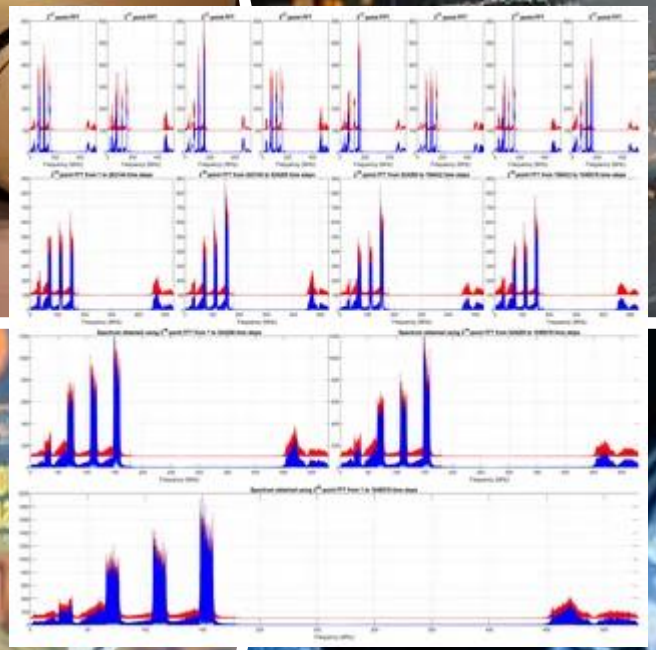
Пространственная обработка изображений





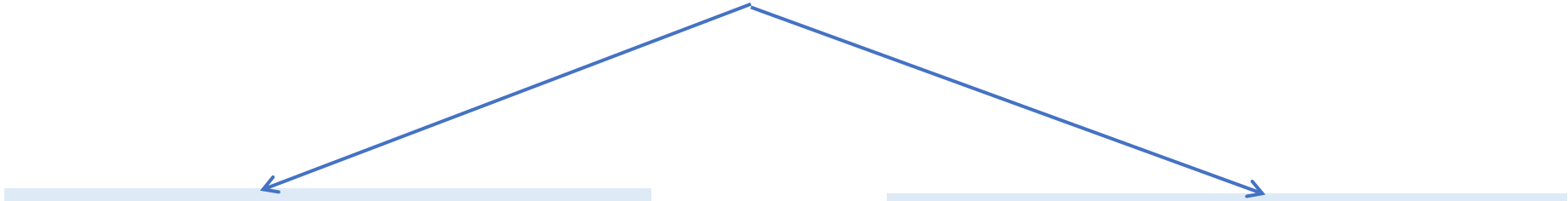




Лекция 5

Лектор: Демяненко Я.М., ФИИТ, 2026

Обработка изображений



В пространственной области

В частотной области

Оба подхода эквивалентны. Выбор — вопрос удобства и скорости

Это классика, построенная на
- линейной алгебре
- математическом анализе
- цифровой обработке сигналов

Работа с фильтрами, градиентами, морфологией позволяет развить интуитивное понимание того, как устроена визуальная информация

Критически важный базис

Многие современные методы (даже глубокое обучение) используют концепции и результаты пространственной обработки как сырые признаки или часть предобработки.

Без понимания, как работают фильтры, пирамиды или детекторы краев, невозможно глубоко понять, что именно «видит» и на чем учится сверточная нейронная сеть.

CNN — это каскад обучаемых фильтров. Понимание свёртки = понимание первого слоя ResNet

Инструмент для дополнения Deep Learning

Не все задачи решаются в лоб нейросетями

Часто гибридные подходы, где пространственная обработка используется для

- предобработки
- постобработки
- выделения областей интереса

оказываются более эффективными, робастными и требующими меньше данных для обучения.

Пример: Гибрид: Оцу + opening + CNN = 97.96% в задаче малярии (лекция 3)

Пространственная обработка



Яркостные преобразования

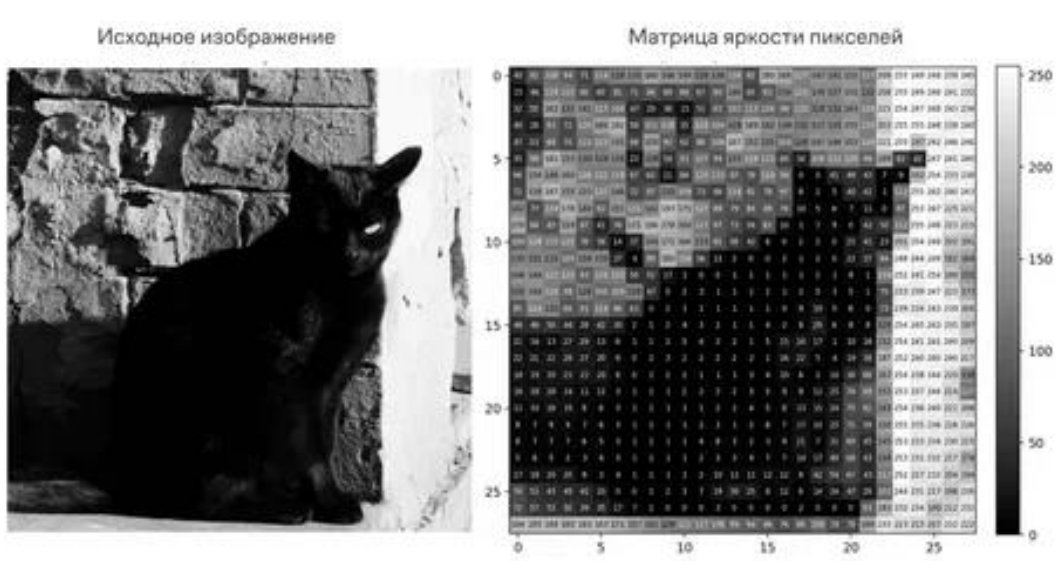
- Оперируют **отдельными пикселями**
- Управляют **контрастом и цветностью**

Пространственная **фильтрация**

- Оперирует **окрестностью** точки
- Управляет **резкостью**

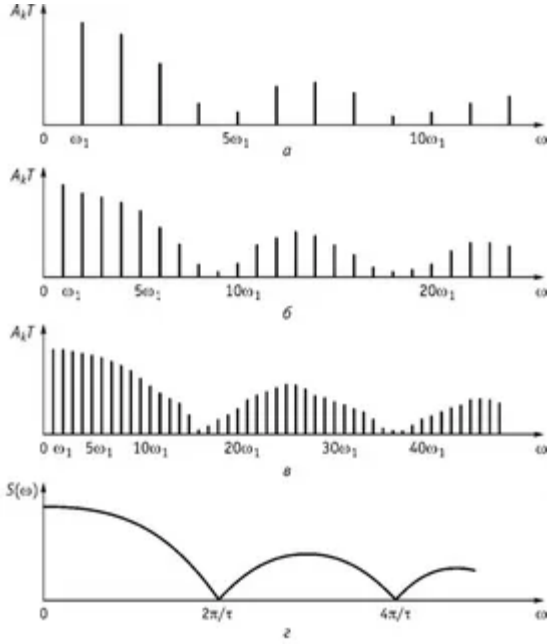
Сегодня мы пройдём путь: пиксель → окрестность → спектр → CNN

Два взгляда на изображение



Прямое ДПФ

Обратное ДПФ



Изображение — это сигнал. Его можно описывать двумя способами

Дорожная карта лекции

Арифметика изображений

Пространственная фильтрация

Спектр и частотные фильтры

Теорема о свёртке — мост

CNN как каскад фильтров

Математическое описание

$$g(x, y) = T[f(x, y)]$$

где T - это:

- для нормализации: линейное преобразование
- для аугментации: аффинное/нелинейное преобразование
- для классической обработки: оператор свертки/морфологии

Процессы пространственной обработки

$$g(x, y) = T[f(x, y)]$$

Входные данные:

- одно изображение или набор изображений

- окрестности разной формы и разного размера
  - окрестность = рецептивное поле фильтра

Арифметические операции

$$g(x, y) = Op\ (f_1(x, y), f_2(x, y))$$

for all pixel positions x,y:

    out[x,y] = func(image1[x,y], image2[x,y])

Addition / Blending

$$g(x, y)= f_1(x, y)+ f_2(x, y)$$



$$g(x, y)= \alpha f1(x, y)+ \beta f2(x, y)$$

Современное применение: MixUp, CutMix — аугментации через арифметику

Subtraction

$$g(x, y)= f_1(x, y)- f_2(x, y)$$



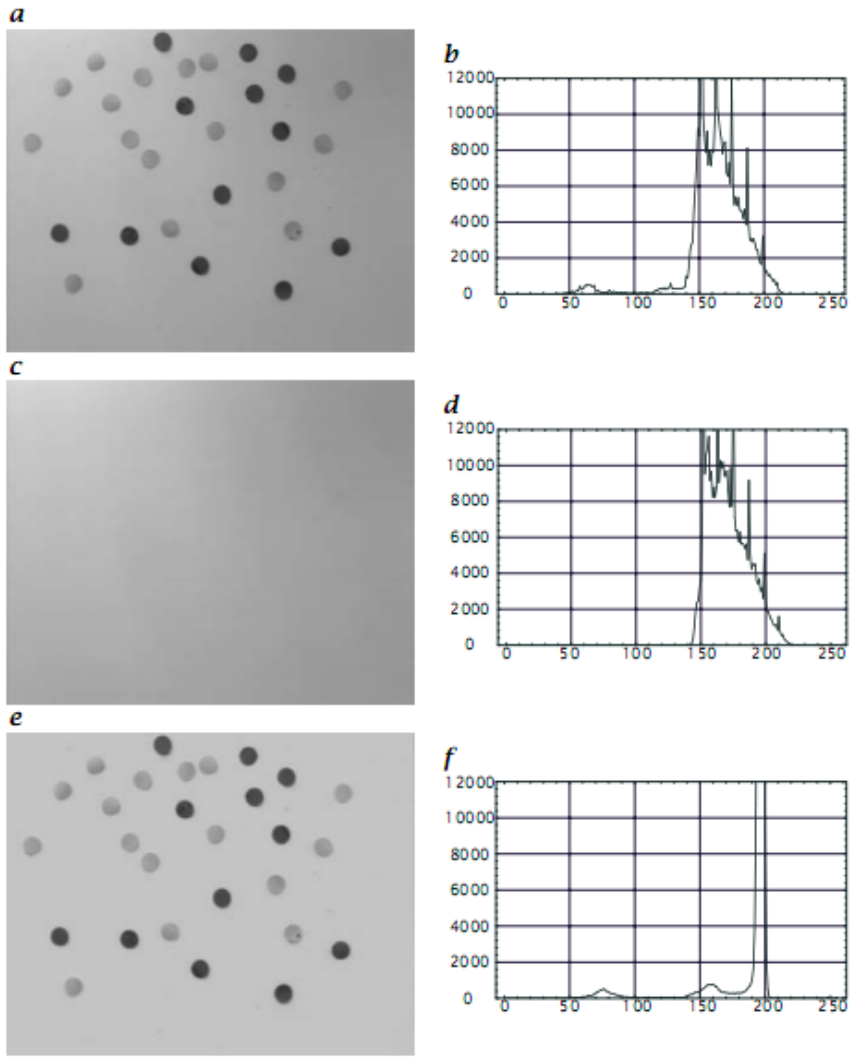
$$g(x, y)= |f1(x, y)- f2(x, y)|$$

Вычитание фона — базовый метод в видеонаблюдении

Коррекция неоднородной освещенности делением изображения на опорное

$$G' = c \cdot G / R$$

Деление на опорное изображение убирает градиент освещения



Виды шума

Гауссов шум — от сенсора
Импульсный шум — от сбоев
Смешанный шум — реальность

Гауссов



Импульсный (соль.перец)





оригинал

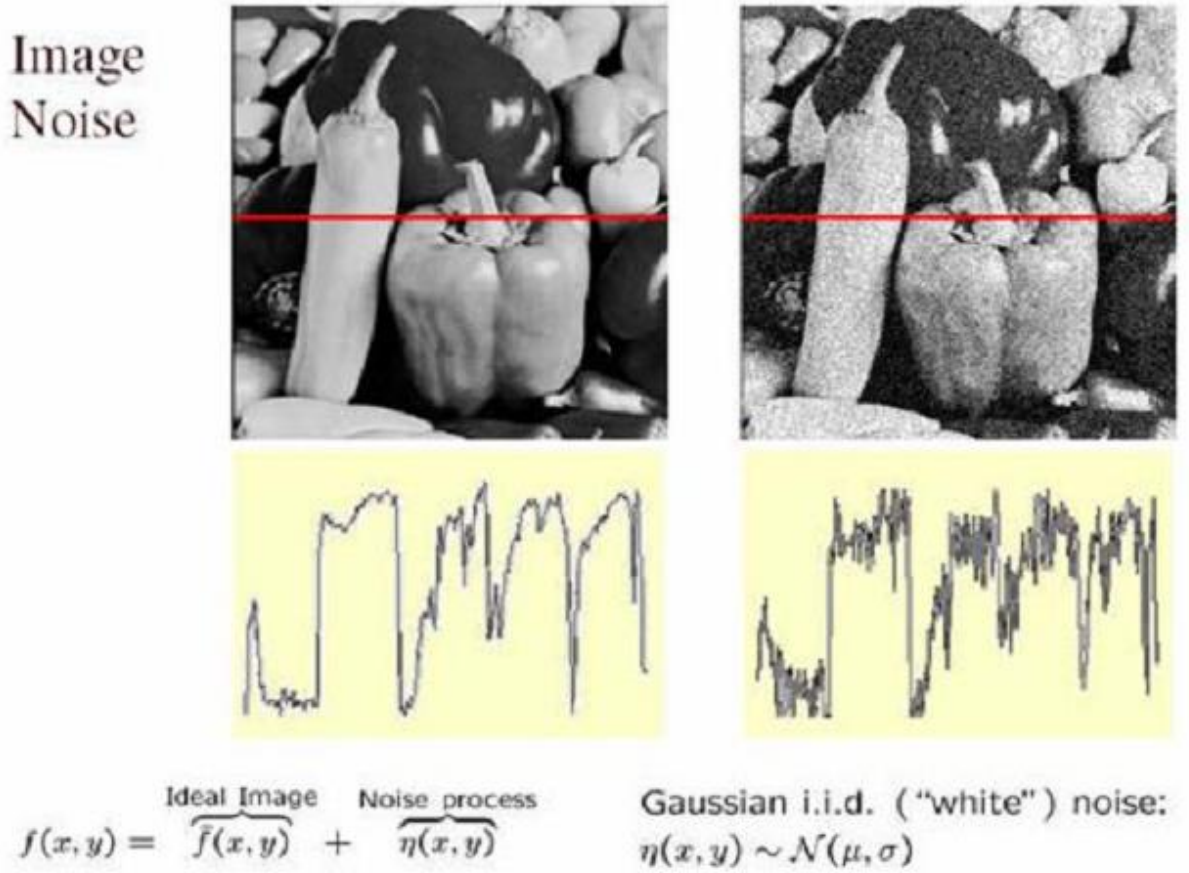


смешанный

Гауссов шум

$I = g + Err, Err \sim N(0, \sigma^2)$

Шум некоррелирован и имеет нулевое среднее



Source: M. Hebert

Усреднение кадров

10 зашумленных изображений

Усреднение по 10 изображениям



$$I(i,j) = g_r(i,j) + Err(i,j);$$

$$\bar{I}(i,j) = \frac{1}{N}\sum_{k=1}^{N} I_k(i,j);$$

Так работает Night Mode в смартфонах

Результаты усреднения

Исходное



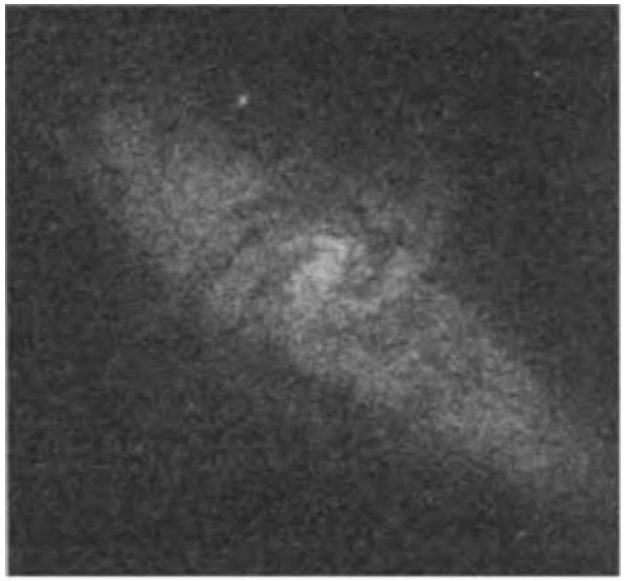
K=8



K=16



Больше кадров — чище, но дольше



K=64



K=128



С шумом

Дорожная карта лекции

Арифметика изображений

Пространственная фильтрация

Спектр и частотные фильтры

Теорема о свёртке — мост

CNN как каскад фильтров

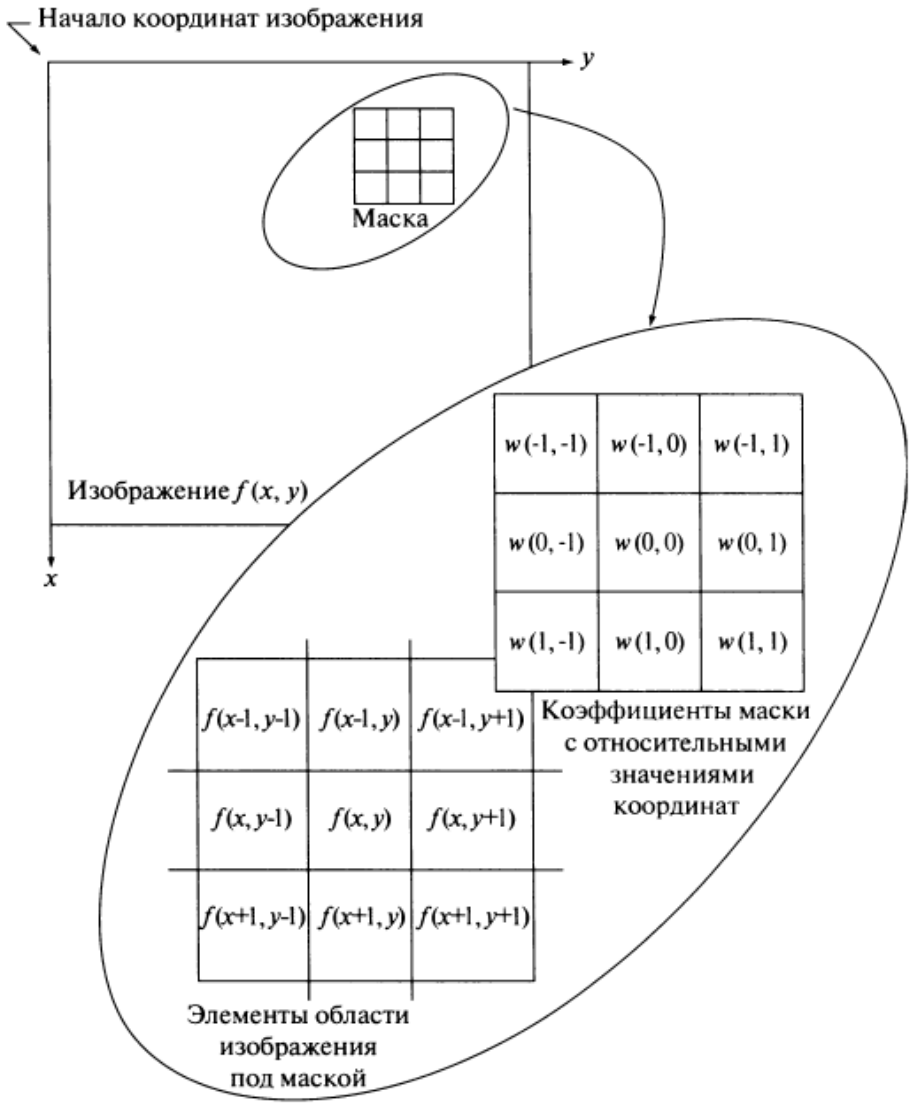
Схема пространственной фильтрации

Фильтр задается матрицей

Фильтр = ядро = маска = оператор = шаблон = окно = оператор

Что может отличаться

- размер окрестности
- положение пикселя относительно окна
- симметрия



Решение проблемы краевых пикселей

Экстраполяция значений краевых пикселей

Зеркальное отражение исходного изображения относительно каждого края, и центральное по углам

Тесселяция исходного изображения /Использование цикличности/

Игнорирование /Меньшее изображение/

Использование части маски

Добавление пограничных областей

В CNN чаще используют padding=0 или reflect

Корреляция vs Свёртка

$$g(x,y) = \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$ корреляция

$$g(x,y) = \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$ свертка

В CNN используют корреляцию, но называют свёрткой
Разница — в перевороте ядра, для обучаемых ядер не важна

Основные свойства свёртки

Линейность:

$\text{filter}(f_1 + f_2) = \text{filter}(f_1) + \text{filter}(f_2)$

Инвариантность к сдвигу — не зависит от сдвига пиксела:

$\text{filter}(\text{shift}(f)) = \text{shift}(\text{filter}(f))$

Любой линейный оператор, инвариантный к сдвигу, может быть записан в виде свертки
т. е.
Любой линейный сдвиг-инвариантный оператор = свёртка

Свойства (коммутативность, ассоциативность и т.д.)

- Коммутативность: a * b = b * a
  - Нет никакой разницы между изображением и ядром фильтра
- Ассоциативность: a * (b * c) = (a * b) * c
  - Последовательное применение фильтров: (((a * b1) * b2) * b3) эквивалентно применению такого фильтра: a * (b1 * b2 * b3)
- Дистрибутивность по сложению:
  a * (b + c) = (a * b) + (a * c)
- Домножение на скаляр можно вынести за скобки:
  ka * b = a * kb = k (a * b)
- Единица: e = [..., 0, 0, 1, 0, 0, ...], a * e = a

Простейшие фильтры: что происходит?



Original

0	0	0
0	0	1
0	0	0

?

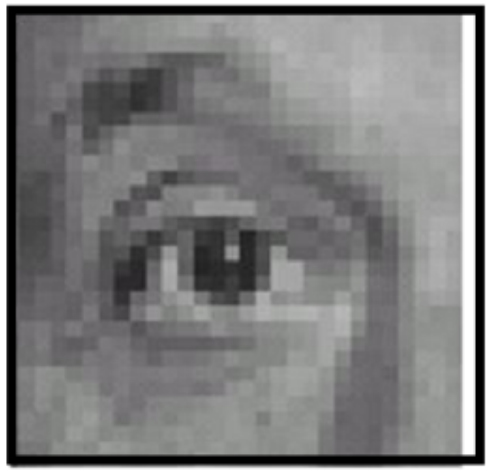
Source: D. Lowe

Простейшие фильтры: identity, shift



Original

0	0	0
0	0	1
0	0	0



Shifted left
By 1 pixel

Identity — единица свёртки

Source: D. Lowe

Простейшие фильтры: что происходит?



Original

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

?

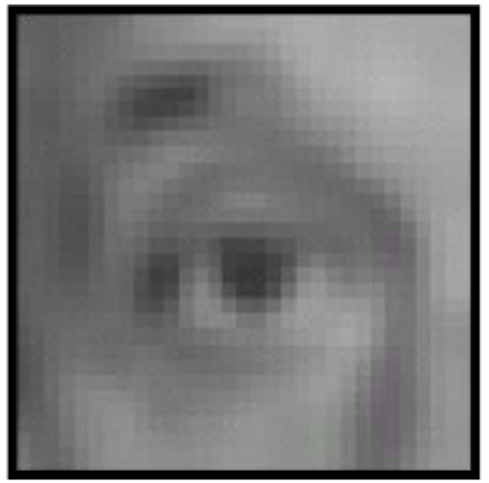
Source: D. Lowe

Простейшие фильтры: Box-фильтр (усреднение 3×3)



Original

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$



Blur (with a box filter)

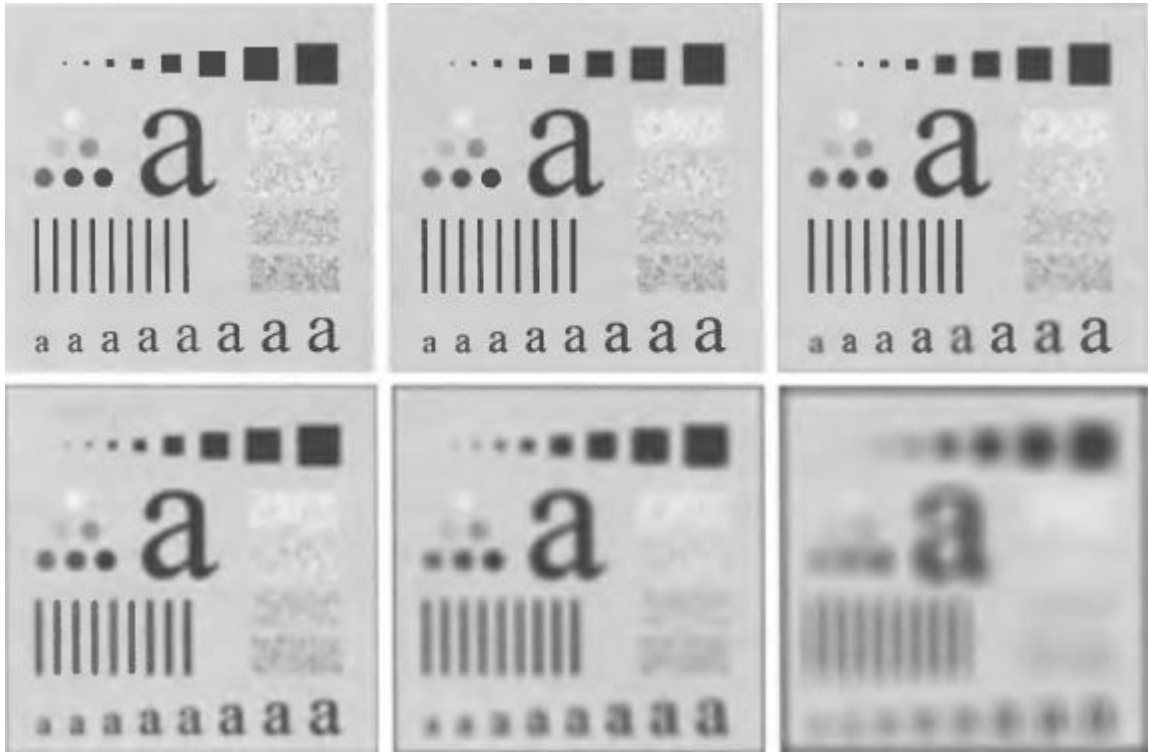
Простое размытие, но даёт артефакты на краях

Source: D. Lowe

Результаты сглаживания усредняющим фильтром

Исходное 500x500

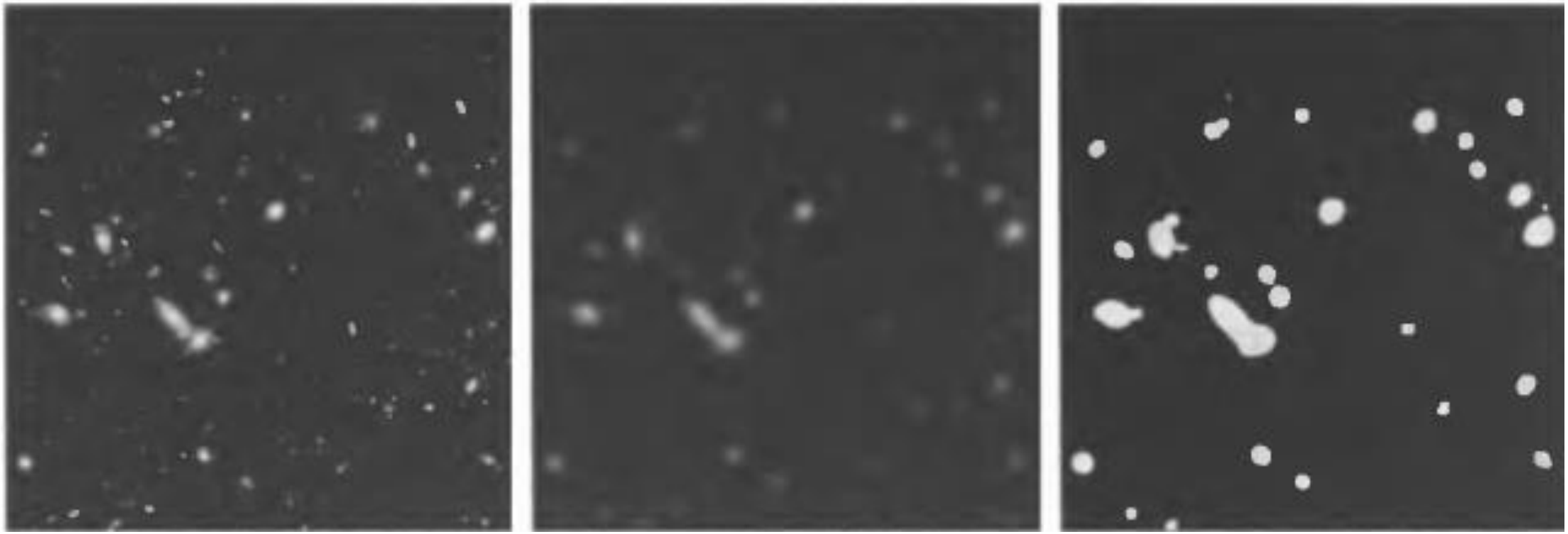
3 5 9 15 25 35 45 55



3 5 9 15 35

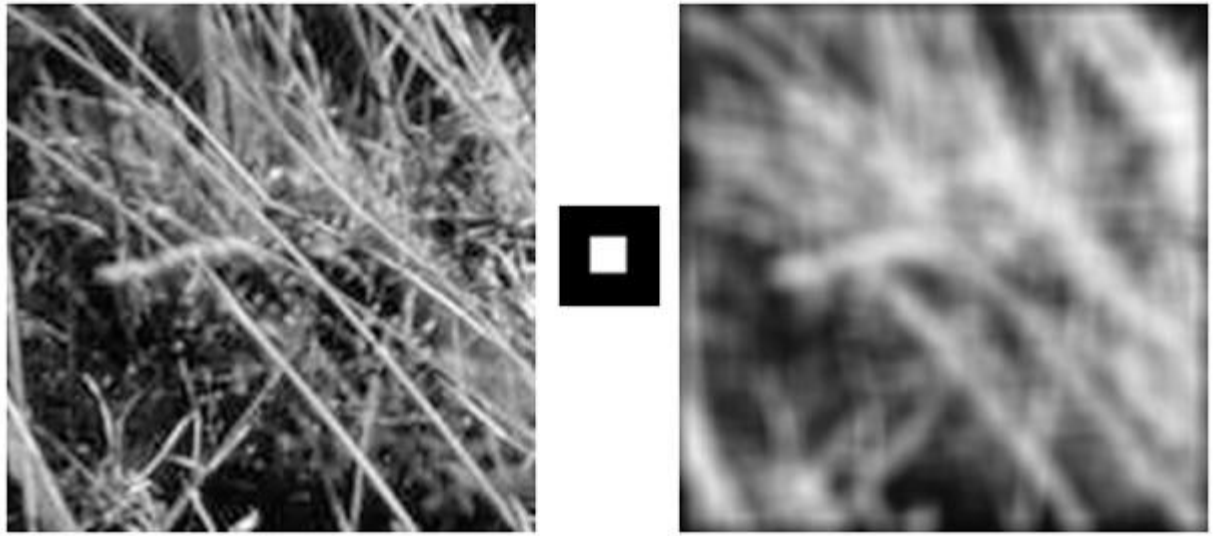
Чем больше ядро, тем сильнее размытие

Сглаживающая маска + пороговое обнаружение



Сглаживание перед бинаризацией убирает шум

Сглаживание с box-фильтром. Что вы видите?



Сглаживание с box-фильтром (паразитные линии)

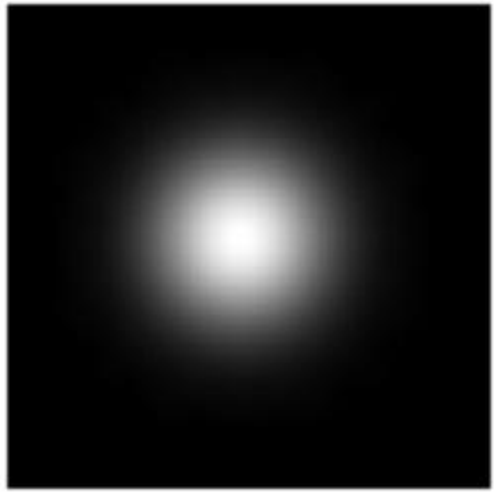


Box даёт квадратные артефакты — нужен Гаусс

Сглаживание: fuzzy blob

- Точка света, наблюдаемая с расфокусированного объектива, выглядит как кружок света, а усреднение дает квадратик

- Другой способ: взвешиваем вклад пикселей по окрестности с учетом близости к центру



Взвешенное среднее учитывает близость к центру

Фильтр взвешенного среднего

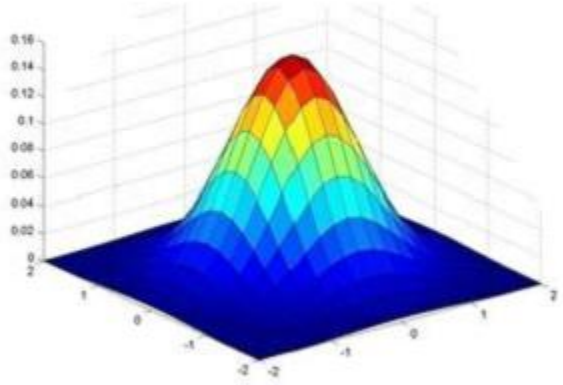
$$\frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Это дискретный Гаусс 3×3

$$g(x,y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)}$$

Гауссов фильтр

$$h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x+y)^2}{2\sigma^2}}$$



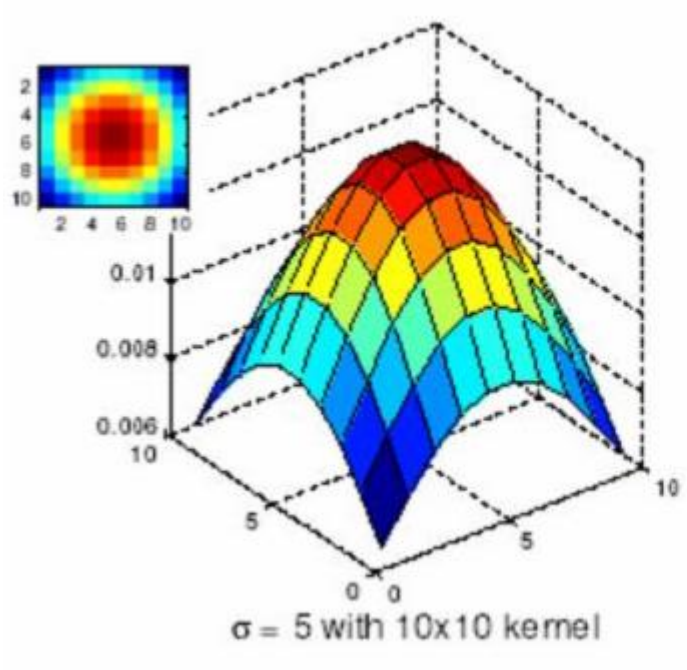


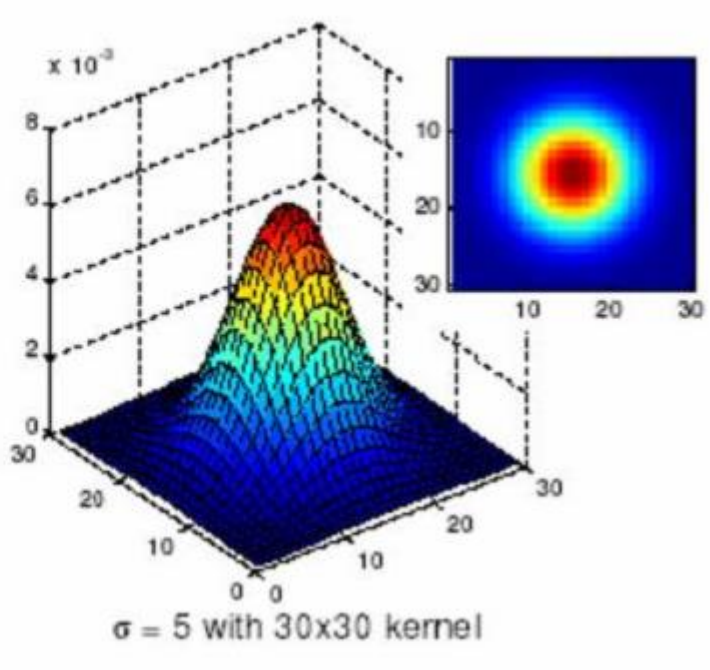
0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

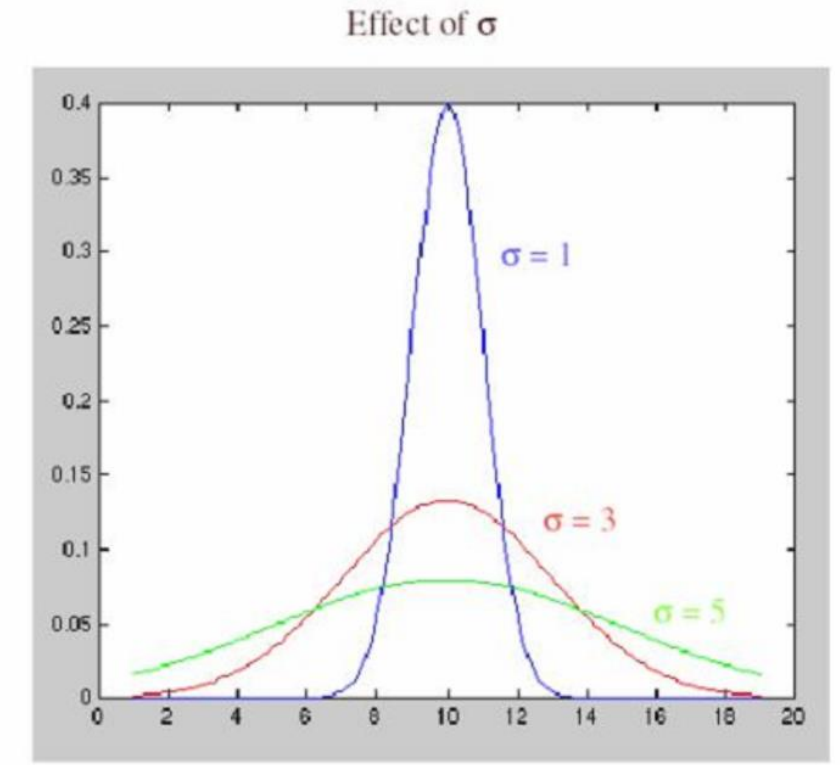
5 x 5, σ = 1

Source: C. Rasmussen

Гауссов фильтр: выбор размера ядра





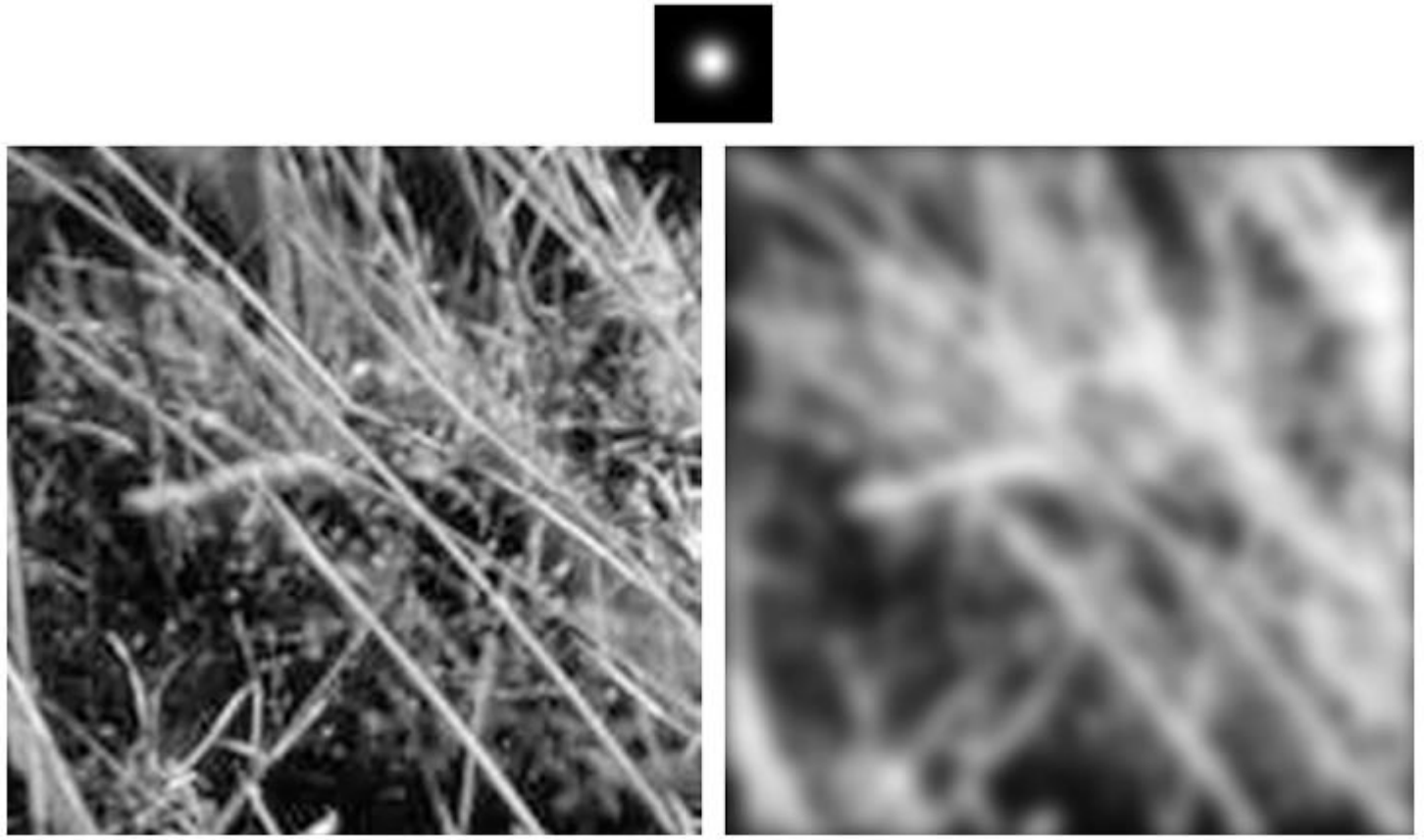


Source: K. Grauman

Размер ядра ≈ 6σ

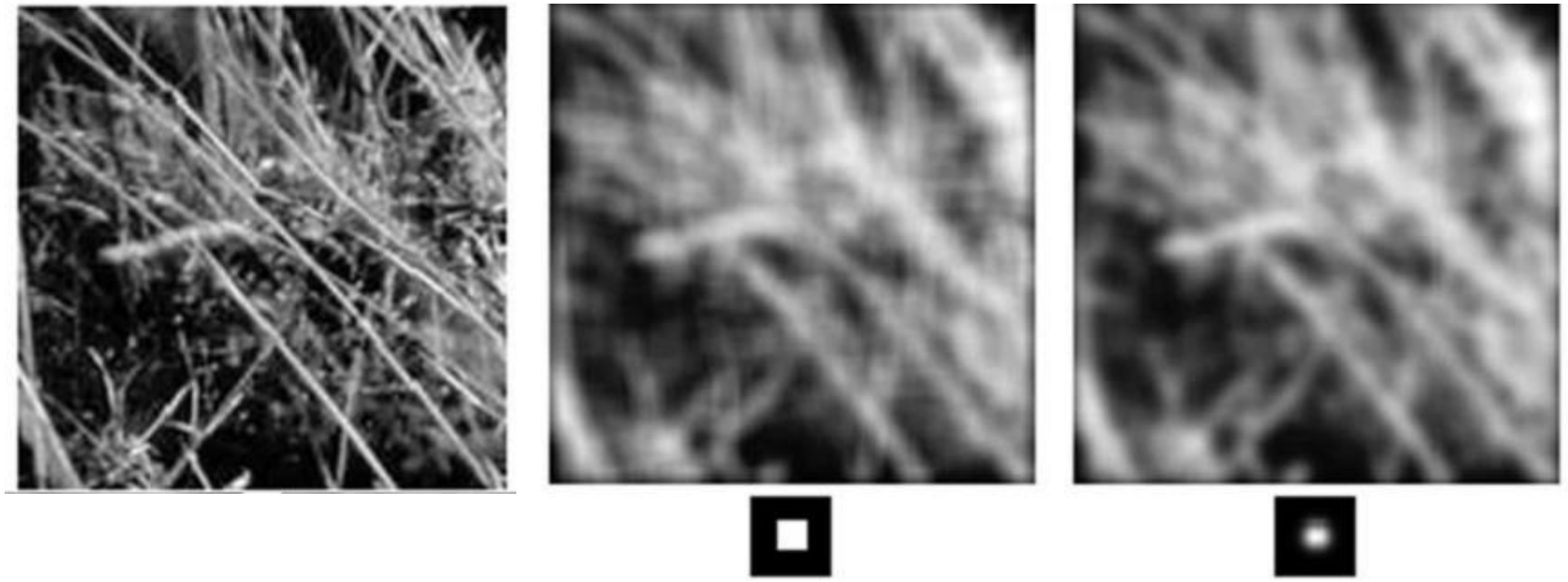
Оптимальный полуразмер фильтра = 3σ
→ 99.7% энергии

Сглаживание методом Гаусса



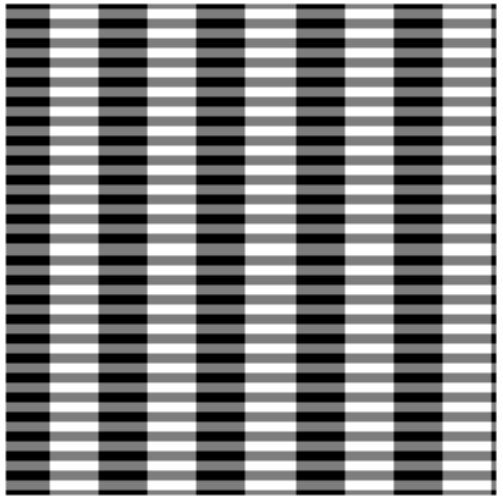
Гаусс не даёт паразитных линий, в отличие от box

Сравнение: box vs Gaussian

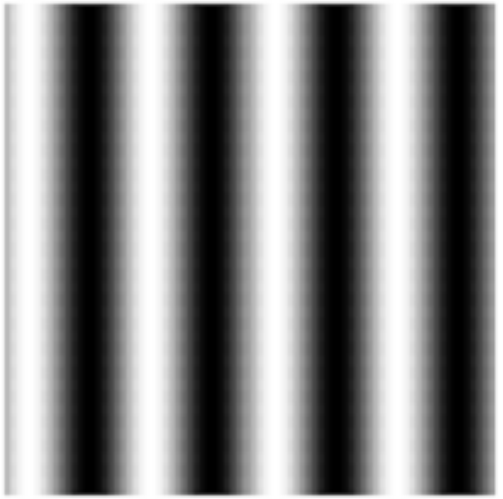


Box → квадраты. Gaussian → мягкое размытие

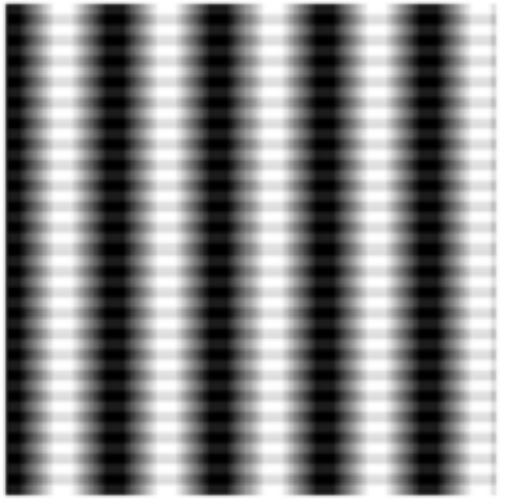
Результаты свёртки по Гауссу и константе



Исходное изображение



Фильтр Гаусса с Sigma = 4



Усреднение по 49 пикселям (7x7)

Результаты свёртки по Гауссу и константе

Свойства фильтра Гаусса

- Сглаживание несколько раз фильтром с маленьким ядром дает результат сглаживания с большим ядром
- Свертка 2 раза фильтром радиуса σ дает тот же результат, что один раз фильтром радиуса σ√2

Это следствие теоремы о свёртке

Сепарабельность

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} = \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}\right)\left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{y^2}{2\sigma^2}}\right)$$

Выигрыш: $O(n^2) \rightarrow O(2n)$. Для 15×15: 225 → 30 операций

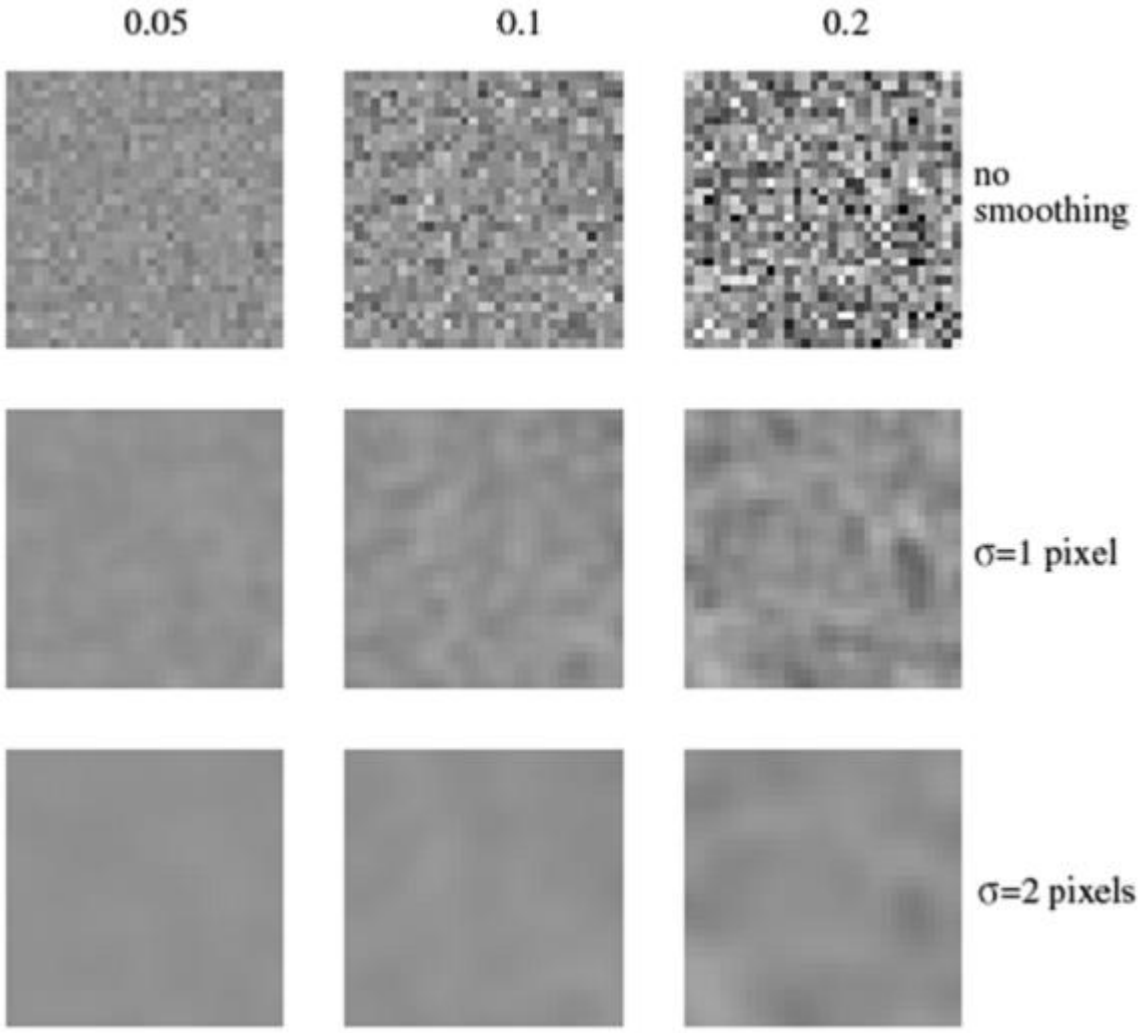
2D свёртка через 1D



Source: K. Grauman

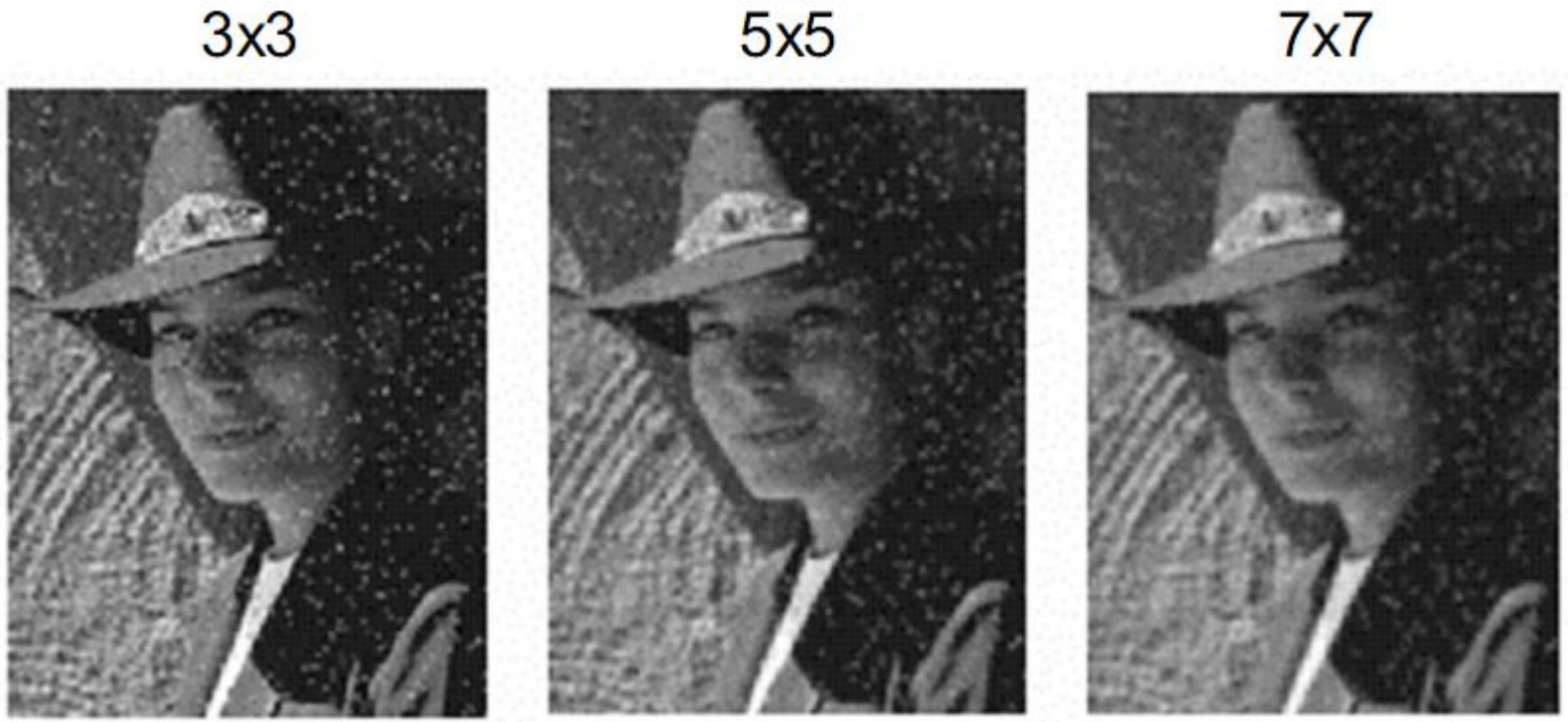
Подавление гауссова шума

уровни шума



Чем больше σ фильтра, тем сильнее размытие

Подавление шума «соль и перец»



Чем результат плох?

Slide by S. Lazebnik

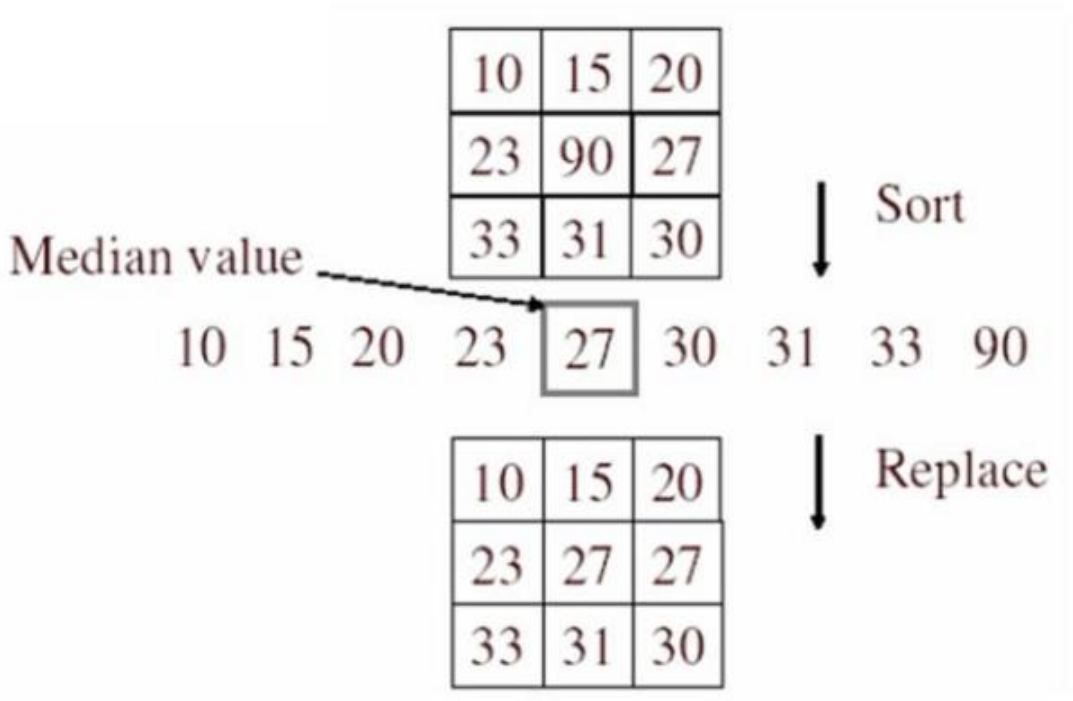
Нелинейное усреднение

Фильтры, основанные на порядковых статистиках:

- классический медианный
- максимума
- минимума

Эффективны при фильтрации импульсных шумов (соль и перец) – значительно меньший эффект расфокусировки

Медианный фильтр: примеры



Slide by S. Lazebnik

x = [2 80 6 3]

y[1] = медиана[2 2 80] = 2

y[2] = медиана[2 80 6] = медиана[2 6 80] = 6

y[3] = медиана[80 6 3] = медиана[3 6 80] = 6

y[4] = медиана[6 3 3] = медиана[3 3 6] = 3

и в итоге:

y = [2 6 6 3] — выход медианного фильтра

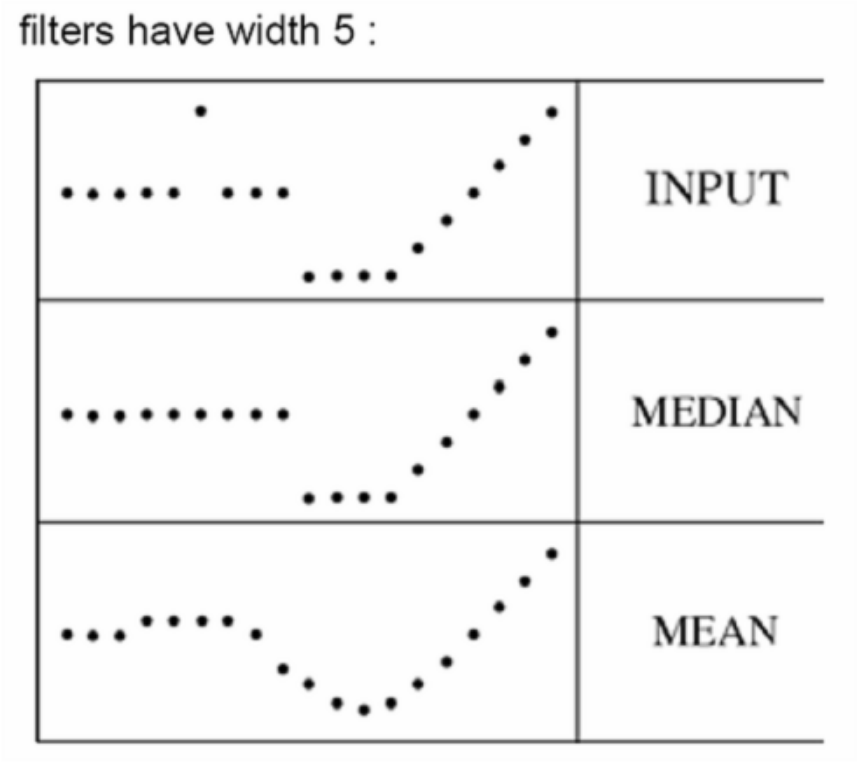
Нелинейность медианного фильтра

$$M([\ldots\ 0\ 1\ 0\ 0\ \ldots]+[\ldots\ 0\ 0\ 1\ 0\ \ldots])=[\ldots\ 0\ 1\ 1\ 0\ \ldots]$$

$$M[\ldots\ 0\ 1\ 0\ 0\ \ldots]+M[\ldots\ 0\ 0\ 1\ 0\ \ldots]=[\ldots\ 0\ 0\ 0\ 0\ \ldots]$$

$M[a+b] \neq M[a]+M[b]$

Медиана vs Среднее

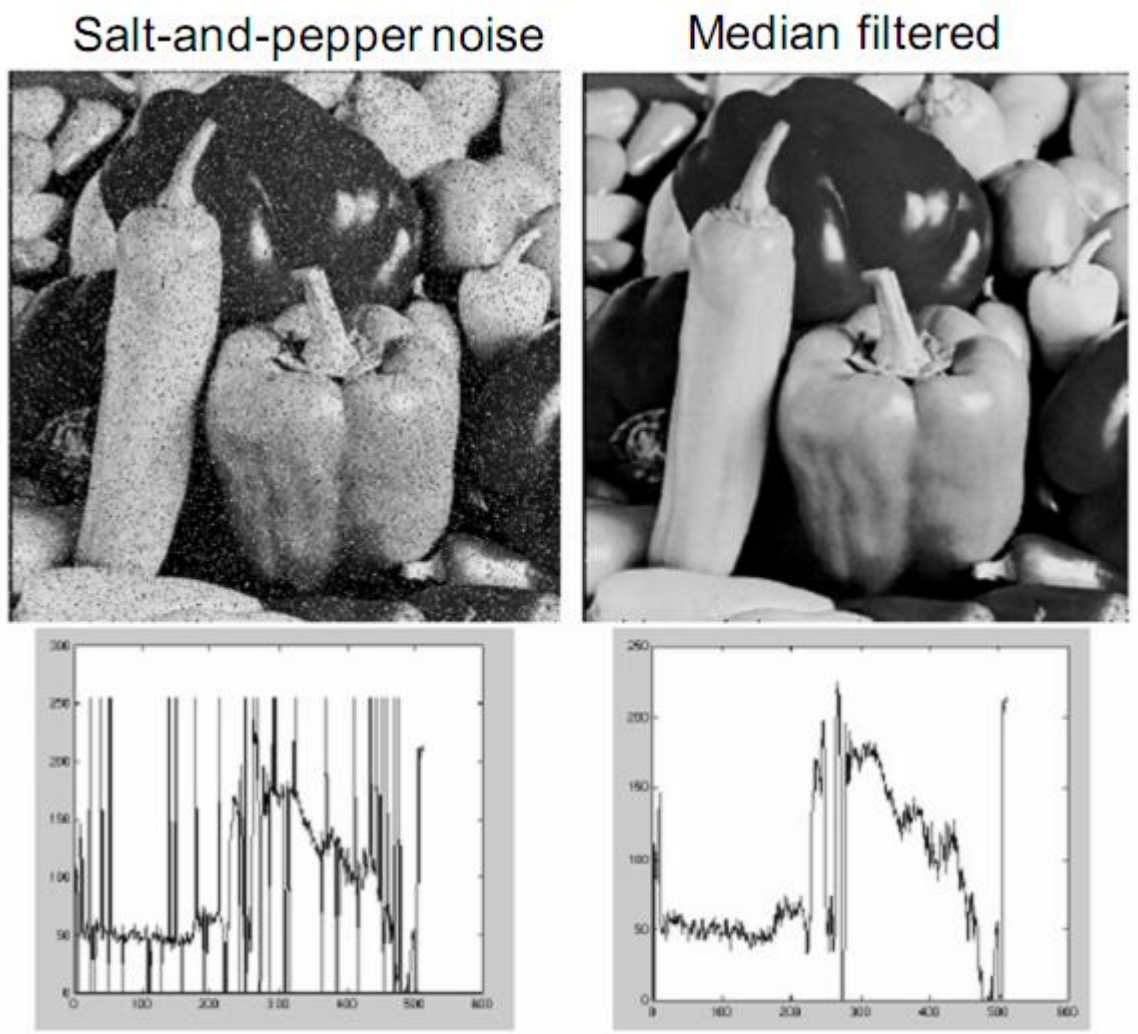


Source: K. Grauman

Медиана сохраняет ступеньки, среднее размывает

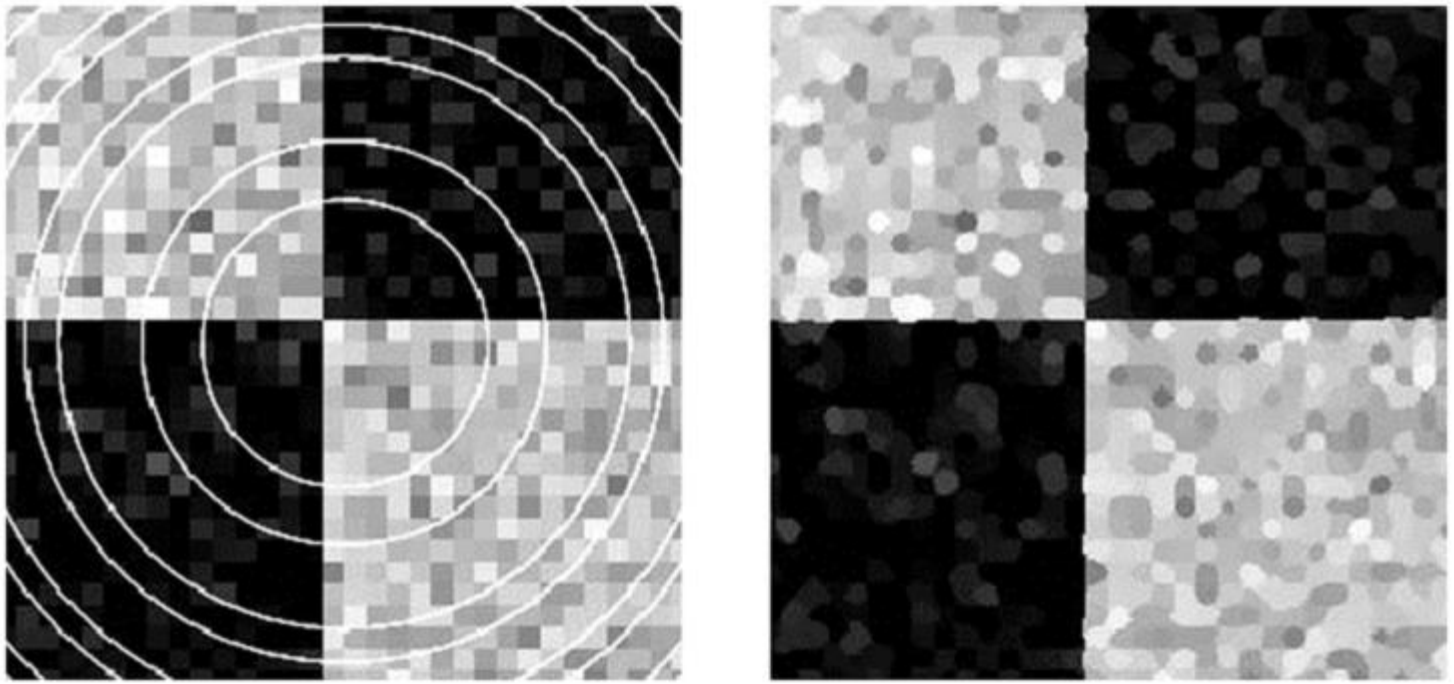
Медианный фильтр на соли-перце

Медиана убирает соль-перец,
Gaussian — нет



Source: M. Hebert

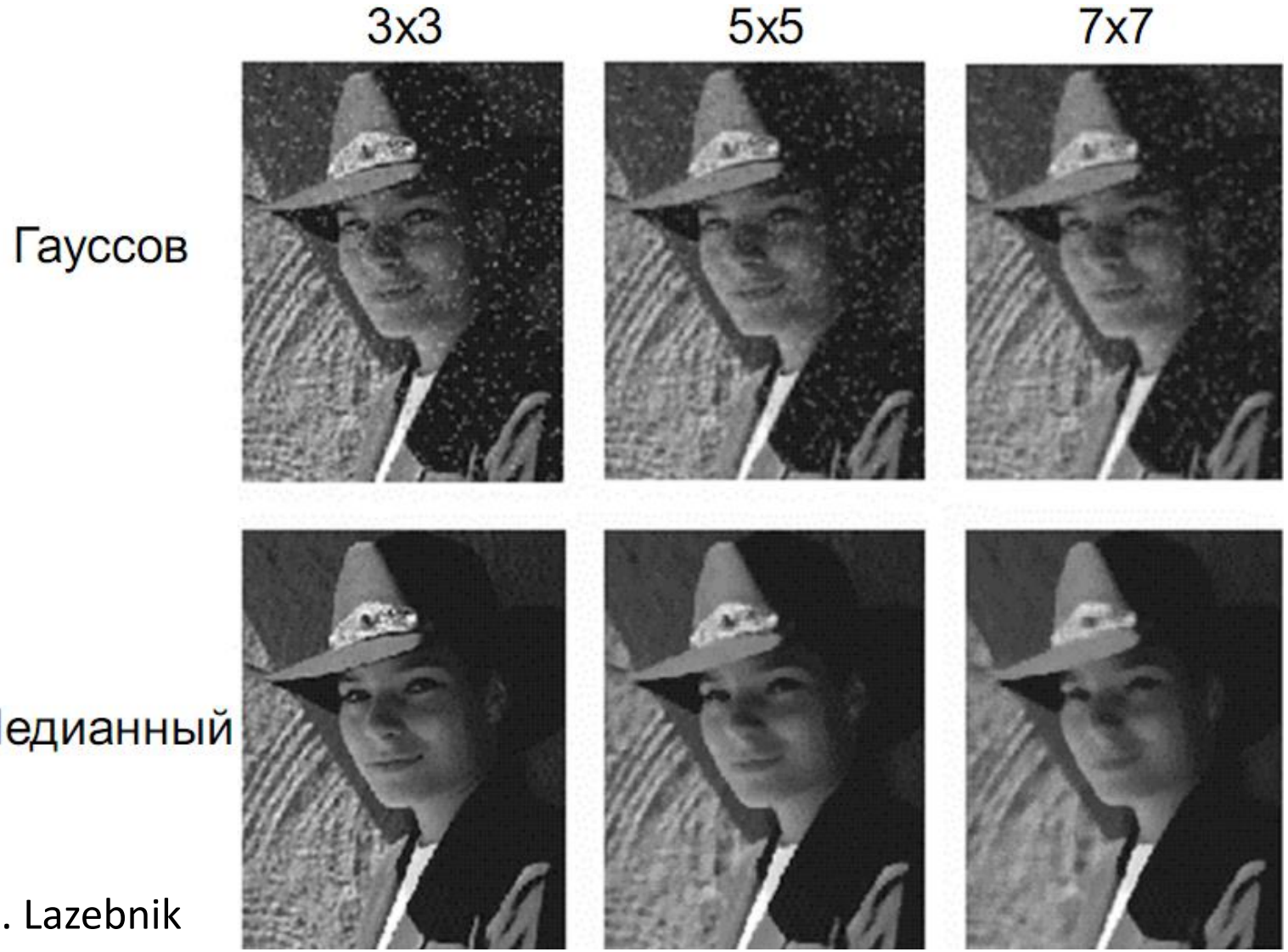
Медианный фильтр на текстуре



Результат применения медианного фильтра с радиусом в 7 пикселей к изображению с шумом и артефактами в виде тонких светлых окружностей

Медиана удаляет тонкие линии

Сравнение 3×3, 5×5, 7×7, Gaussian, Median



Source: S. Lazebnik

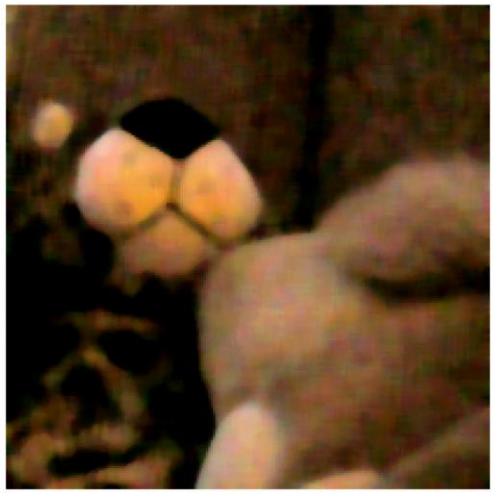
Медианный фильтр: 1px, 3px, 10px



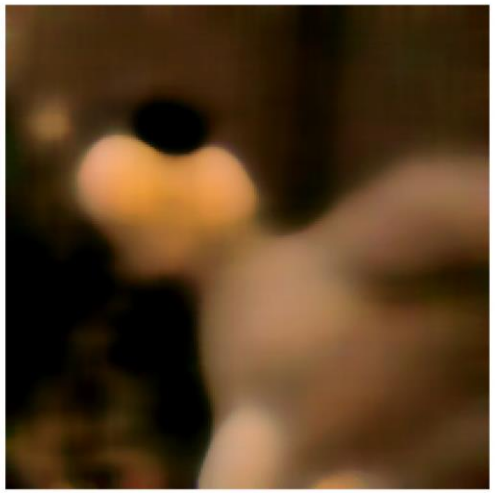
original image



1px median filter



3px median filter



10px median filter

Большой радиус → эффект пластика

Быстрая реализация медианного фильтра

Медианный фильтр считается дольше, чем операция свертки, поскольку требует частичной сортировки массива яркостей окрестных пикселей.

Возможности ускорения:

Через гистограмму окрестности — O(n)

Использовать алгоритмы быстрой сортировки

Конкретная реализация для каждого радиуса (3x3, 5x5)

Не использовать сортировку вообще – считать через гистограмму окрестности точки

Фильтры min/max

Фильтр «**минимум**» – также известный как фильтр **эрозии**, заменяет значение минимальным в окрестности.

Фильтр «**максимум**» – также известный как фильтр **расширения**, заменяет значение максимальным в окрестности.

Это erosion/dilation из лекции 4

Морфология = нелинейная фильтрация

Фильтр выборочной уборки шума

Фильтр выборочной уборки шума – проверяет, не является ли значение экстремумом в окрестности.

Если да, то заменяет значение следующим самым большим (наименьшим) значением.

Фильтр «убрать шахматную доску»

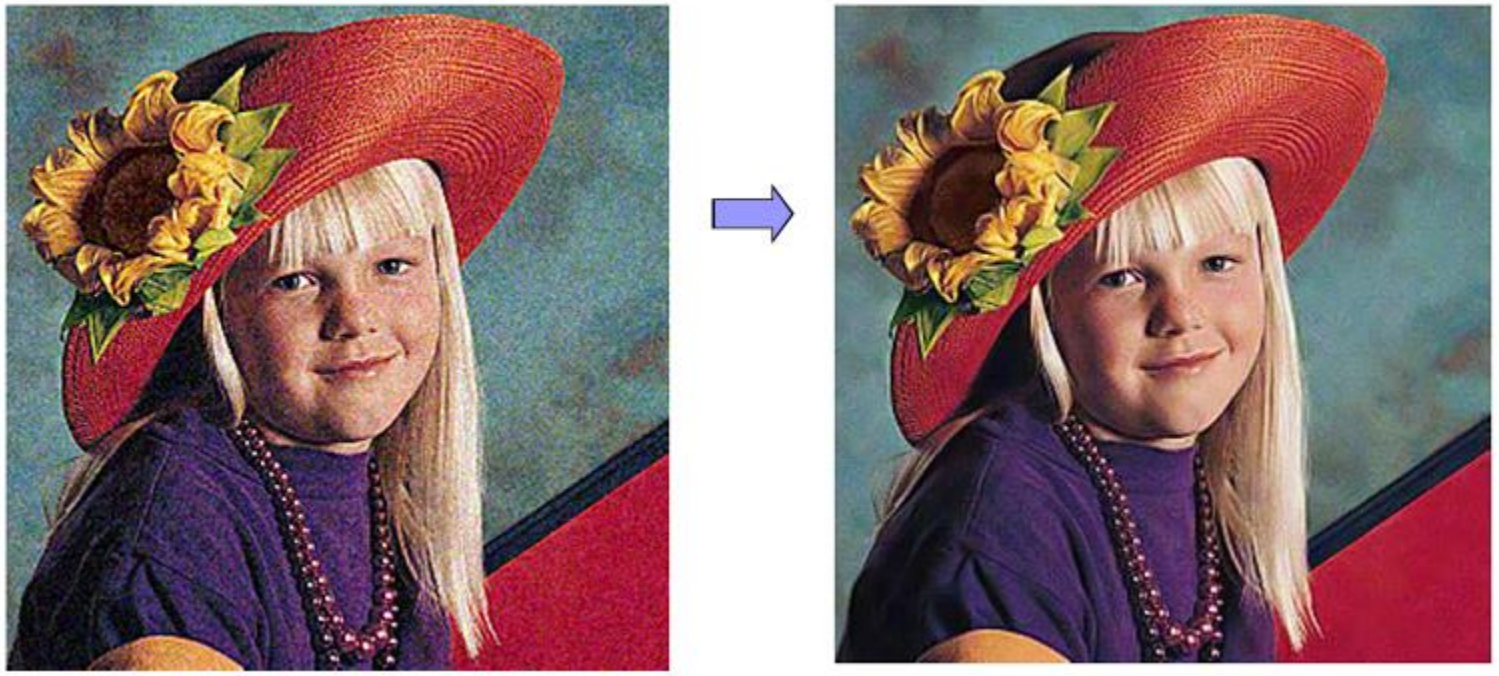
Фильтр «убрать шахматную доску» – сглаживающий фильтр, специально предназначенный для удаления текстур, напоминающих шахматную доску из изображения при сохранении других особенностей.

Это фильтр свёртки с ядром

$$w_{\text{dechecker}} = \begin{pmatrix} 0 & 1/144 & -1/72 & 1/144 & 0 \\ 1/144 & -1/18 & 1/9 & -1/18 & 1/144 \\ -1/72 & 1/9 & 7/9 & 1/9 & -1/72 \\ 1/144 & -1/18 & 1/9 & -1/18 & 1/144 \\ 0 & 1/144 & -1/72 & 1/144 & 0 \end{pmatrix}$$

Относится к группе специализированных фильтров для конкретной текстуры

«Продвинутые» фильтры



Bilateral, guided — сохраняют края. Дорогие

В чем отличие разных фильтров?

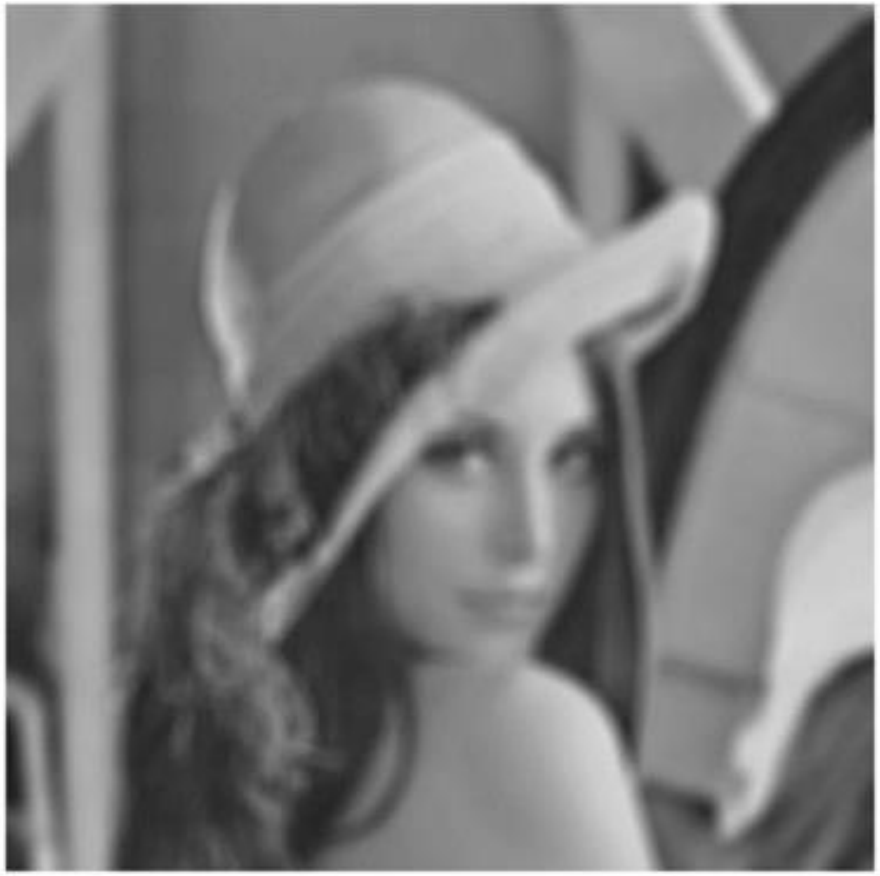
- Box filer (простое размытие) – помимо подавления шума портит резкие границы и размывает мелкие детали изображения
- Gaussian filter – меньше размывает мелкие детали, лучше убирает шум
- Median filter – резких границ не портит, убирает мелкие детали, изображение становится менее естественным
- «Продвинутые» фильтры – лучшее сохранение деталей, меньше размытие. Часто сложны в реализации и очень медленные

Что лучше? – зависит от конкретной задачи

Повышение резкости: постановка

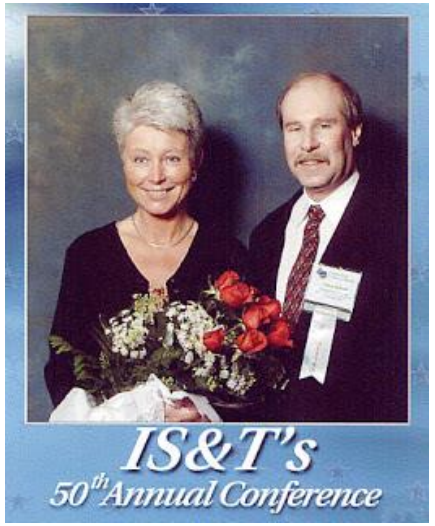
Повышение резкости: постановка задачи

Необходимо повысить резкость (четкость) изображения



Лирическое отступление: Lena (Lenna)







Lena Söderberg, born **Sjööblom** (born 31 March 1951) is a Swedish, as **Lenna Sjööblom**

- Обрезанная (512x512) часть изображения с разворота Playboy, Nov 1972
- Самый популярный, но не первый случай использования Playboy в обработке изображений (первый в 1961)
- Пригласили на 50ую конференцию Society for Imaging Science and Technology (IS&T) in 1997

Этическая дискуссия в CV: использование изображения

Пример повышения резкости

Ядро свертки

$$\frac{1}{10}\cdot\begin{Vmatrix} -1 & -2 & -1 \\ -2 & 22 & -2 \\ -1 & -2 & -1 \end{Vmatrix}$$

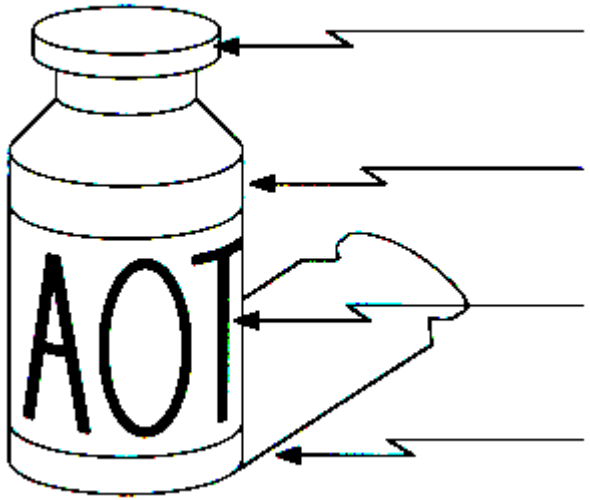




Highboost = original + α·(original − blurred)

Откуда берутся границы

Существует множество причин формирования границ на изображении



Резкое **изменение нормали** поверхности

Резкое **изменение глубины**

Резкое **изменение цвета** поверхности

Резкое **изменение освещенности**

Резкое изменение = «разрыв»

Пространственные фильтры повышения резкости: производные

- достигается пространственным дифференцированием
- основаны на первой и второй производной: 1-я — для контуров, 2-я — для деталей

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

Упрощённая схема (профиль яркости + производные)



2-я производная даёт двойной отклик на ступеньку

Кросс-корреляция с маской первой производной [-1,0,1]

S_1	12	12	12	12	12	24	24	24	24	24
$S_1 * M$	0	0	0	0	12	12	0	0	0	0
S_2	24	24	24	24	24	12	12	12	12	12
$S_2 * M$	0	0	0	0	-12	-12	0	0	0	0
S_3	12	12	12	12	15	18	21	24	24	24
$S_3 * M$	0	0	0	3	6	6	6	3	0	0
S_4	12	12	12	12	24	12	12	12	12	12
$S_4 * M$	0	0	0	12	0	12	0	0	0	0

1-я производная → толстые контуры

Кросс-корреляция с маской второй производной M=[-1,2,-1]

S_1	12	12	12	12	12	24	24	24	24	24
$S_1 * M$	0	0	0	0	-12	12	0	0	0	0
S_2	24	24	24	24	24	12	12	12	12	12
$S_2 * M$	0	0	0	0	12	-12	0	0	0	0
S_3	12	12	12	12	15	18	21	24	24	24
$S_3 * M$	0	0	0	-3	0	0	0	3	0	0
S_4	12	12	12	12	24	12	12	12	12	12
$S_4 * M$	0	0	0	-12	24	-12	0	0	0	0

2-я производная → тонкие контуры + двойной отклик

Первая и вторая производные: свойства

- Первая производная обычно формирует более толстые контуры
- Вторая производная формирует больший по величине отклик на мелкие детали (на отдельных точках и на тонких линиях)
- Отклик на ступенчатое изменение яркости у первой производной, как правило, выше, чем у второй
- Для участков постоянного изменения яркости вторая производная формирует двойной отклик
- При одинаковых амплитудах изменения сигнала вторая производная формирует более сильный отклик для тонкой линии, чем для ступеньки, а для отдельной точки - более сильный, чем для линии.

Применение производных

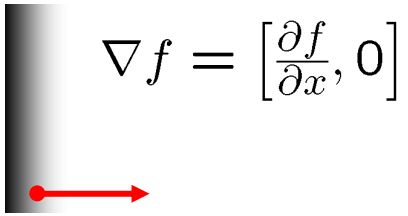
Первая производная – в задачах выделения контуров

Вторая производная – в задачах улучшения изображений (Эффект двойных контуров – помеха в задаче обнаружения контуров)

Объединение методов — стандарт

Градиент изображения

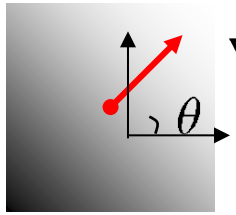
Градиент изображения: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$



$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$



$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$



$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

Градиент направлен в сторону наибольшего изменения интенсивности

Направления градиента задается как: $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$

- Как направление градиента соответствует направлению края?
- *Сила края* задается величиной (нормой) градиента:

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Source: Steve Seitz

Лапласиан

Лапласиан – простейший изотропный оператор

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

Изотропные фильтры – отклик не зависит от направления неоднородностей

Изотропные фильтры – инвариантность к повороту

Дискретный вид лапласиана

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y)$$

Лапласианы (маски)

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

a)

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

Инвариантны относительно поворота на углы, кратные 90º и 45 º

Восстановление фона с сохранением резкости

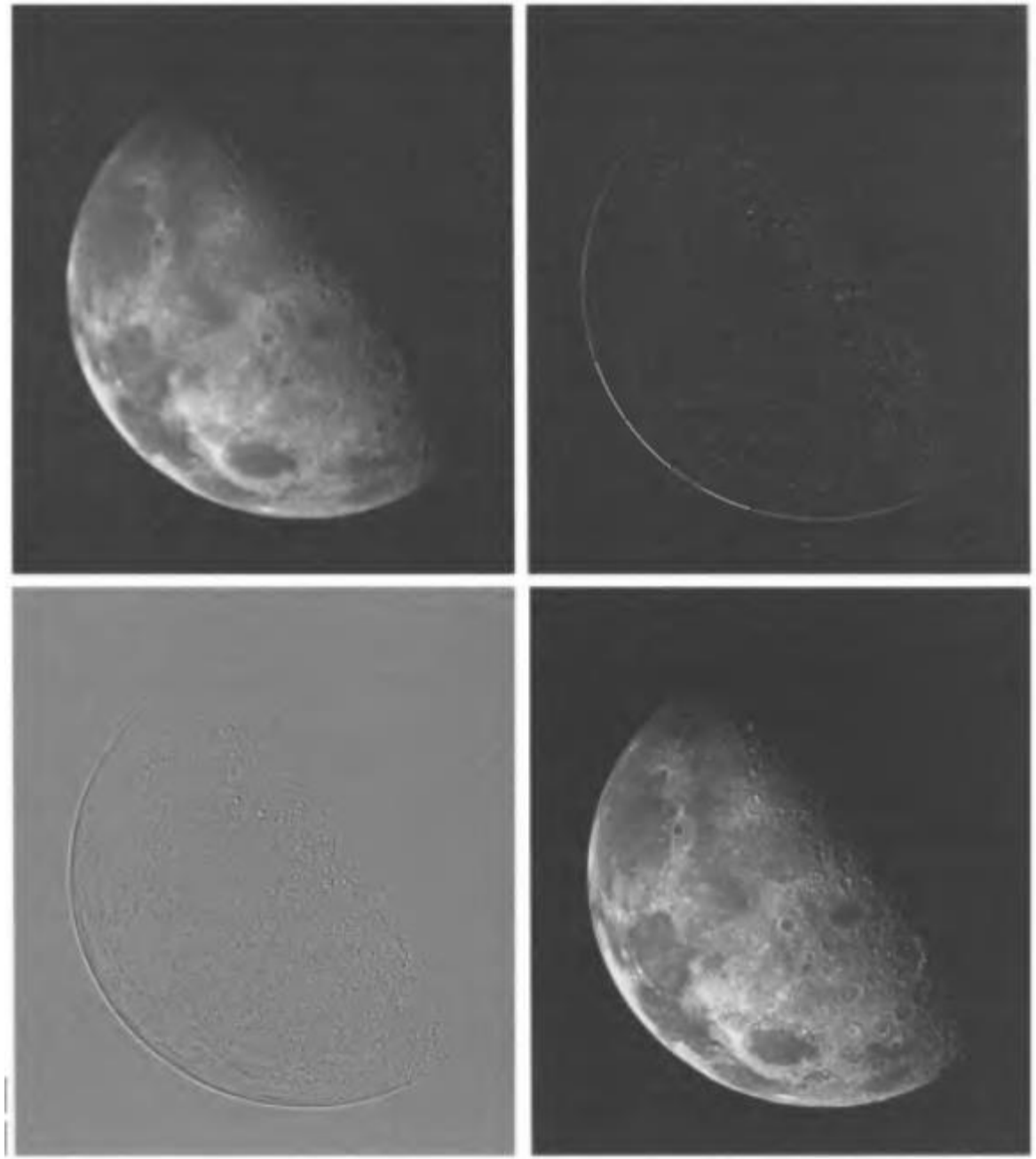
Сложить исходное изображение и лапласиан

$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y), & \text{если } w(0,0)<0 \\ f(x,y)+\nabla^2 f(x,y), & \text{если } w(0,0)\geq 0 \end{cases}$$

w(0,0) – центральное значение маски лапласиана

Знак зависит от центрального элемента маски

Повышение резкости Луны лапласианом



Лапласиан выделяет мелкие детали

Упрощение: составная маска лапласиан

$$g(x,y) =$$
$$f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] + 4f(x,y) =$$
$$5f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)]$$

0	-1	0
-1	5	-1
0	-1	0

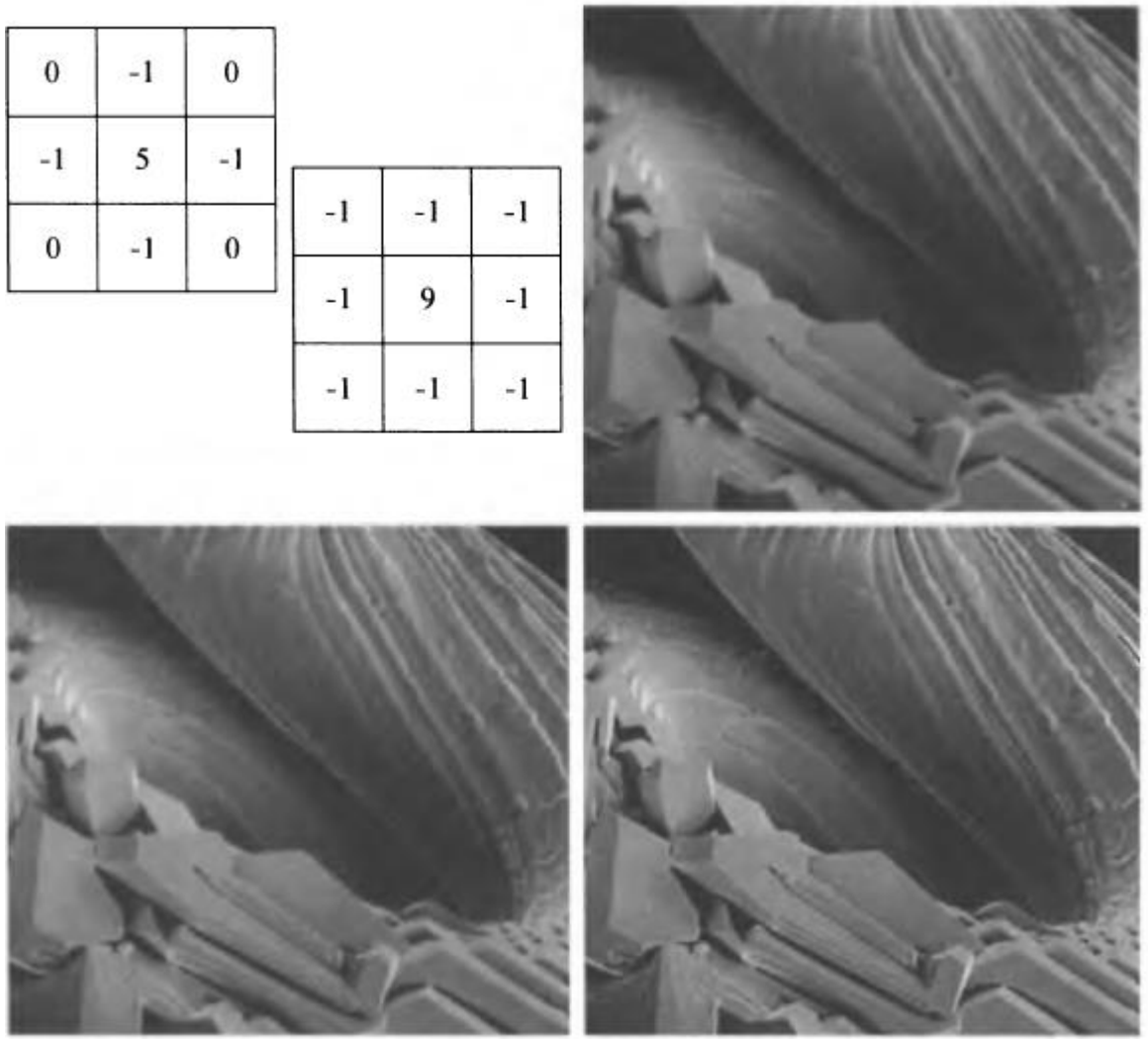
-1	-1	-1
-1	9	-1
-1	-1	-1

Одна свёртка вместо двух

Вольфрамовая нить накала

0	-1	0
-1	5	-1
0	-1	0

-1	-1	-1
-1	9	-1
-1	-1	-1



Диагональная маска даёт более резкий результат

Что теряется при сглаживании?



Добавим дополнительно высокие частоты:



Source: S. Lazebnik

Нерезкое маскирование
Фильтрация с подъемом высоких частот

Для повышения резкости

Применяется в полиграфии

$$f_{mask}(x,y) = f(x,y) - \overline{f}(x,y)$$

$$f_{hb}(x,y) = f(x,y) + Af_{mask}(x,y)$$

Создание нерезкого изображения

Вычитание из изображения его расфокусированной копии (результат - маска)

Прибавление маски к исходному изображению

A = 1 → нерезкое маскирование, A > 1 → фильтрация с подъемом высоких частот

Фильтры с подъемом высоких частот на базе лапласиана

0	-1	0
-1	A+4	-1
0	-1	0

-1	-1	-1
-1	A+8	-1
-1	-1	-1

Типичное приложение фильтрации с подъемом высоких частот

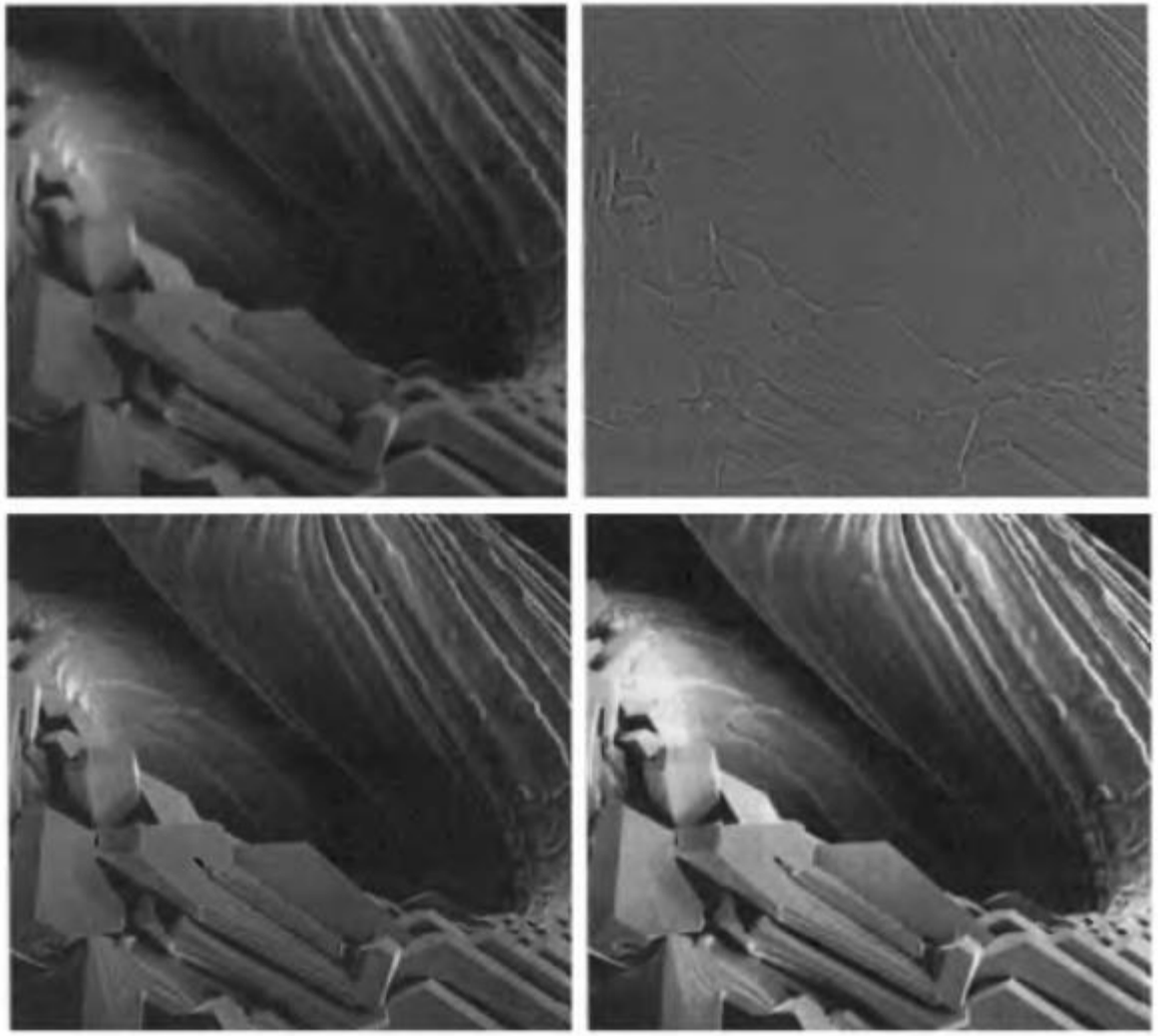
Исходное изображение темнее чем требуется

а) исходное

б) маска

в) A=1

г) A=1.7



Увеличение среднего уровня яркости

$$f_{\text{hb}}(x,y)=\begin{cases} Af(x,y)-\nabla^2 f(x,y), & \text{если } w(0,0)<0 \\ Af(x,y)+\nabla^2 f(x,y), & \text{если } w(0,0)\geq 0 \end{cases}$$

Использование первых производных: градиент

$$\nabla \mathrm{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

$$|\nabla \mathbf{f}| = \left[G_x^2 + G_y^2 \right]^{1/2} = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

$|\nabla f| \approx |G_x| + |G_y|$ — быстрая аппроксимация

Оператор Робертса (1965)

$$G_x = (z_9 - z_5) \quad \text{и} \quad G_y = (z_8 - z_6)$$

$$\nabla f = \left[(z_9 - z_5)^2 + (z_8 - z_6)^2 \right]^{1/2}$$

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

Первый оператор краёв

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

-1	0
0	1

0	-1
1	0

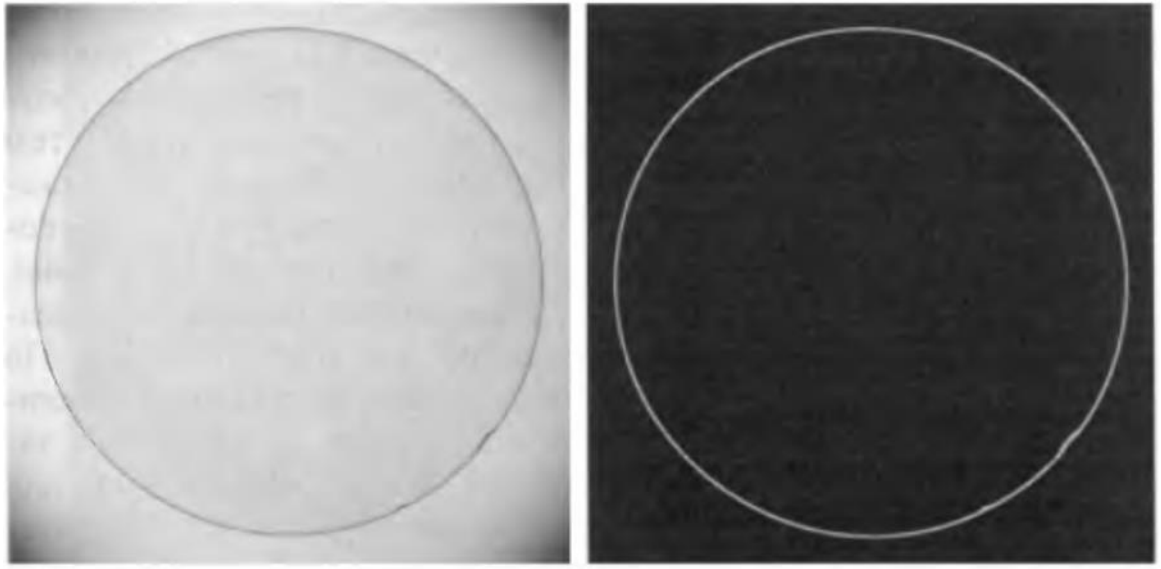
Оператор Собела

$$\nabla f \approx \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right|.$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1



Взвешенная сумма — устойчивее к шуму

Оператор Превитта

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

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

Похож на Собела, но с равными весами

Примеры: Робертс, Превитт, Собел



Робертса



Превитт



Собеля

Свойства масок производных

Для получения высокого отклика в областях сигнала с высоким контрастом элементы масок производных имеют **разные знаки**

Сумма элементов масок производных **равна 0**, поэтому в областях сигнала с постоянными значениями маски дают нулевой результат

Точкам **высокого контраста** соответствуют **большие абсолютные** значения отклика масок **первых производных**

Среди значений отклика масок **второй производной** точкам **высокого контраста** соответствуют точки **перехода через 0**

Свойства масок производных

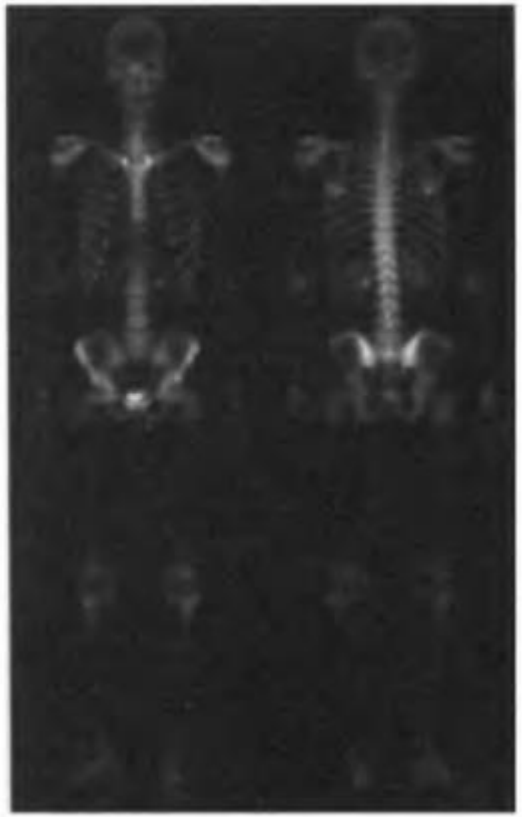
Элементы сглаживающих масок имеют **положительные значения** и их **сумма равна 1**

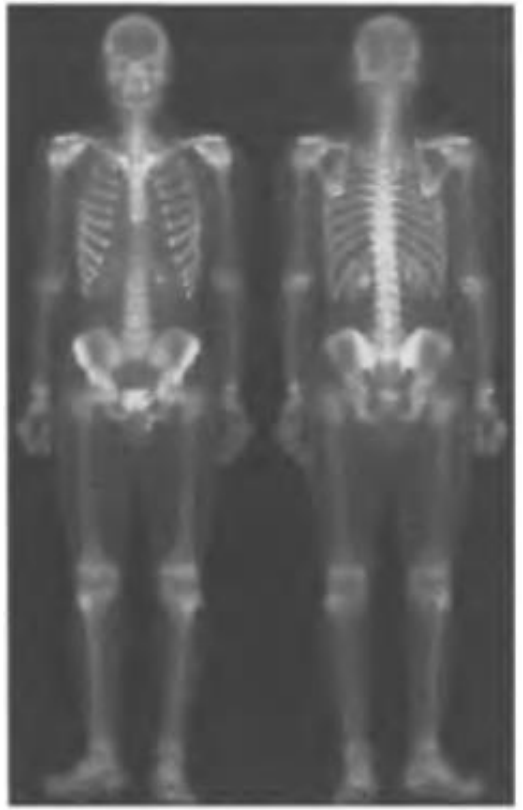
Отклик масок **на областях постоянных** значений **совпадает с входными** значениями

Степень сглаживания и снижения шума **пропорциональна размеру** маски

Ступенчатые края размываются **пропорционально** размеру маски

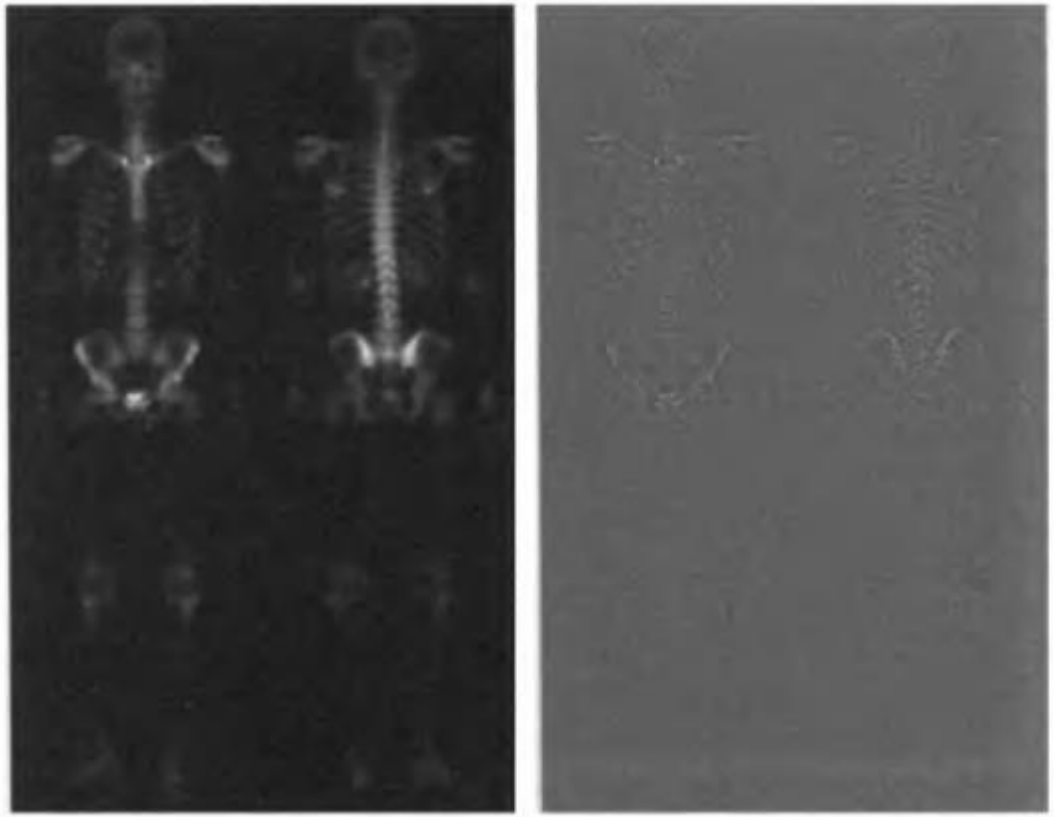
Было – будет (гамма-снимок)





Пример комбинирования методов

Комбинирование: лапласиан



1 2

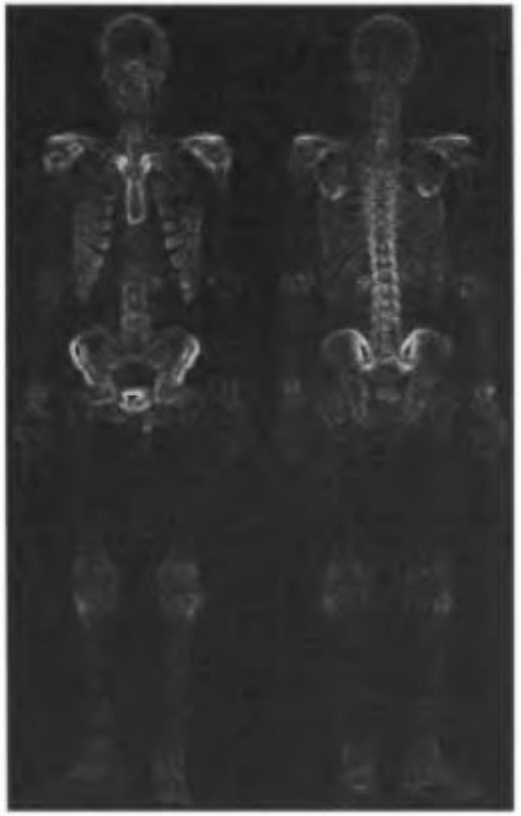
Применение Лапласиана для выявления мелких деталей

Комбинирование: сложение + Собел

Повышение резкости сложением двух предыдущих изображений



3

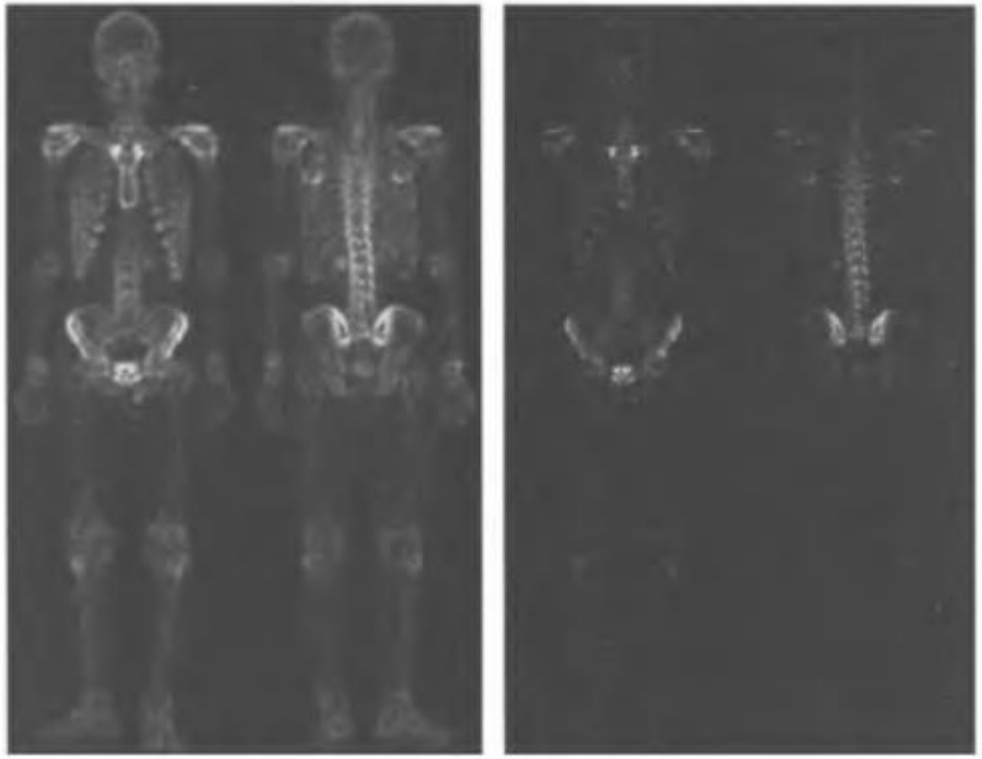


4

Применение оператора Собела – для улучшения выступающих краев

Комбинирование: перемножение

Сглаживание градиента усредняющим фильтром 5x5



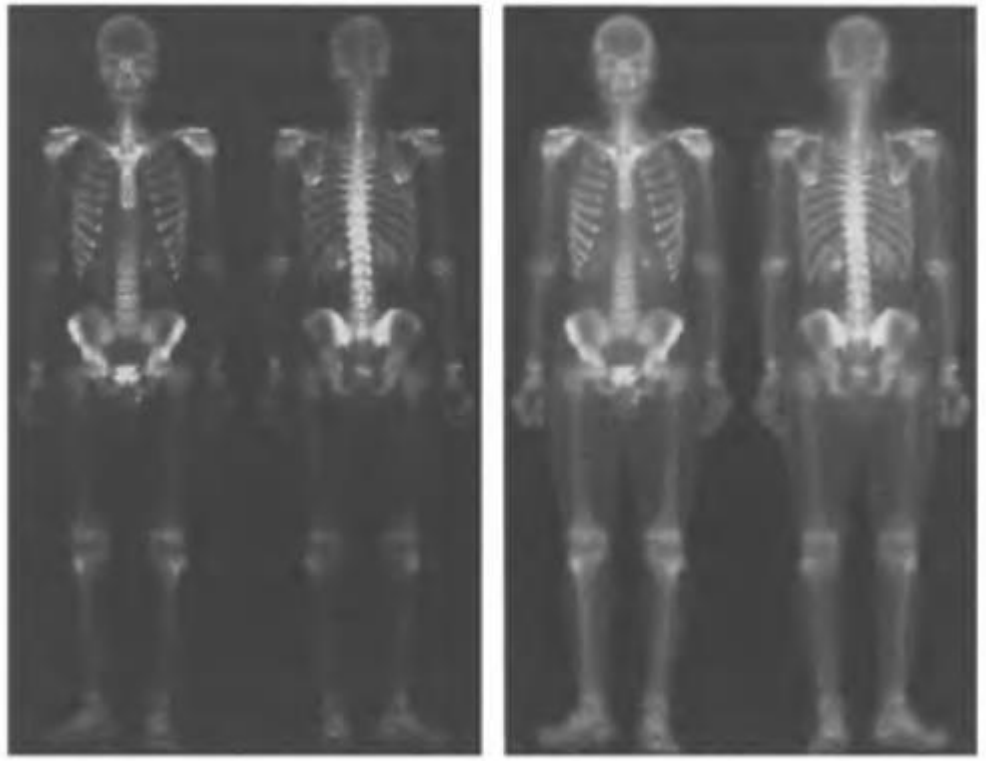
5

6

Перемножение изображений 3 и 5 (с повышенной резкостью и сглаженного)

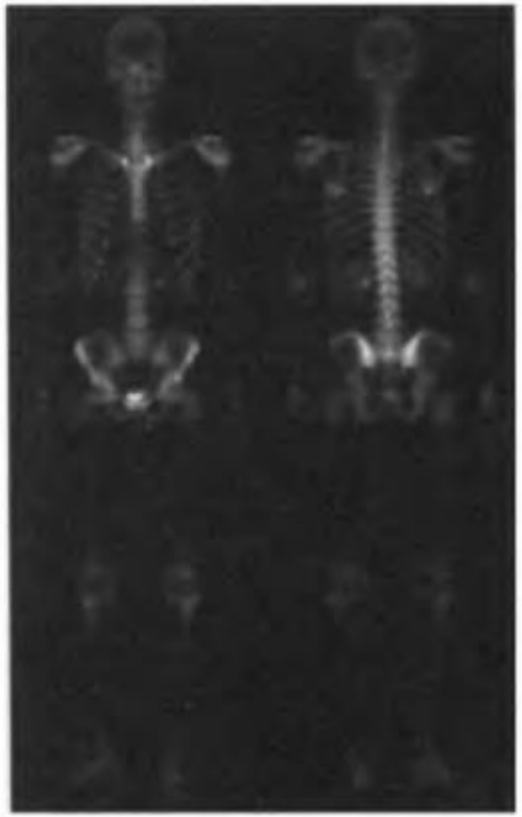
Комбинирование: сложение + гамма

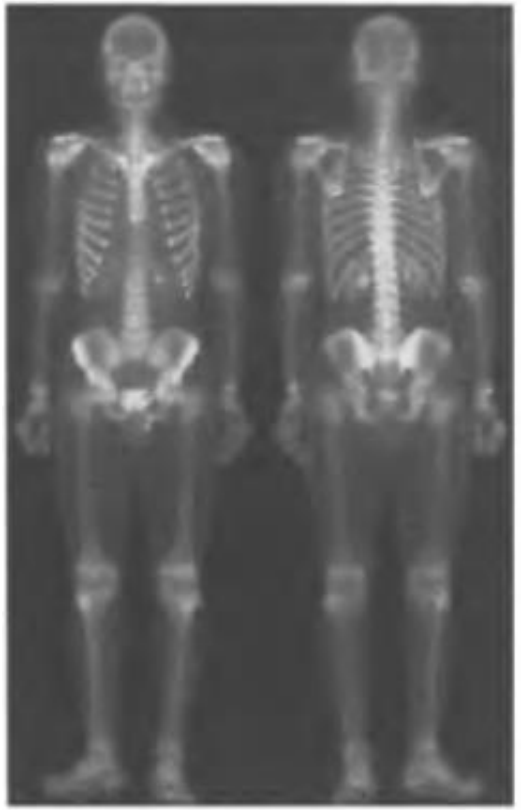
Повышение резкости сложением изображений 1 и 6



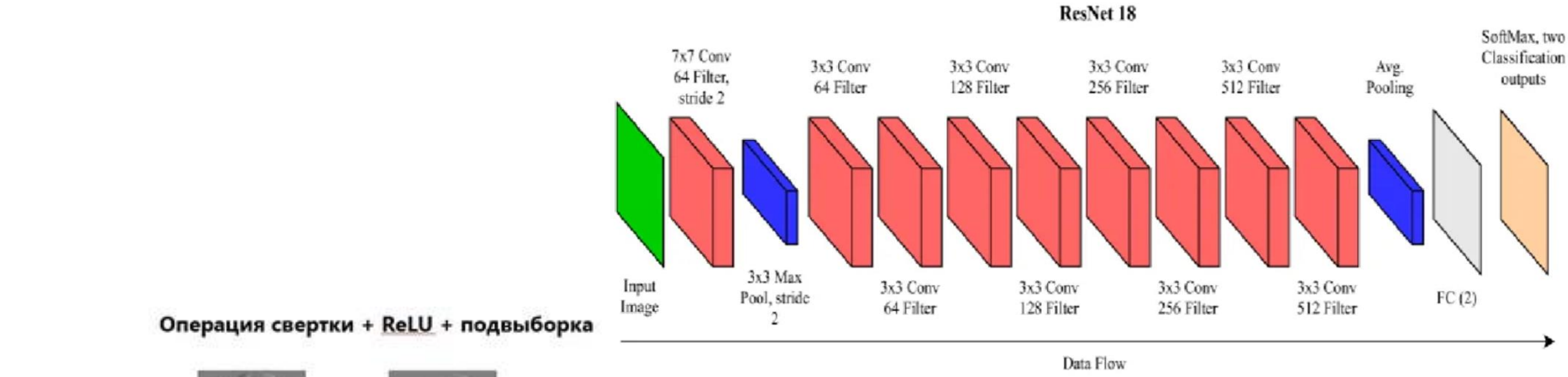
Применение градационной γ-коррекции (γ=0,5 c=1)

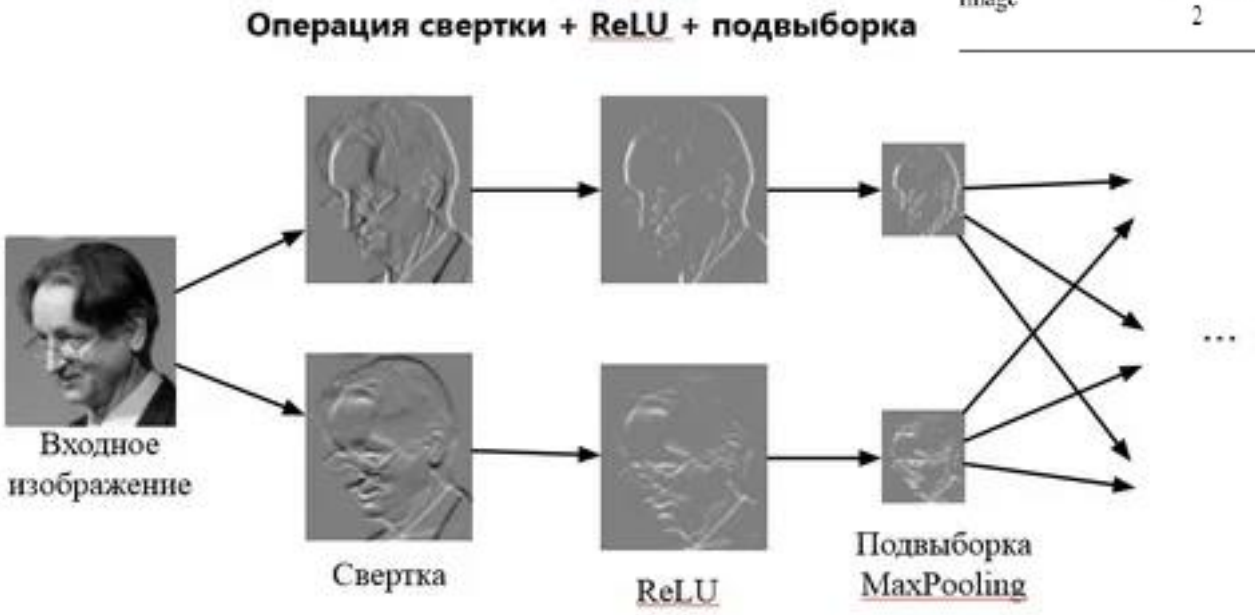
Было – стало





Визуализация фильтров первого свёрточного слоя в предобученной ResNet-18 (Собель)





Почему они похожи на фильтры Собела/Габора?

- Сходство с операторами выделения границ (Sobel, Габора) объясняется эволюцией свёрточных сетей. Первые слои сети учатся выделять базовые визуальные признаки — такие как края, границы, перепады яркости
  - Фильтры Собела акцентируют внимание на вертикальных и горизонтальных перепадах яркости, эффективно вычисляя приближённое значение производной изображения
  - Фильтры Габора объединяют гауссовский шум и выделение границ, что делает их более устойчивыми к повороту и изменению освещения

Таким образом, предобученная ResNet-18 на раннем этапе обучения подтягивает классические признаки, чтобы эффективно решать задачу классификации