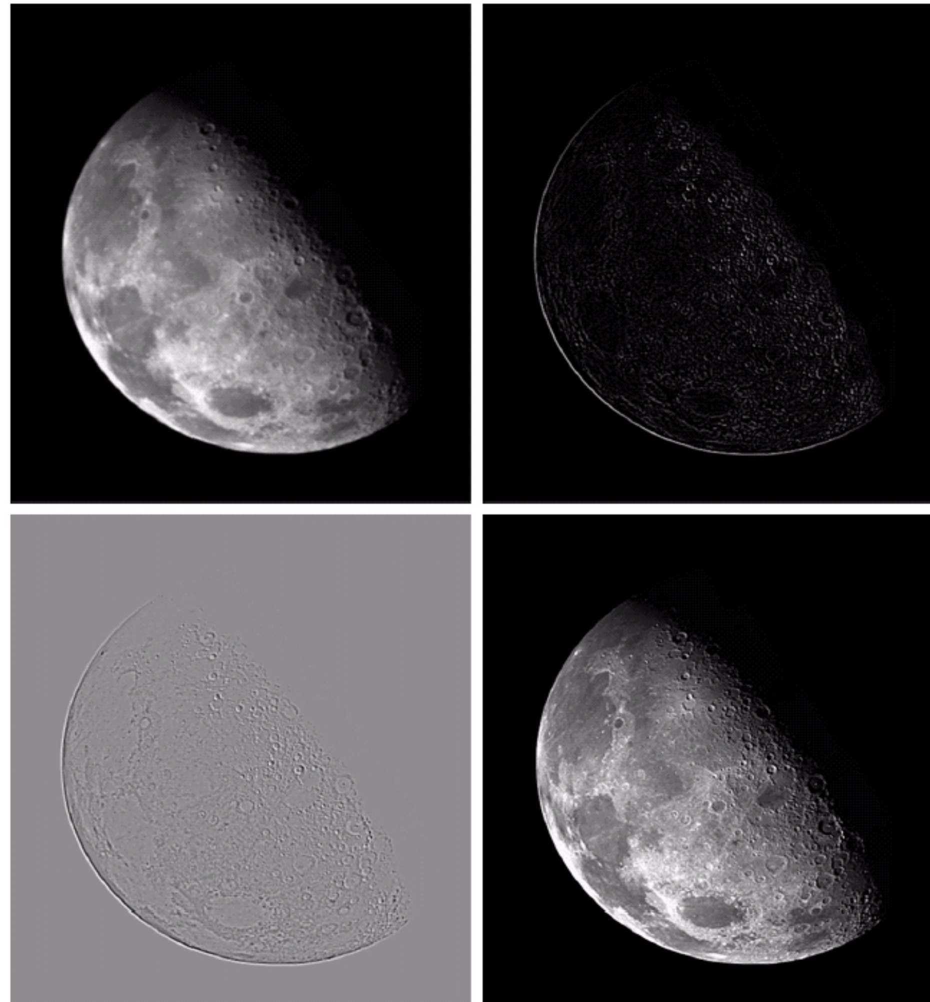
(b) Результат действия фильтра Лапласа

(c) Изображение (b), обработанное для просмотра

(d) Изображение, улучшенное в соответствии с формулой

$$I^*(x,y) = \begin{cases} I(x,y) - {}^2 I(x,y), & I(x,y) < 0; \\ I(x,y) + {}^2 I(x,y), & I(x,y) > 0. \end{cases}$$

(a) (b)



(c) (d)

Дифференциальный оператор LoG

- Дифференциальные операторы второго порядка очень чувствительны к шуму.
- Для вычисления устойчивой второй производной часто производят предварительное сглаживание изображения (подавление шума).
- Широкое применение находит оператор LoG (Laplacian of Gaussian), показывающий величину второй производной после сглаживания изображения фильтром Гаусса G:

$$^2(G \quad I) = \; ^2G \quad I,$$

^2G — оператор LoG

Гауссова фильтрация и LoG

- Одномерная гауссиана с размахом σ имеет следующий вид:

$$G(x) = ce^{-\frac{x^2}{2\sigma^2}}$$

- Двумерная гауссиана имеет вид:

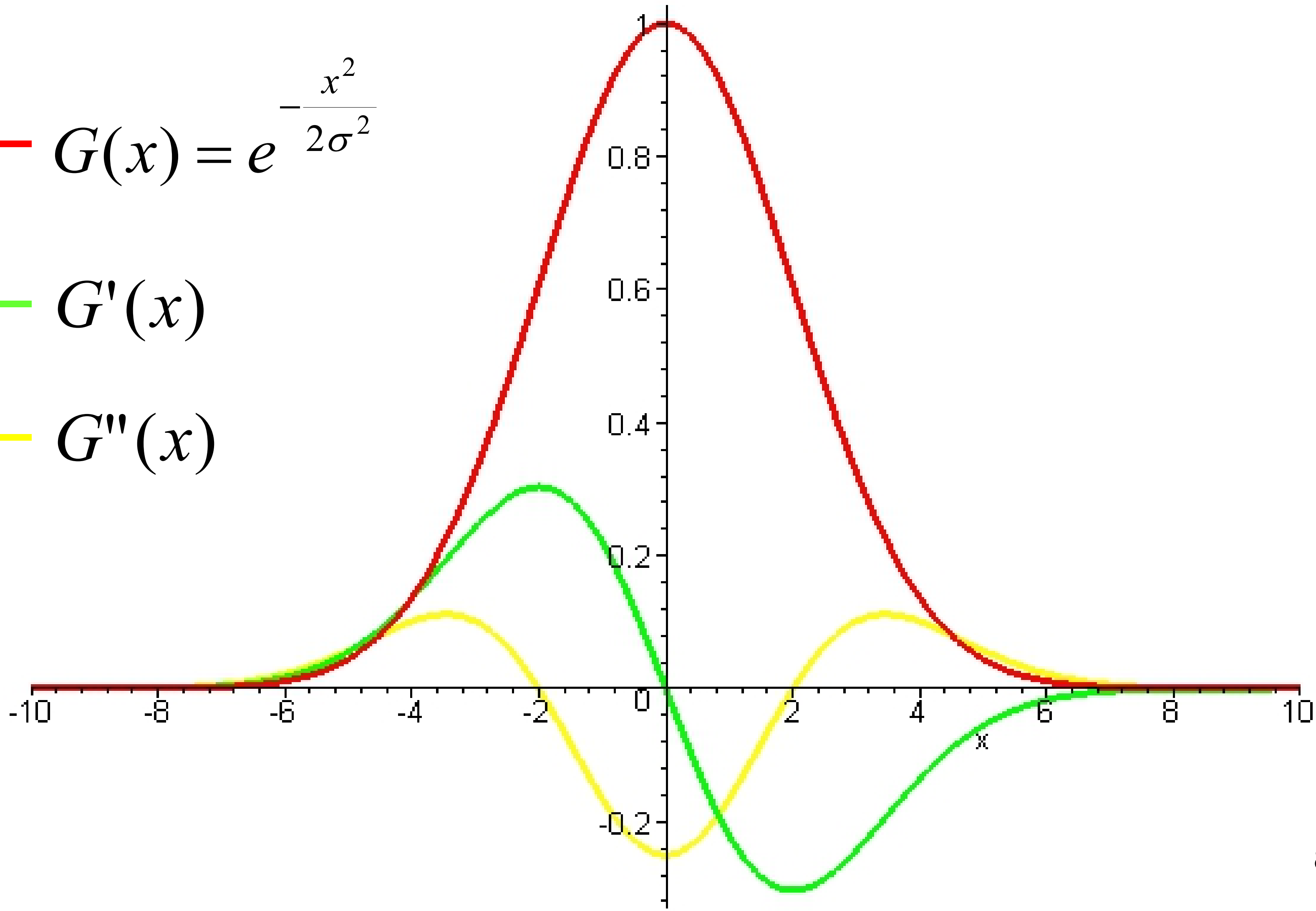
$$G(x, y) = ce^{-\frac{x^2+y^2}{2\sigma^2}}$$

Гауссиана (σ=2) и ее производные

$$G(x) = e^{-\frac{x^2}{2\sigma^2}}$$

$$G'(x)$$

$$G''(x)$$

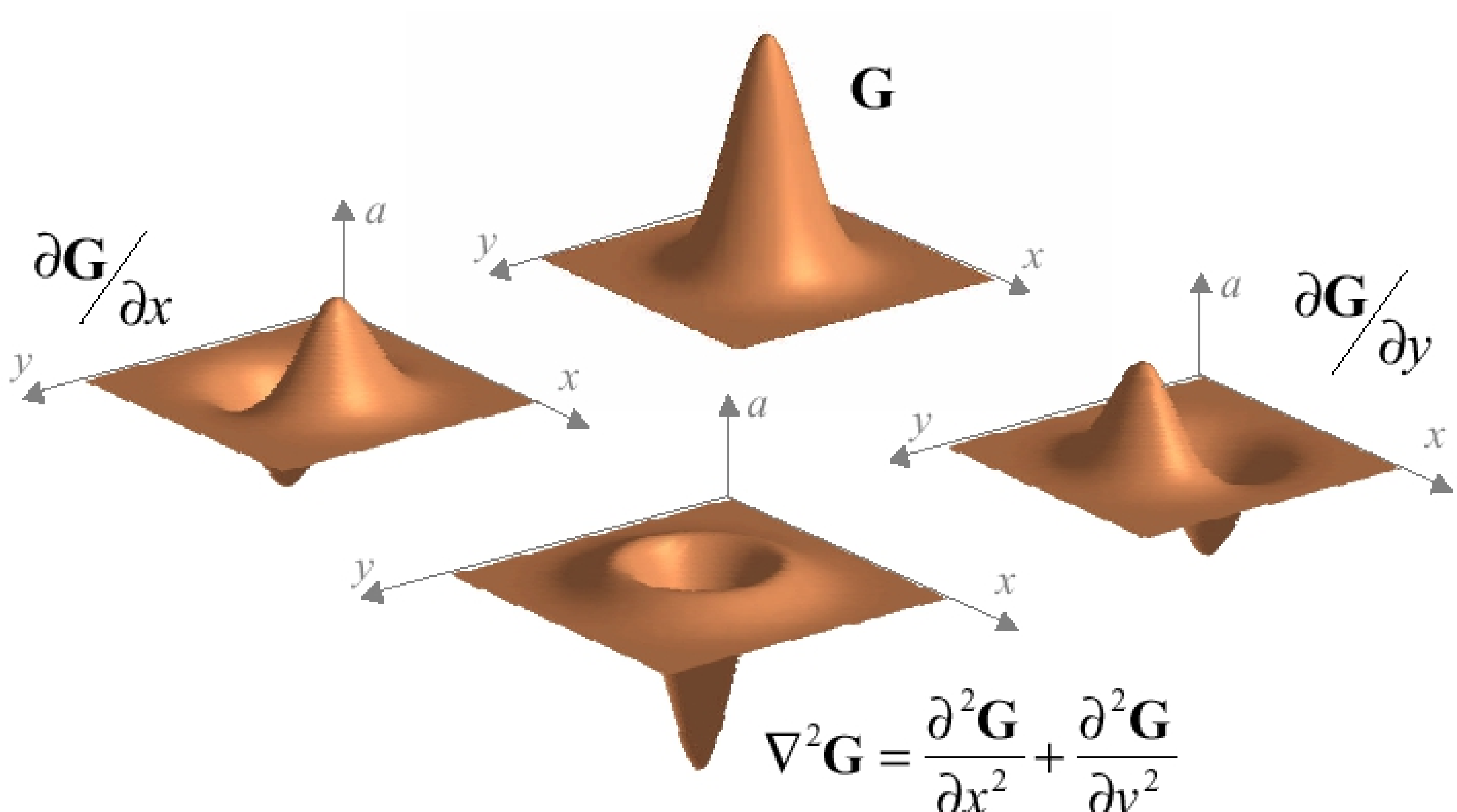


Производные гауссианы

- Первая производная одномерной гауссианы есть гауссиана, умноженная на функцию $-x$ и деленная на $2\sigma^2$.
- Вторая производная одномерной гауссианы есть разность двух четных функций. В окрестности $x = 0$ вторая производная отрицательна. Нулями второй производной являются точки $x = \pm\sigma$.

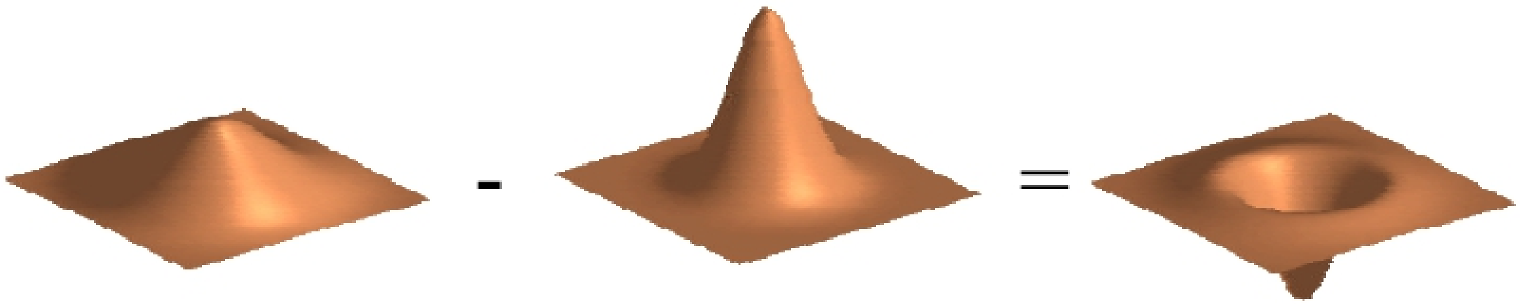
Дифференциальный оператор LoG

- Частные производные и лапласиан гауссианы $\boldsymbol{G}$ (масштаб по оси интенсивности a не соблюден).



Разность гауссиан

- Оператор LoG может быть апроксимирован разностью двух гауссиан.



- В обработке изображений часто употребляется оператор DoG (Difference of Gaussians), также известный под именем «мексиканская шляпа» (сомбреро).

Примеры масок гауссовых фильтров

$g_{3x3} = 1/16 \cdot$

1	2	1
2	4	2
1	2	1

$g_{7x7} = 1/1098 \cdot$

1	3	7	9	7	3	1
3	12	26	33	26	12	3
7	26	55	70	55	26	7
9	33	70	90	70	33	9
7	26	55	70	55	26	7
3	12	26	33	26	12	3
1	3	7	9	7	3	1

Действие фильтра LoG

Маска фильтра LoG_{3x3}

0	–1	0
–1	4	–1
0	–1	0

Исходное изображение

5	5	5	5	5	5
5	5	5	5	5	5
5	5	10	10	10	10
5	5	10	10	10	10
5	5	5	10	10	10
5	5	5	5	10	10

Результат действия фильтра LoG_{3x3}

-	-	-	-	-	-
-	0	–5	–5	–5	-
-	–5	10	5	5	-
-	–5	10	0	0	-
-	0	–10	10	0	-
-	-	-	-	-	-

Маска фильтра LoG_{11x11} ($\sigma=2$)

0	0	0	–1	–1	–2	–1	–1	0	0	0
0	0	–2	–1	–8	–9	–8	–1	–2	0	0
0	–2	–7	–15	–22	–23	–22	–15	–7	–2	0
–1	–4	–15	–24	–14	–1	–14	–24	–15	–4	–1
–1	–8	–22	–14	52	103	52	–14	–22	–8	–1
–2	–9	–23	–1	103	178	103	–1	–23	–9	–2
–1	–8	–22	–14	52	103	52	–14	–22	–8	–1
–1	–4	–15	–24	–14	–1	–14	–24	–15	–4	–1
0	–2	–7	–15	–22	–23	–22	–15	–7	–2	0
0	0	–2	–1	–8	–9	–8	–1	–2	0	0
0	0	0	–1	–1	–2	–1	–1	0	0	0

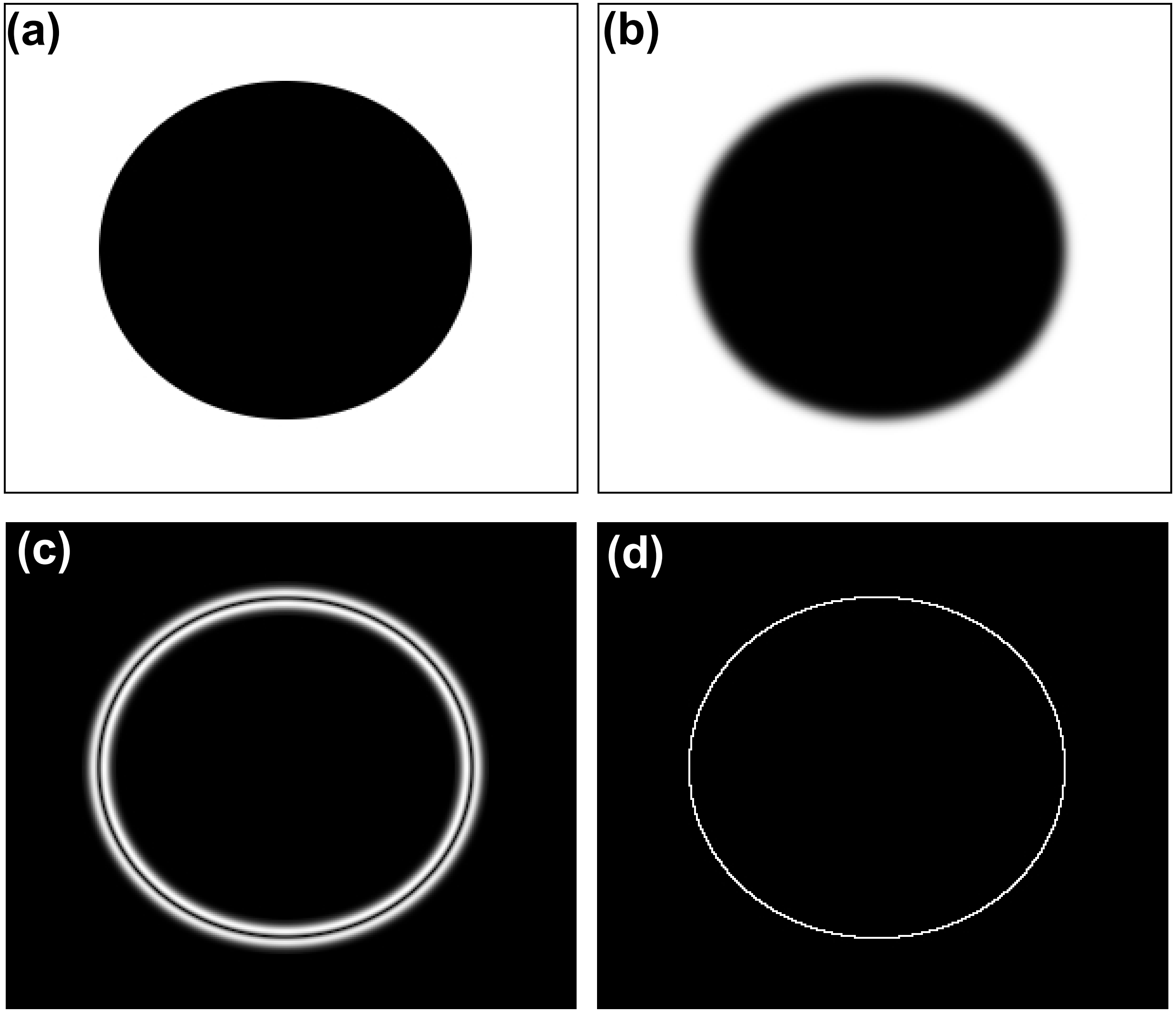
Нули второй производной и выявление контуров

- Поиск границ путем определения нулей второй производной изображения основан на том факте, что ступенеобразная граница есть резкий скачок интенсивности.
- Первая производная изображения должна иметь экстремум в точках границы, а вторая производная в этих точках должна обращаться в нуль.
- Главным недостатком детекторов границ, основанных на поиске нулей второй производной является их сильная зависимость от размера объектов и чувствительность к шуму.

Детектор контуров Марра–Хильдрета

- Оператор Марра–Хильдрета обнаруживает контуры на изображении путем поиска нулей оператора LoG от изображения ($^{2}G \quad I$).

Пример действия детектора контуров Марра–Хильдрета



(a) **Исходное изображение**

(b) **Изображение (a), сглаженное фильтром Гаусса 17x17**

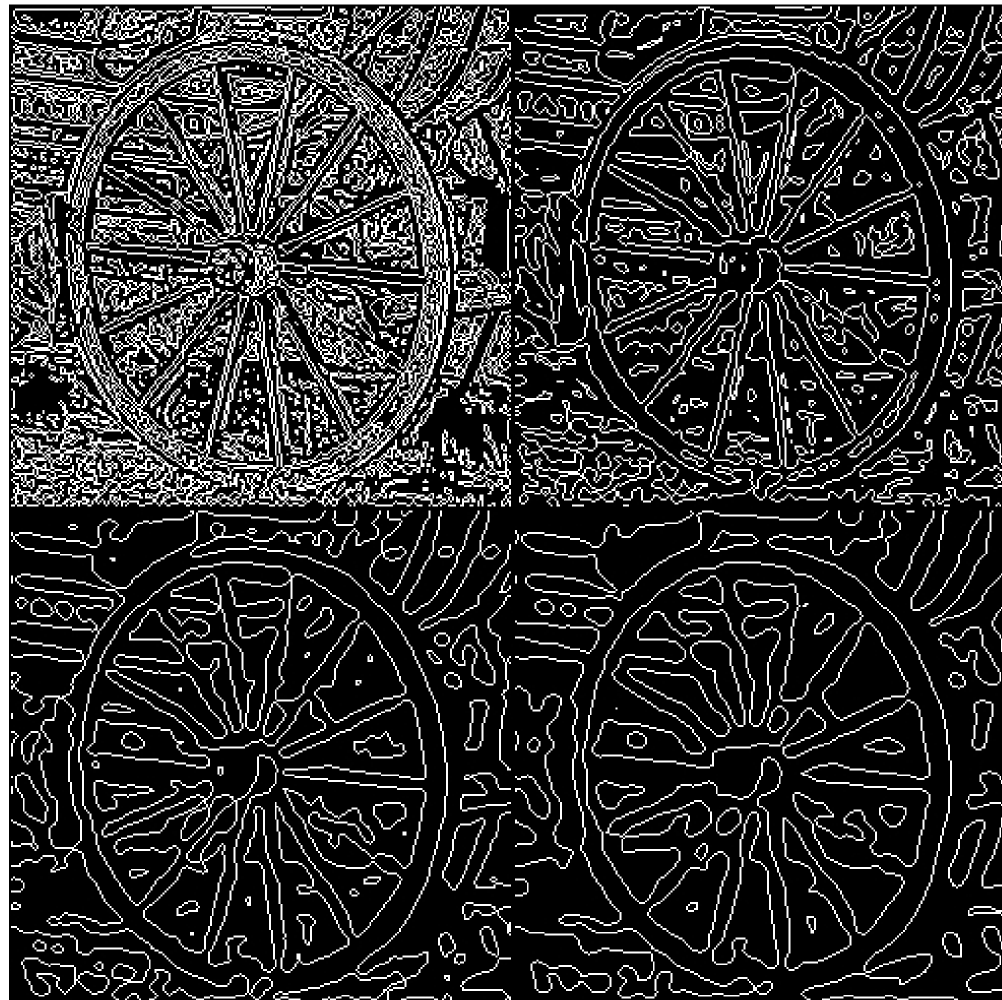
(c) **Абсолютное значение Лапласиана изображения (b)**

(d) **Нули Лапласиана — локальные минимумы изображения (c)**

Пример действия детектора контуров Марра–Хильдрета

(a)

(b) (c)



(d) (e)

(a) Исходное изображение

Результаты действия алгоритма Марра–Хильдрета при размахе фильтра Гаусса:

(b) $\sigma = 1$

(c) $\sigma = 2$

(d) $\sigma = 3$

(e) $\sigma = 4$

Детектор контуров Марра–Хильдрета

- Преимущества
  - Поиск контуров ведется во всех направлениях одновременно
  - Нули легко обнаруживаются по изменениям знака
  - Нули двумерной функции являются связными замкнутыми кривыми (полезно при выделении объектов на изображении)
- Недостатки
  - Вторая производная слишком чувствительна к шуму
  - Не все контуры являются частями замкнутых кривых

Детектор контуров Канни

1. Произвести свертку изображения с гауссианой
2. Вычислить градиент ∇I
3. Выполнить подавление немаксимальных пикселов на $\|\nabla I\|$ в направлении $\angle \nabla I$
4. Выполнить двухпороговую бинаризацию, используя $\|\nabla I\|$ как меру силы контуров

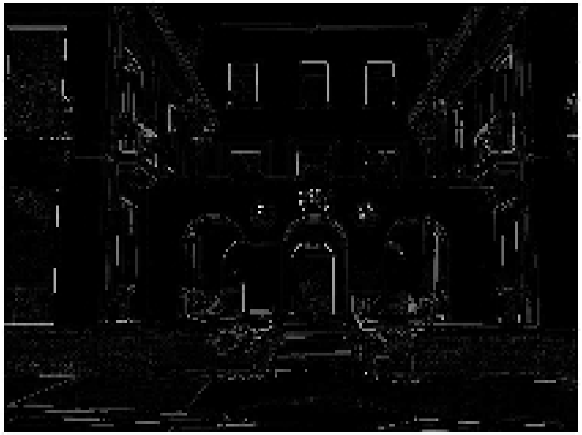
Двухпороговая бинаризация (бинаризация с гистерезисом)

- Используется два порога T_{high} и T_{low}
- Порог T_{high} используется для поиска сильных контуров
- Порог T_{low} используется для поиска слабых контуров
- Результат – объединение сильных контуров и слабых контуров, связных с сильными.

Детектор контуров Канни

Исходное изображение

Модуль градиента



Модуль градиента после подавления немаксимальных пикселов

Детектор контуров Канни

Слабые контуры

Сильные контуры

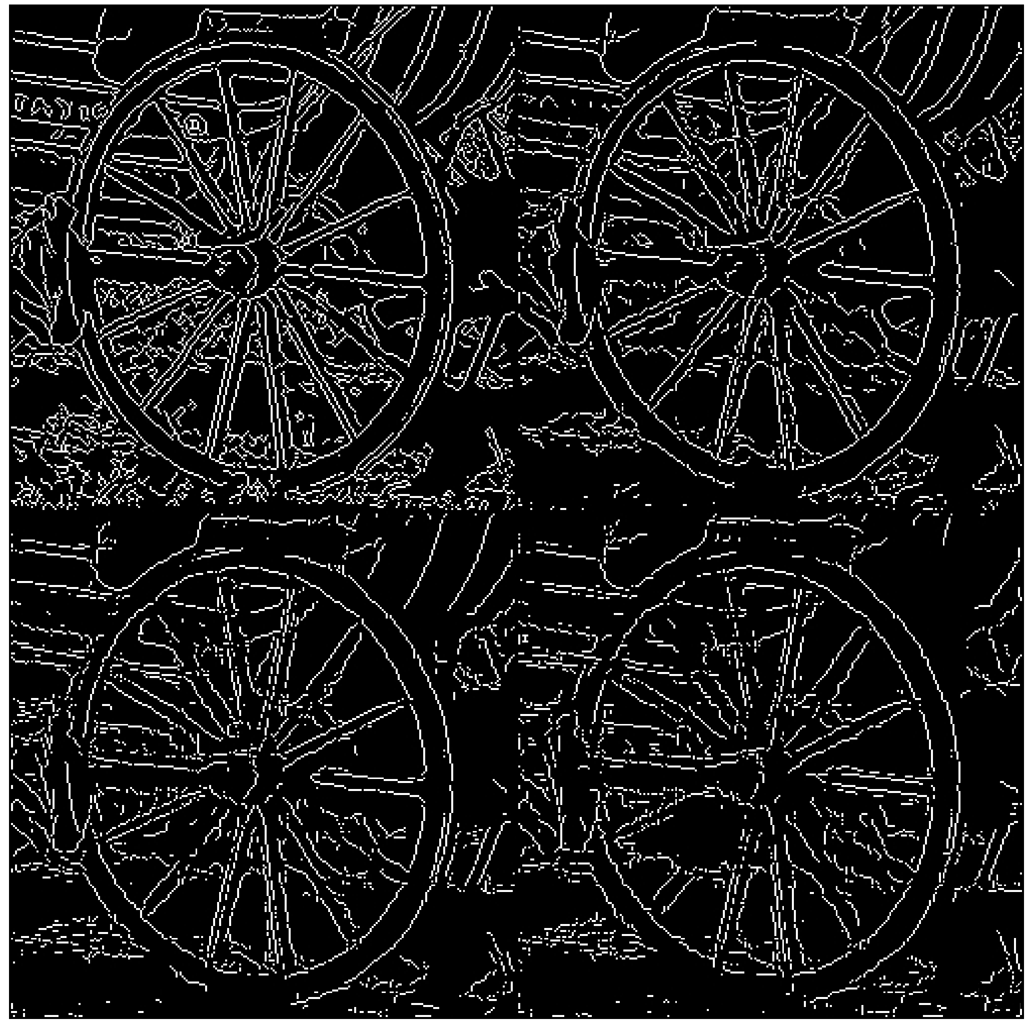
Сильные контуры и связные с ними слабые контуры

Пример действия детектора контуров Канни

Пример действия алгоритма детектора контуров Канни

(a)

(b) (c)



(d) (e)

(a) Исходное изображение

Результаты действия алгоритма Канни, при размахе фильтра Гаусса:

(b) $\sigma = 1$

(c) $\sigma = 2$

(d) $\sigma = 3$

(e) $\sigma = 4$