

Пространственная обработка изображений

Лекция 5

Обработка изображений

- В пространственной области
- В частотной области

Пространственная обработка

- Яркостные преобразования
 - Оперируют отдельными пикселями
 - Управляют контрастом и «цветностью»
- Пространственная фильтрация
 - Оперирует окрестностью точки
 - Управляет резкостью

Процессы пространственной обработки

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

- Одно изображение или набор изображений
- Окрестности разной формы и разного размера

Арифметические операции

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

for all pixel positions x, y :

$$\text{out}[x, y] = \text{func}(\text{image1}[x, y], \text{image2}[x, y])$$

Addition / Blending

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



$$g(x, y) = \alpha f_1(x, y) + \beta f_2(x, y)$$

Subtraction

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



-



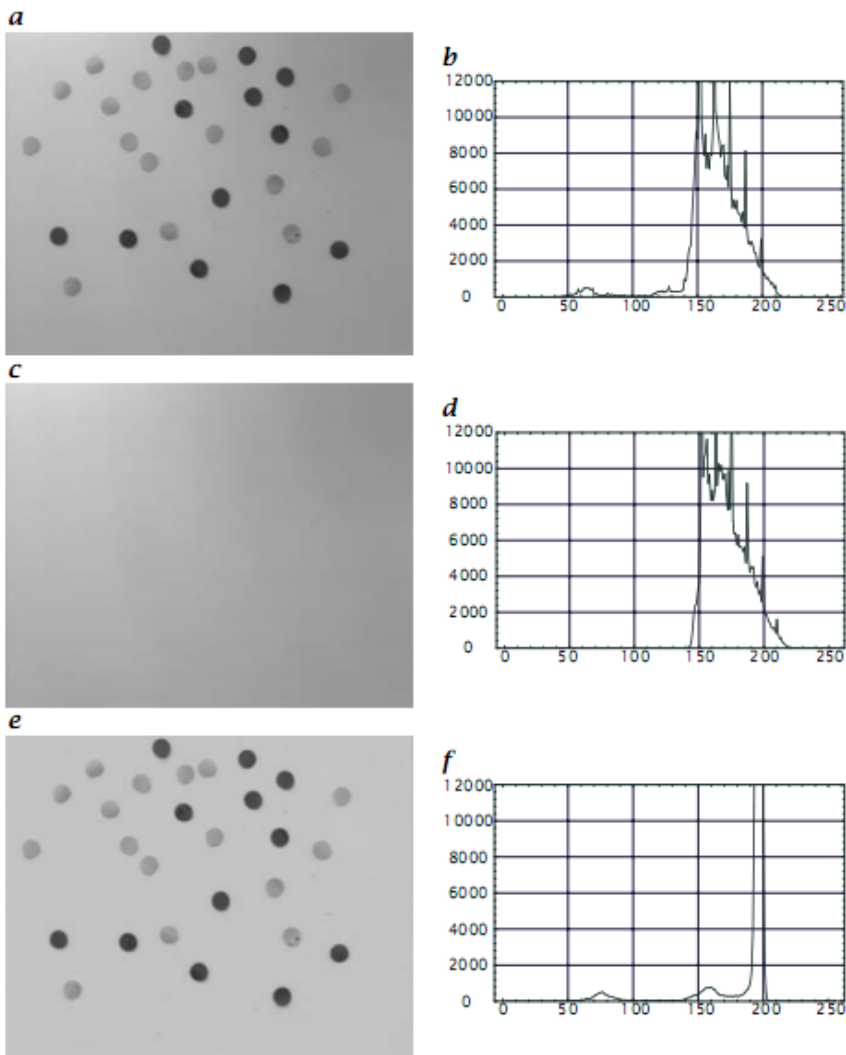
=



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

Коррекция неоднородной освещенности делением изображения на опорное

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



Виды шума



Гауссов



Импульсный (соль.перец)

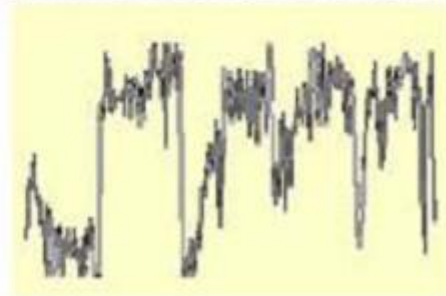


смешанный



Гауссов шум

Image
Noise



Шум
некоррелирован и
имеет нулевое
среднее

$$f(x, y) = \overbrace{\hat{f}(x, y)}^{\text{Ideal Image}} + \overbrace{\eta(x, y)}^{\text{Noise process}}$$

Gaussian i.i.d. ("white") noise:
 $\eta(x, y) \sim \mathcal{N}(\mu, \sigma)$

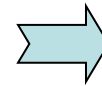
Source: M. Hebert

Усреднение кадров

10 зашумленных изображений



+ +



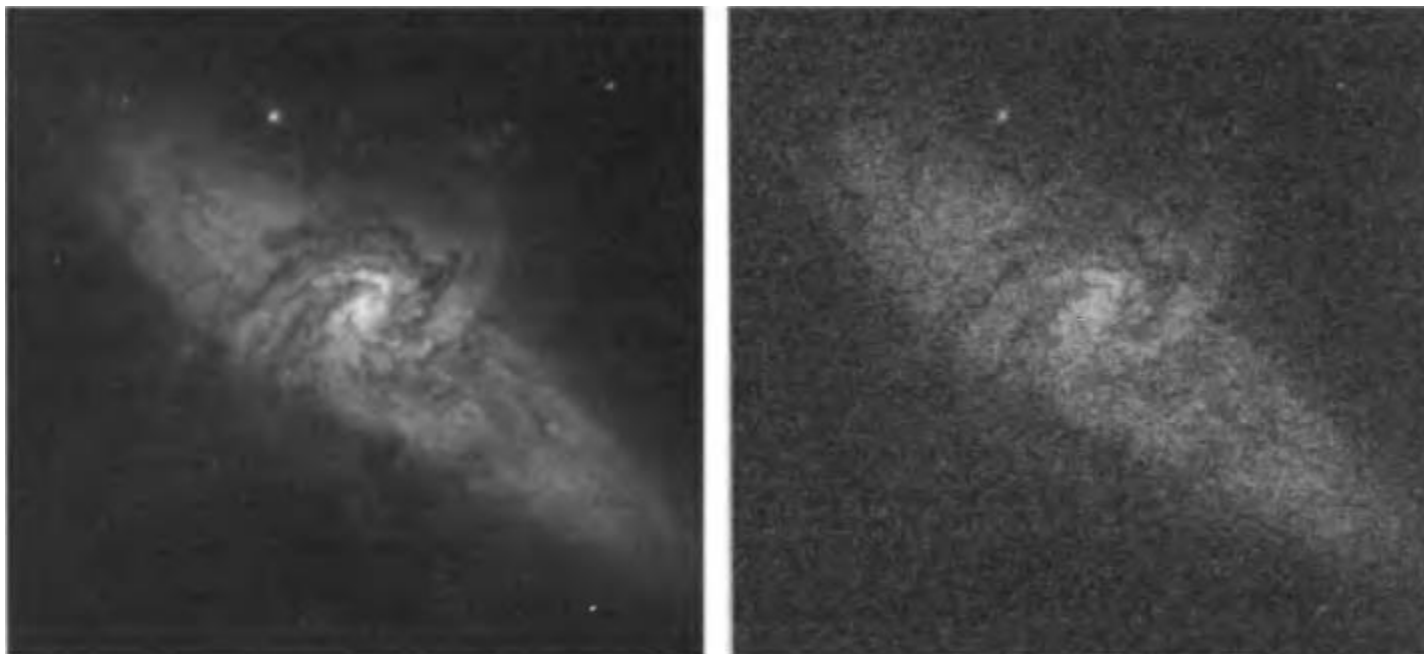
Усреднение по 10 изображениям



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

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

Исходное и шум



Результаты усреднения

K=8



K=16



K=64



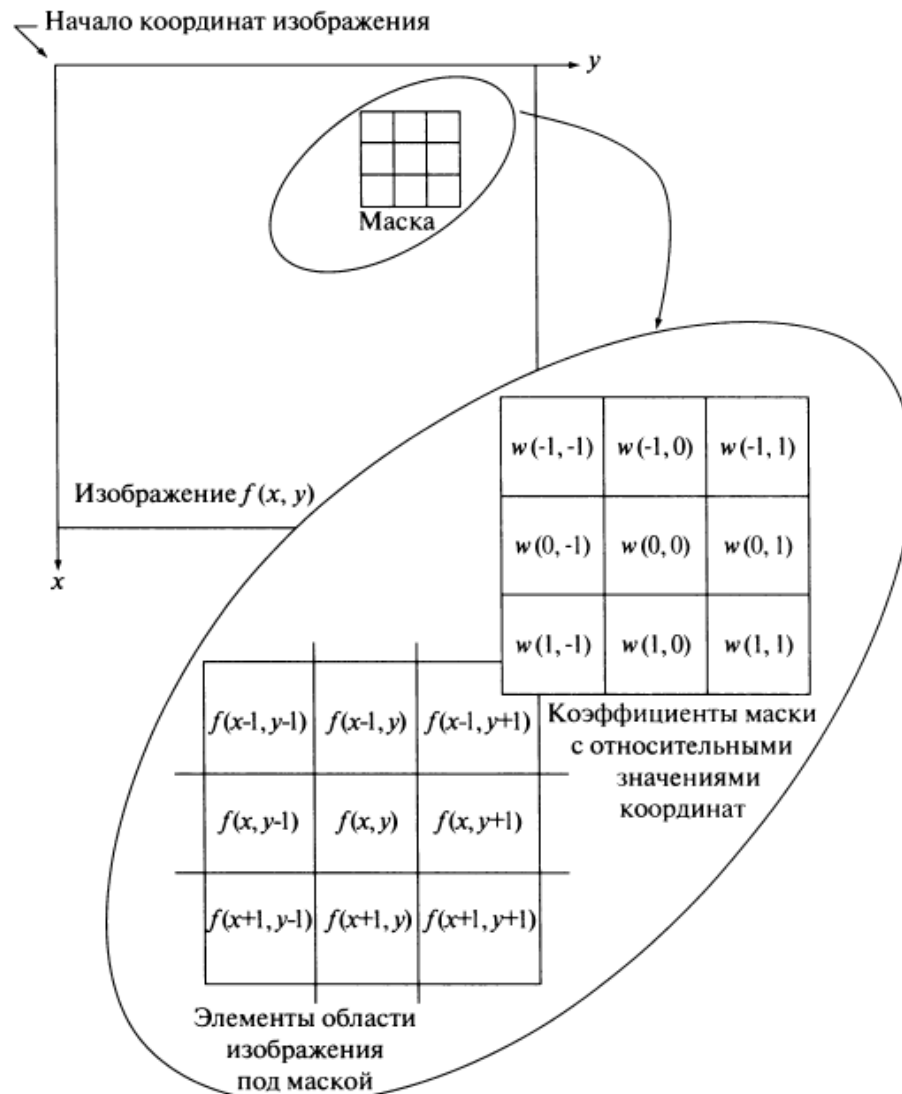
K=128



Схема пространственной фильтрации

Фильтр задается матрицей

- фильтр
- маска
- ядро
- шаблон
- окно
- оператор



- 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

Фильтрация

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

корреляция

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

свертка

Свертка маски с изображением

Основные свойства

- 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

Свойства

- Коммутативность: $a * b = b * a$
 - Нет никакой разницы между изображением и ядром фильтра
- Ассоциативность: $a * (b * c) = (a * b) * c$
 - Последовательное применение фильтров: $((a * b_1) * b_2) * b_3$ эквивалентно применению такого фильтра: $a * (b_1 * b_2 * b_3)$
- 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: $e = [\dots, 0, 0, 1, 0, 0, \dots]$, $a * e = a$

Простейшие фильтры



Original

0	0	0
0	0	1
0	0	0

?

Source: D. Lowe

Простейшие фильтры



Original

0	0	0
0	0	1
0	0	0



Shifted left
By 1 pixel

Source: D. Lowe

Простейшие фильтры



Original

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

?

Source: D. Lowe

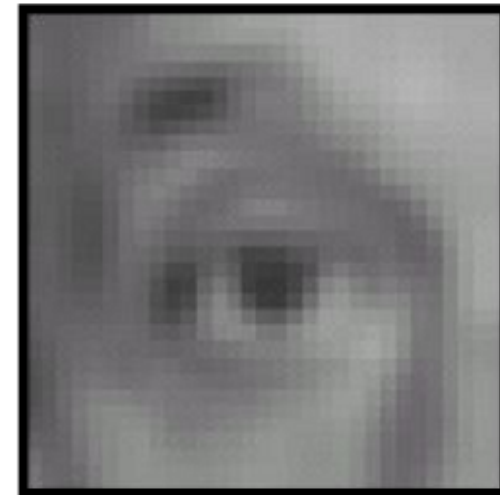
Простейшие фильтры



Original

$$\frac{1}{9}$$

1	1	1
1	1	1
1	1	1



Blur (with a
box filter)

Source: D. Lowe

Сглаживающие пространственные фильтры

Применяются для:

расфокусировки изображения и подавления шума

Выделяют:

линейные и нелинейные

Линейные сглаживающие фильтры

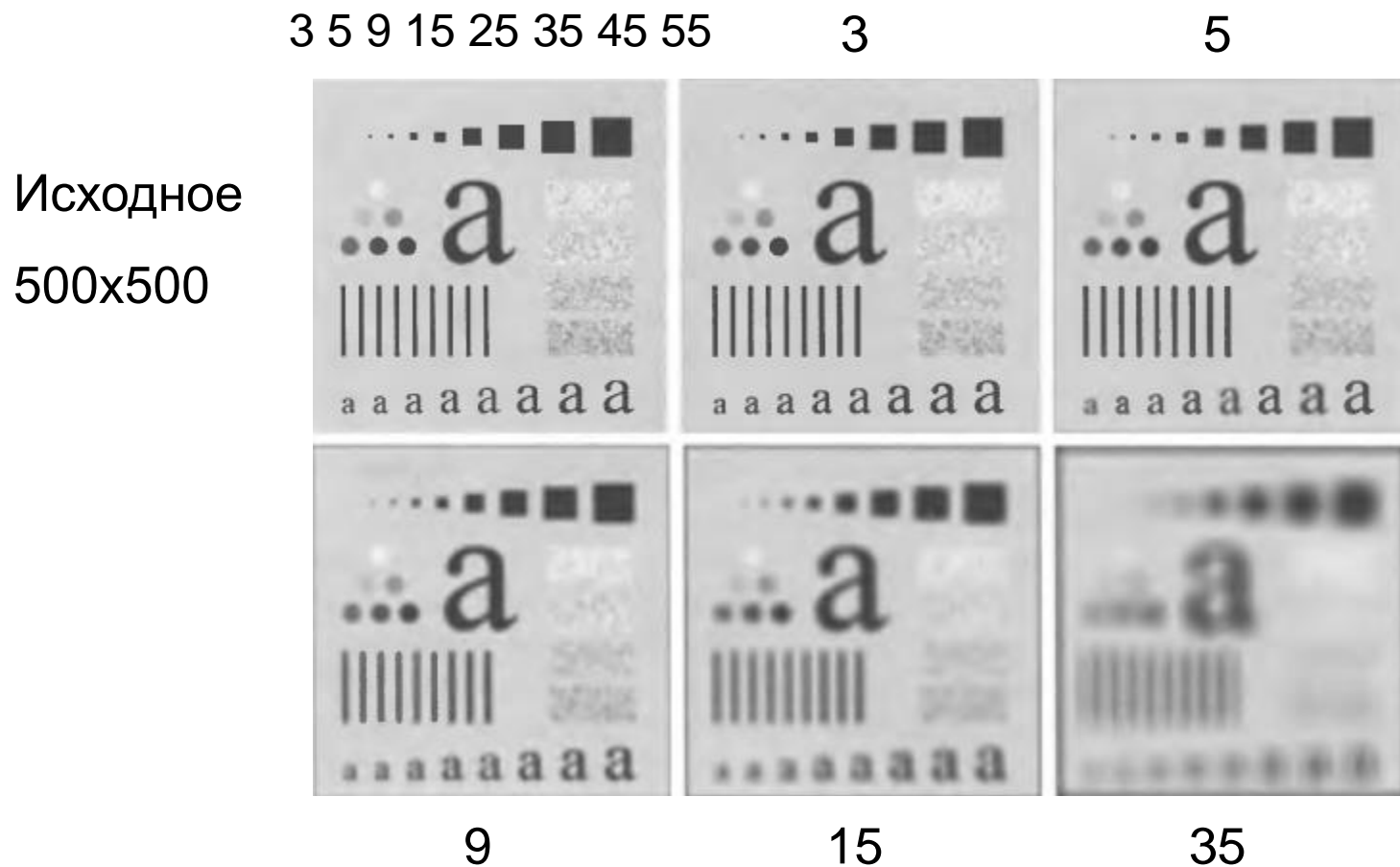
Низкочастотные

 $\frac{1}{9} \times$

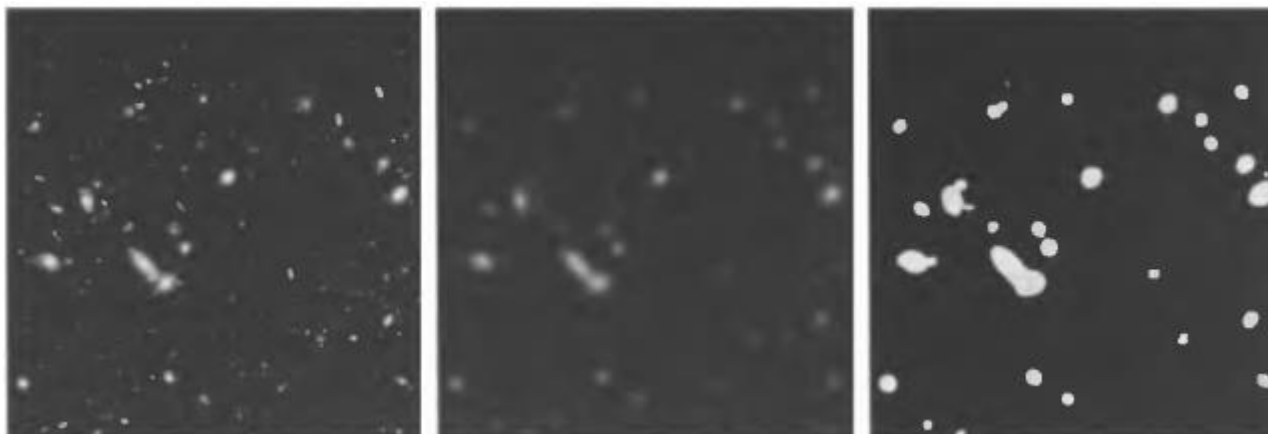
1	1	1
1	1	1
1	1	1

Однородный усредняющий фильтр

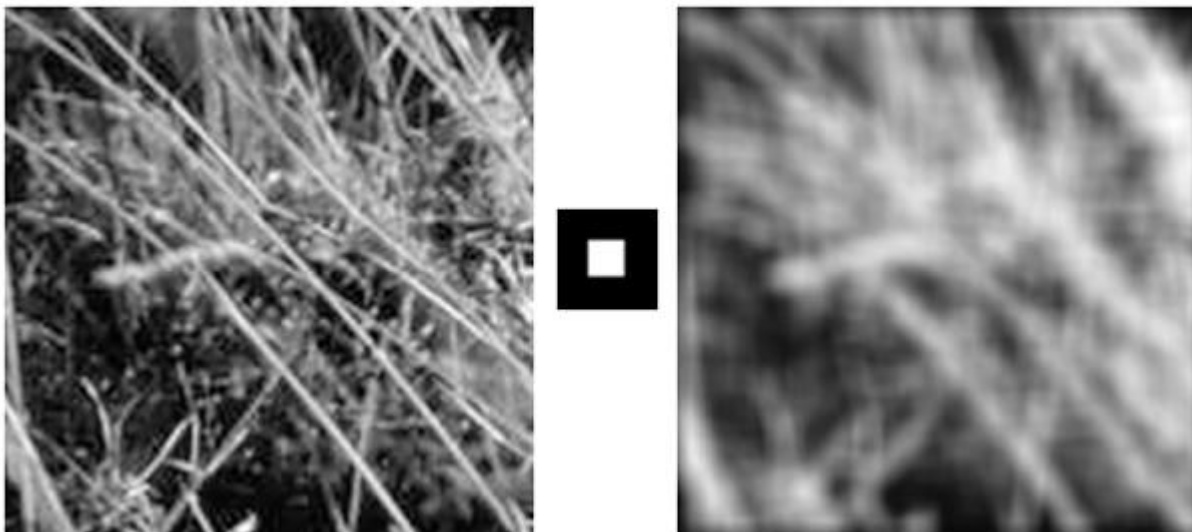
Результаты сглаживания усредняющим фильтром



Сглаживающая маска + пороговое обнаружение

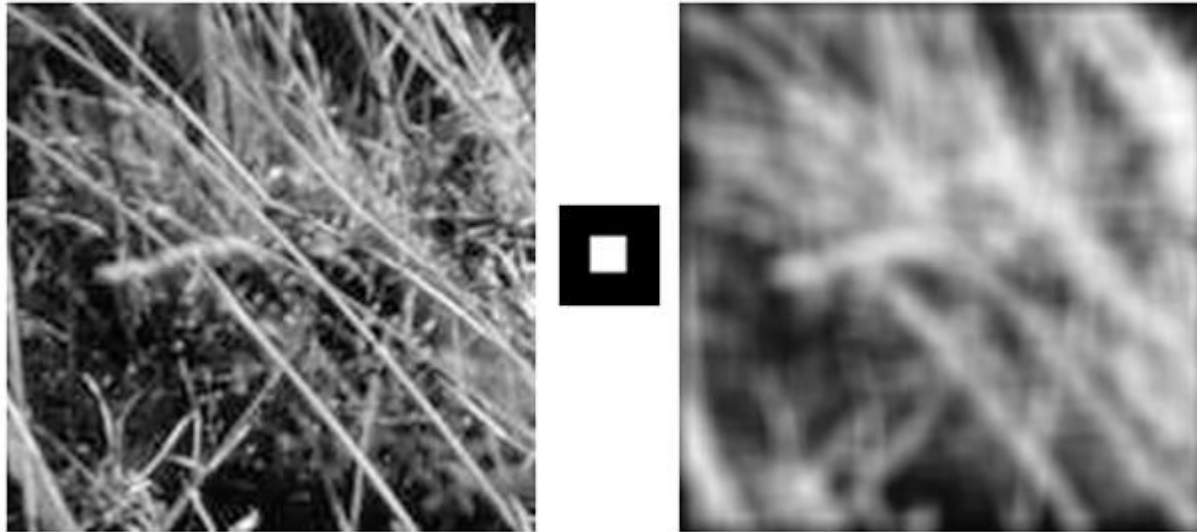


Сглаживание с box-фильтром



Что вы видите?

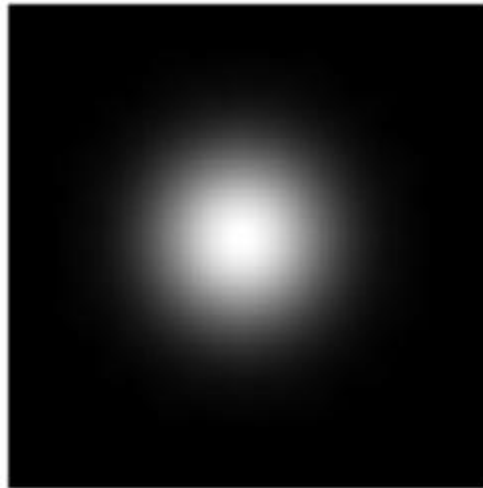
Сглаживание с box-фильтром



При сглаживании с box-фильтром на изображении могут образовываться паразитные линии

Сглаживание

- 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:



fuzzy blob

Фильтр взвешенного среднего

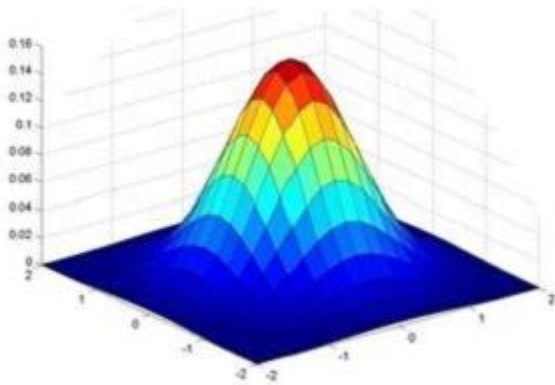
$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x+s, y+t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)}$$

 $\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

Гауссов фильтр

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

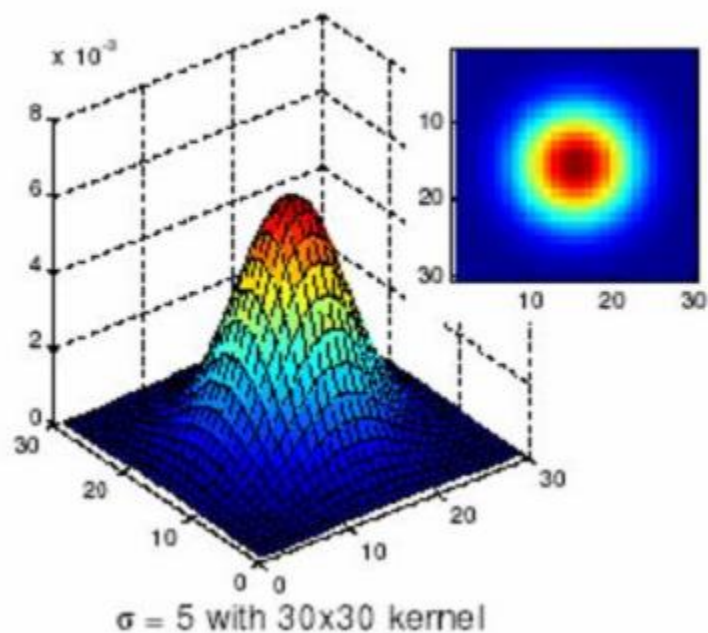
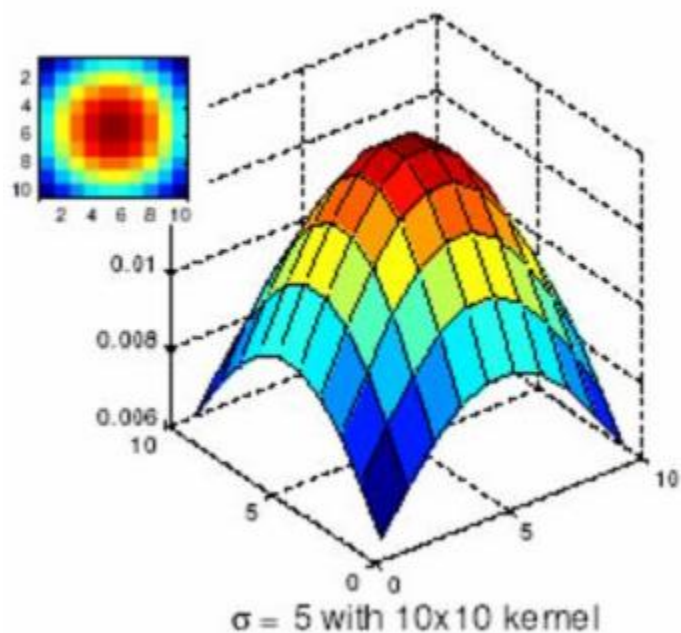


0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, $\sigma = 1$

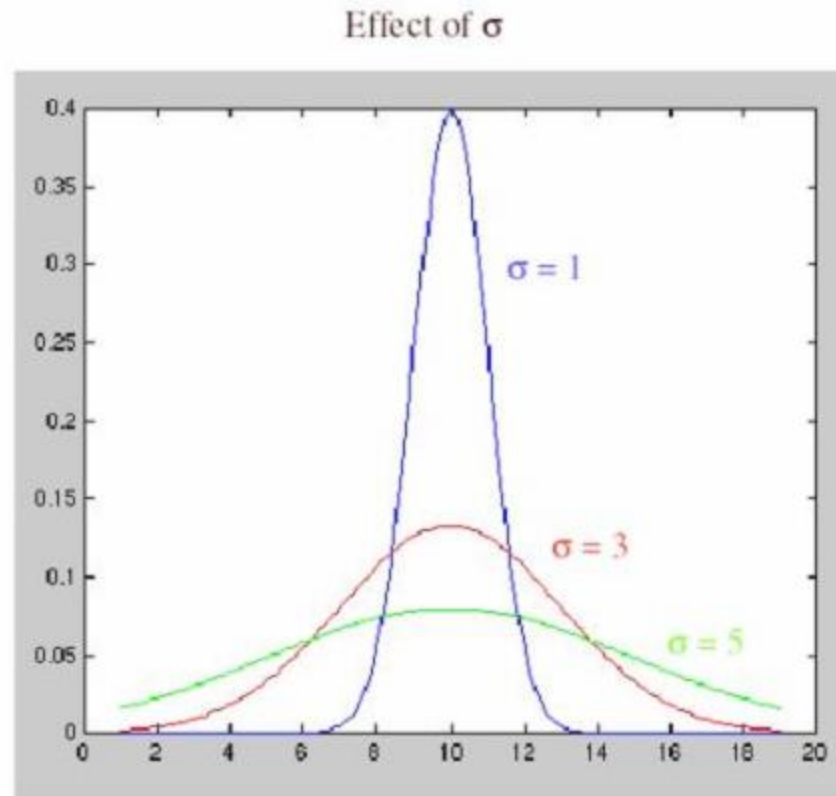
Source: C. Rasmussen

Выбор размера ядра



Source: K. Grauman

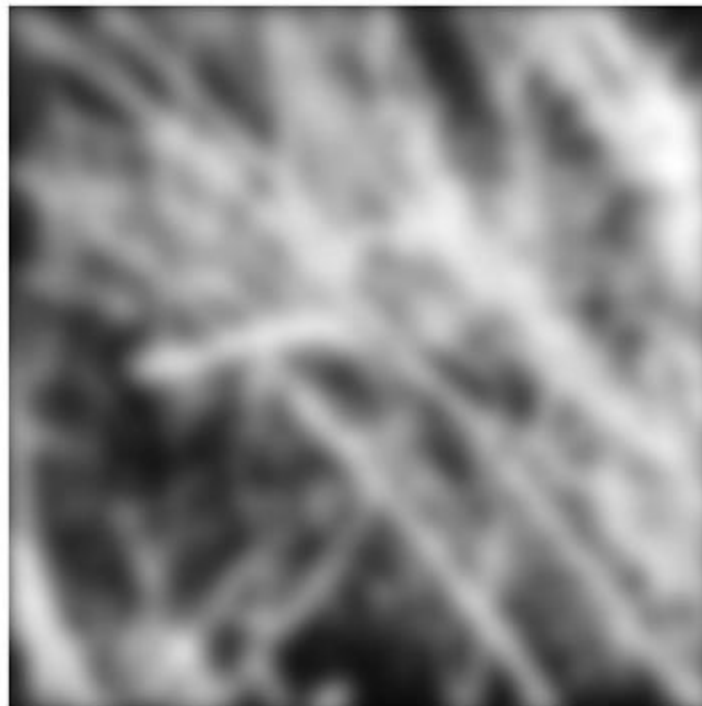
Выбор размера ядра



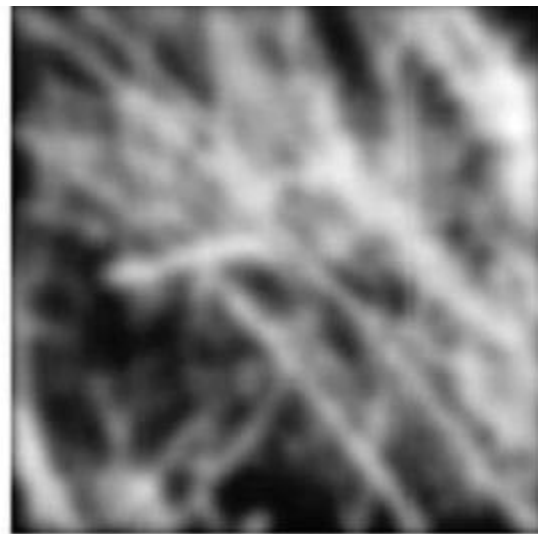
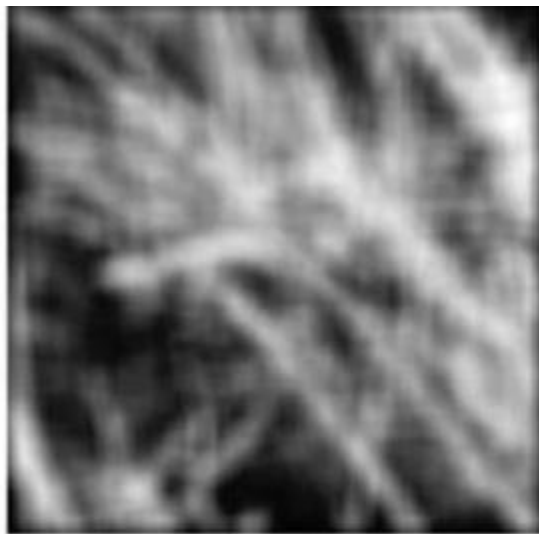
Полуразмер
фильтра = 3σ

Source: S. Lazebnik

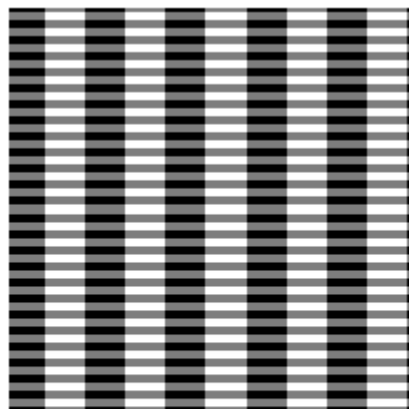
Сглаживание методом Гаусса



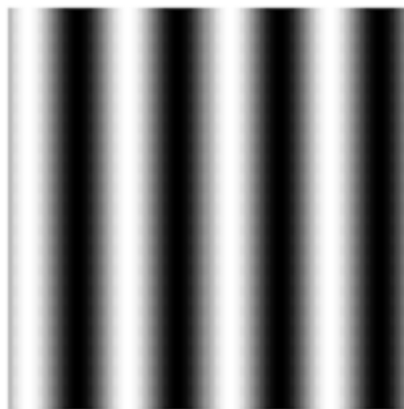
Сравнение



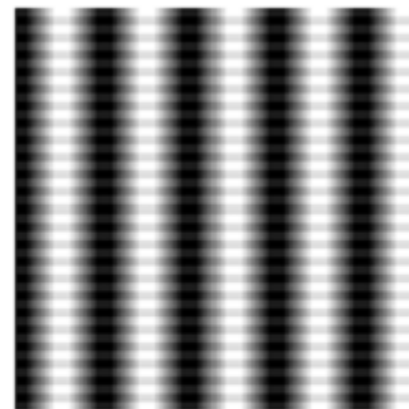
Результаты свертки по функции Гаусса и по константной функции (усреднения).



Исходное изображение



Фильтр Гаусса с
Sigma = 4



Усреднение по 49
пикселям (7x7)

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 $\sigma\sqrt{2}$

Сепарабельность

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

Пример

2D свертка

$$\begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 2 & 3 & 3 \\ \hline 3 & 5 & 5 \\ \hline 4 & 4 & 6 \\ \hline \end{array} =$$

Фильтр раскладывается
в произведение двух 1D
фильтров:

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

Свертка по строкам:

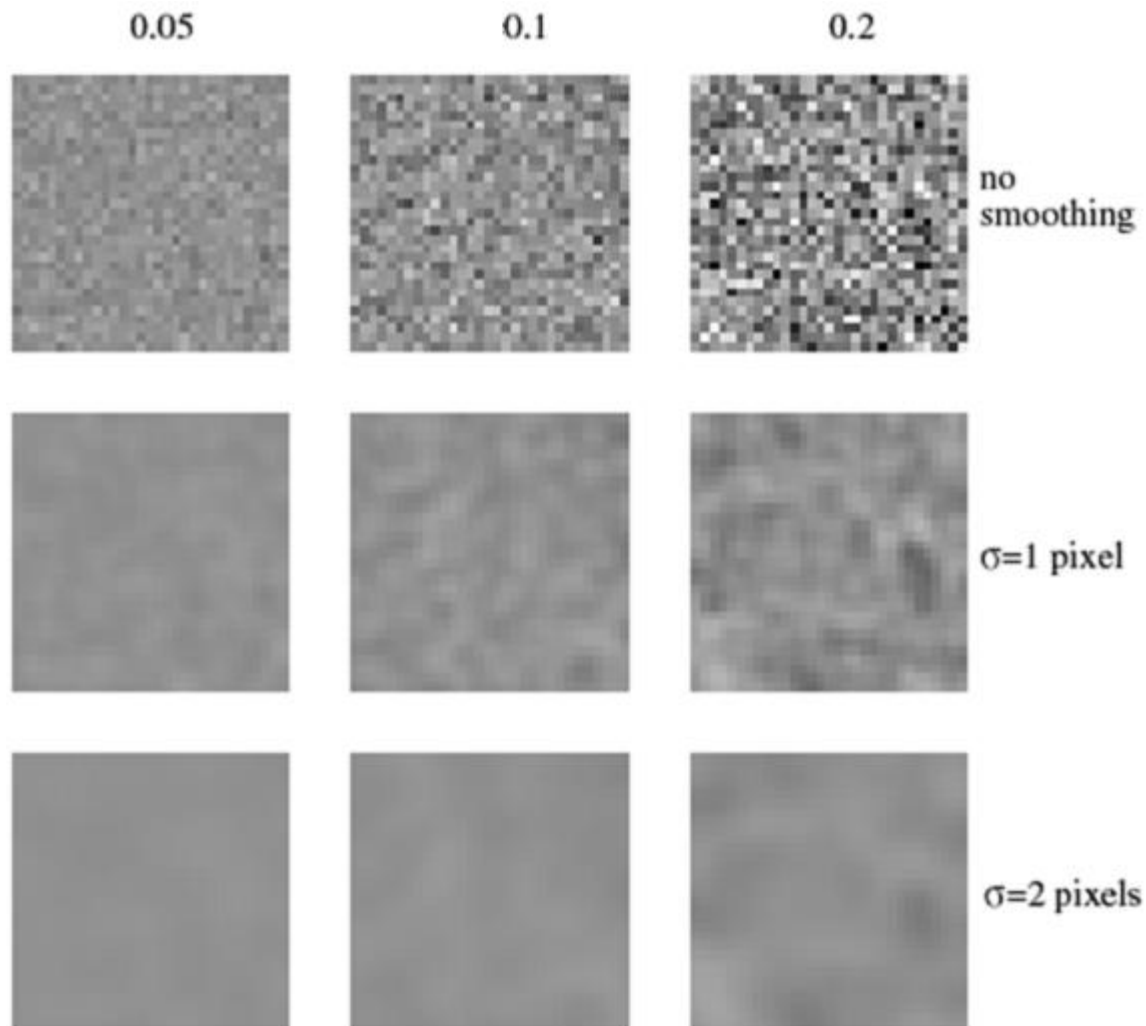
$$\begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 2 & 3 & 3 \\ \hline 3 & 5 & 5 \\ \hline 4 & 4 & 6 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline & 11 & \\ \hline & 18 & \\ \hline & 18 & \\ \hline \end{array}$$

Затем свертка по столбцу:

$$\begin{array}{|c|} \hline 1 \\ \hline 2 \\ \hline 1 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline & 11 & \\ \hline & 18 & \\ \hline & 18 & \\ \hline \end{array} =$$

Source: K. Grauman

Подавление гауссова шума



Подавление шума «соль и перец»

3x3



5x5



7x7



Чем результат плох?

Slide by S. Lazebnik

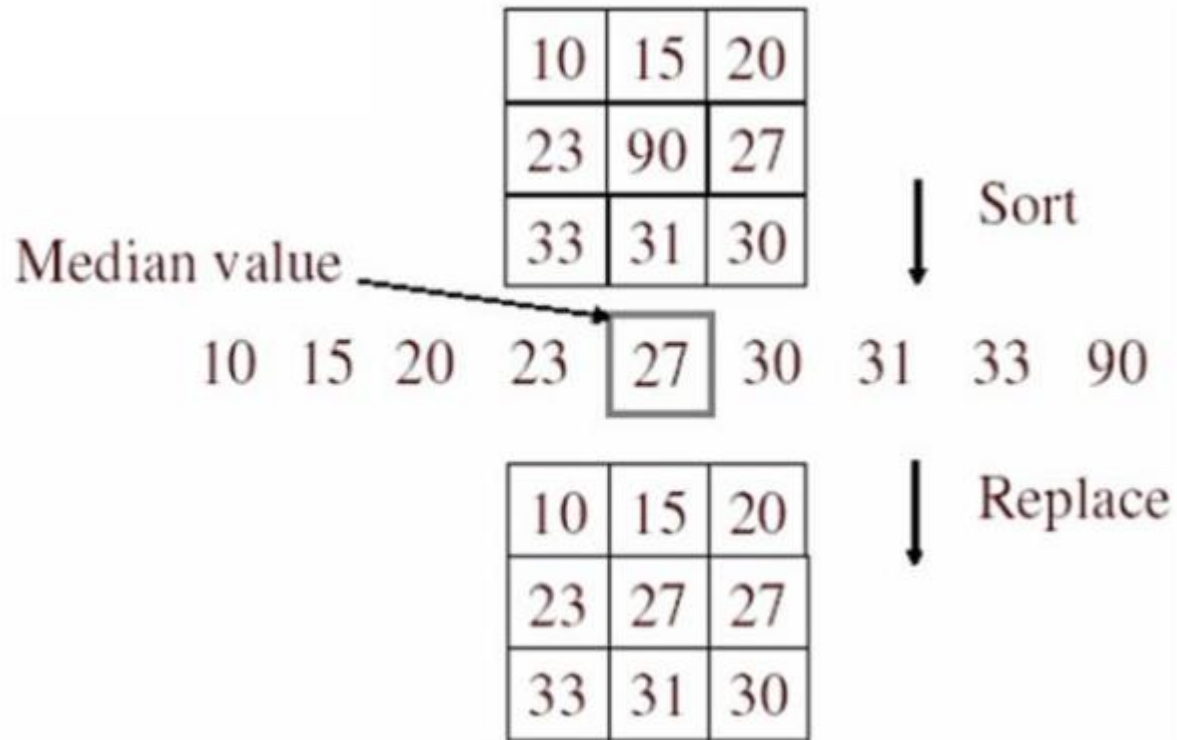
Нелинейное усреднение

Фильтры, основанные на порядковых статистиках:

- классический медианный
- максимума
- минимума

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

Медианный фильтр



Source: K. Grauman

Пример медианного фильтра

$$x = [2 \ 80 \ 6 \ 3]$$

$$y[1] = \text{медиана}[2 \ 2 \ 80] = 2$$

$$y[2] = \text{медиана}[2 \ 80 \ 6] = \text{медиана}[2 \ 6 \ 80] = 6$$

$$y[3] = \text{медиана}[80 \ 6 \ 3] = \text{медиана}[3 \ 6 \ 80] = 6$$

$$y[4] = \text{медиана}[6 \ 3 \ 3] = \text{медиана}[3 \ 3 \ 6] = 3$$

и в итоге:

$$y = [2 \ 6 \ 6 \ 3] \text{ — выход медианного фильтра}$$

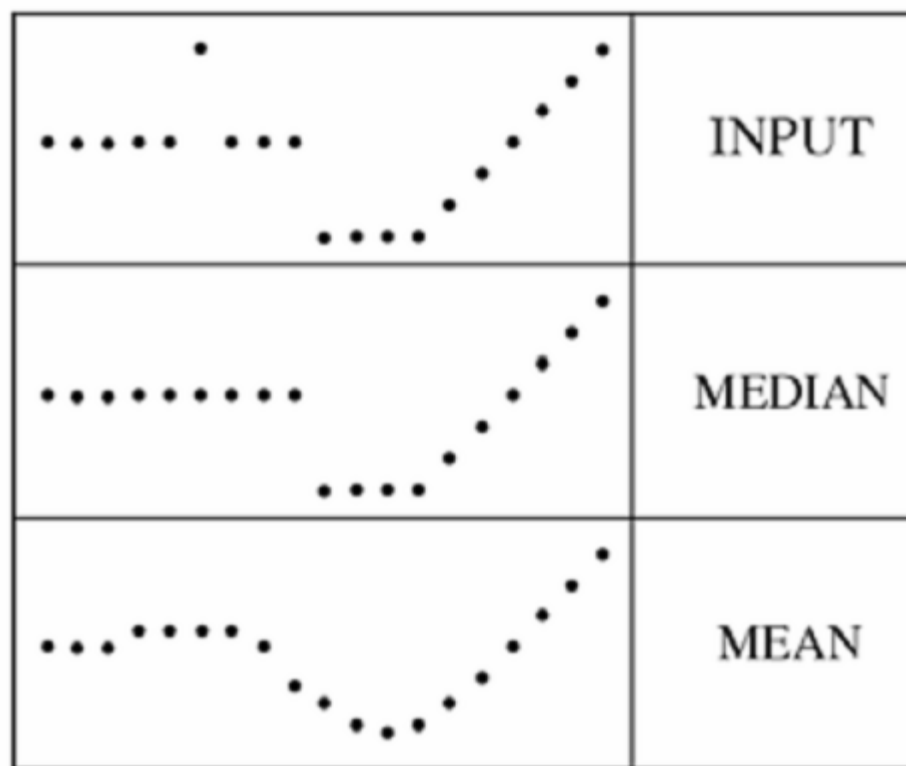
Нелинейность медианного фильтра

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

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

В чем преимущество медианного фильтра перед фильтром Гаусса?

filters have width 5 :



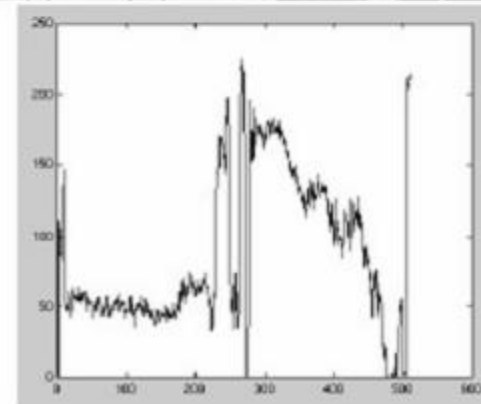
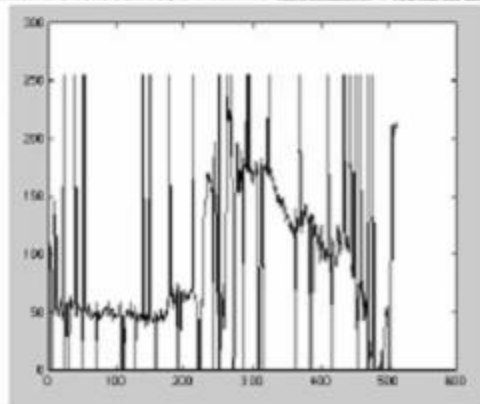
Source: K. Grauman

Медианный фильтр

Salt-and-pepper noise

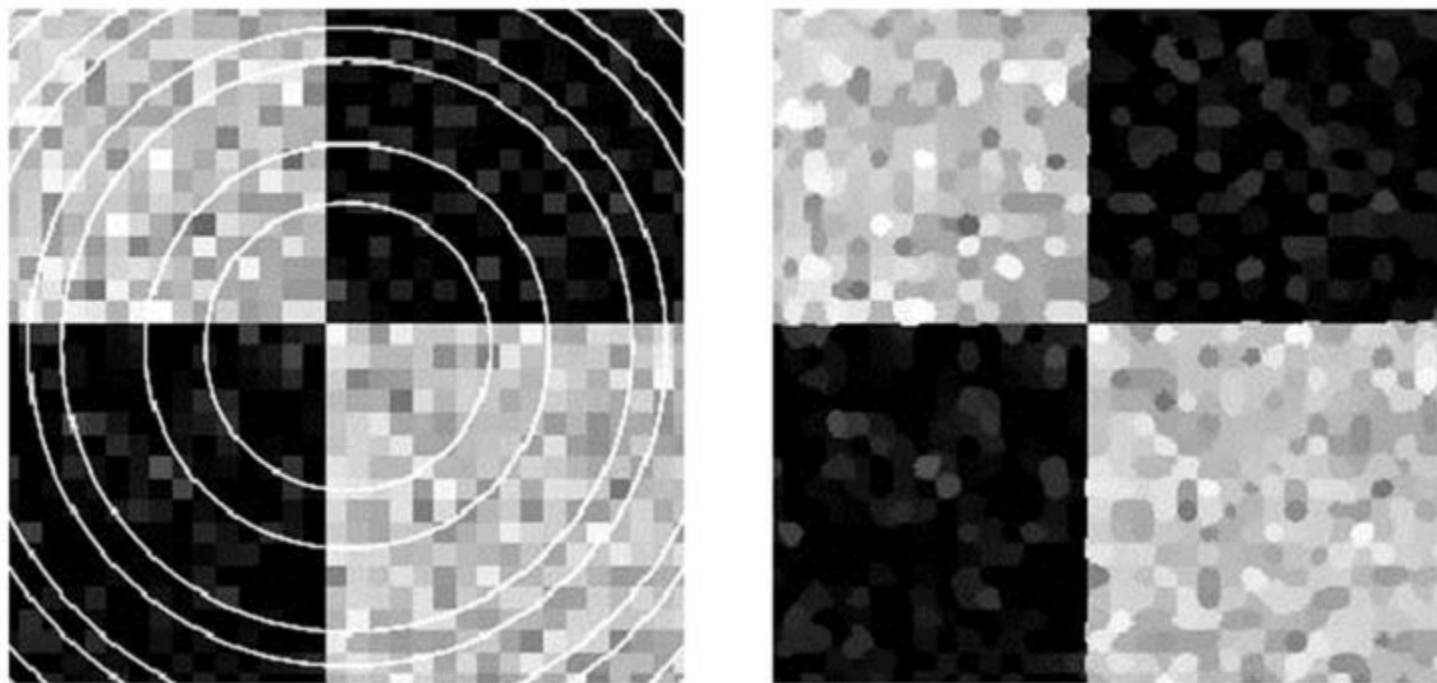


Median filtered



Source: M. Hebert

Медианный фильтр



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

Сравнение

3x3

5x5

7x7

Гауссов



Медианный



Source: S. Lazebnik

Медианный фильтр



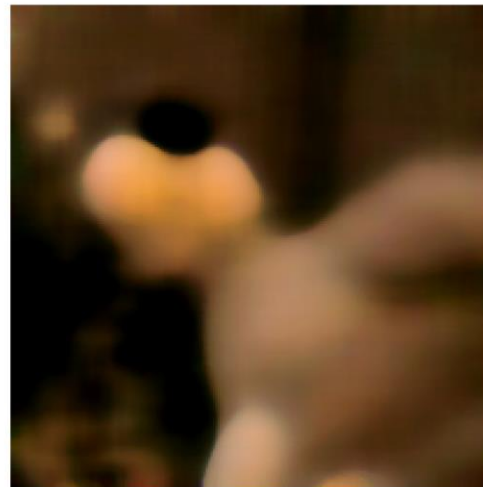
original image



1px median filter



3px median filter



10px median filter

Быстрая реализация медианного фильтра

- Медианный фильтр считается дольше, чем операция свертки, поскольку требует частичной сортировки массива яркостей окрестных пикселей.
- Возможности ускорения:
 - Использовать алгоритмы быстрой сортировки
 - Конкретная реализация для каждого радиуса (3x3, 5x5)
 - Не использовать сортировку вообще – считать через гистограмму окрестности точки

Фильтры максимума и минимума

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

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

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

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

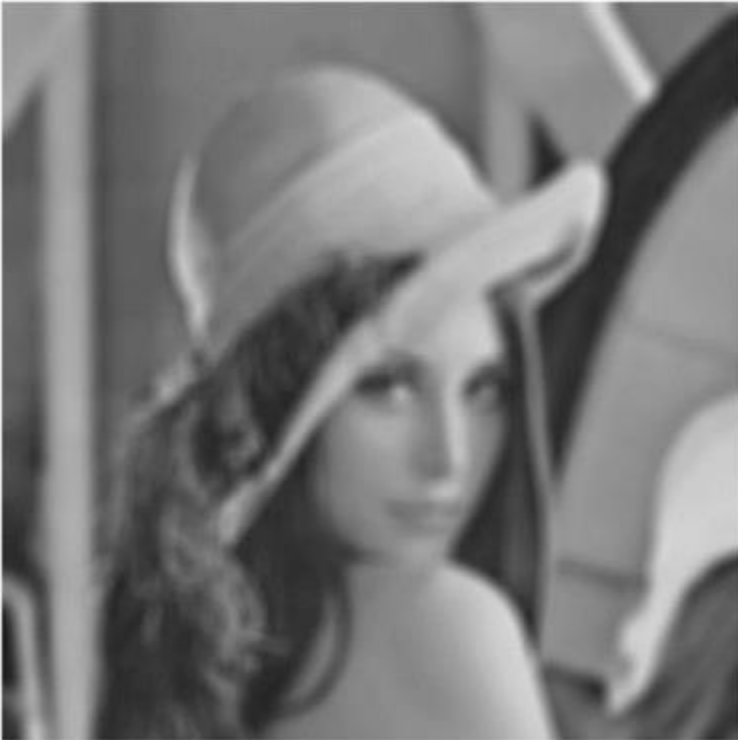
«Продвинутые» фильтры



В чем отличие разных фильтров?

- Box filter (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



Необходимо повысить
резкость (четкость)
изображения

Лирическое отступление

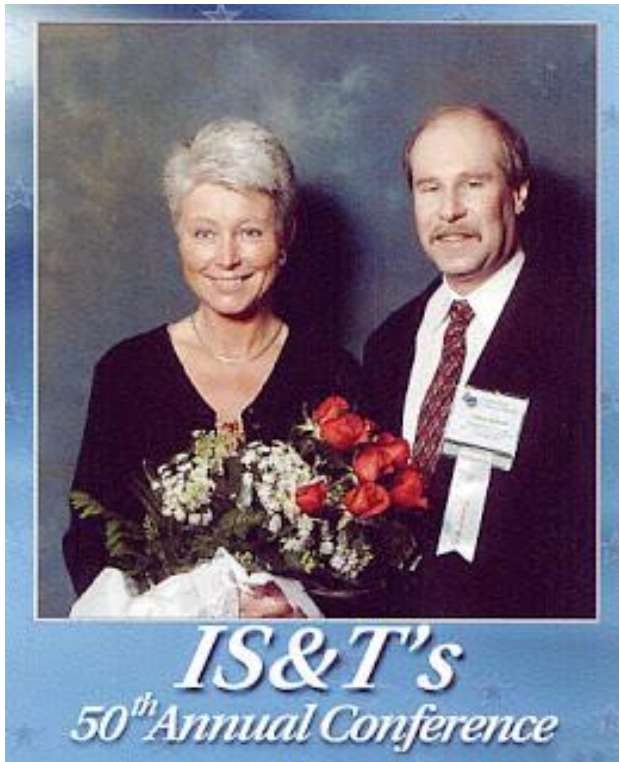


Lena (Lenna)

- 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

Lena Söderberg, born Sjööblom (born 31 March 1951)

is a Swedish, as **Lenna Sjööblom**



Пример повышения резкости

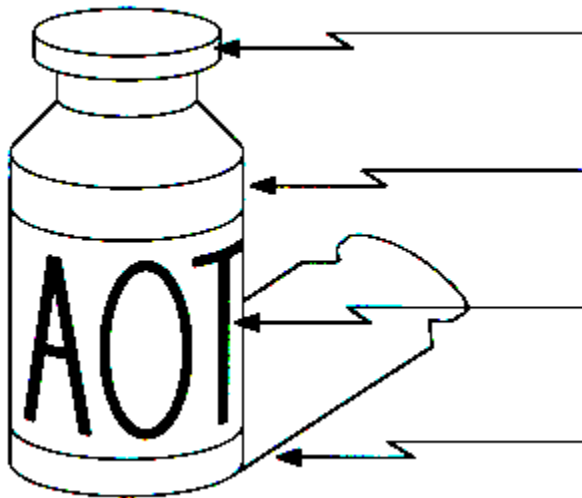
Ядро
свертки

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



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

- достигается пространственным дифференцированием
- основаны на первой и второй производной

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

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

Упрощенная схема



Кросс-корреляция сигналов с маской первой производной $M=[-1, 0, 1]$

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

Кросс-корреляция сигналов 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

Первая и вторая производные

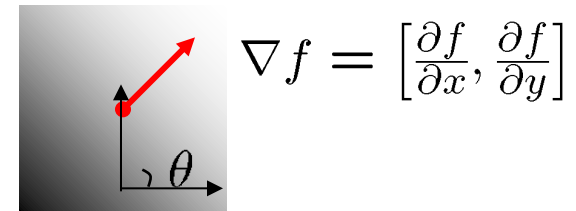
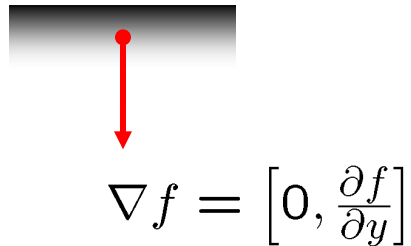
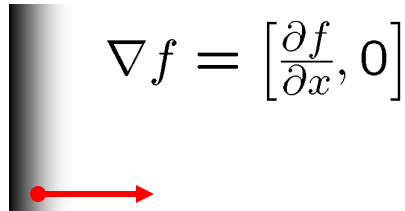
- **Первая** производная обычно формирует более **толстые** контуры.
- **Вторая** производная формирует **большой** по величине отклик на **мелкие детали** (на отдельных точках и на тонких линиях).
- Отклик на ступенчатое изменение яркости у первой производной, как правило, выше, чем у второй.
- Для 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

Градиент изображения

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



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:

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

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

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 90° и 45°

Повышение
резкости
изображения
Северного
полюса Луны
лапласианом



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

Упрощения – Составная маска лапласиан

$$g(x, y) =$$

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

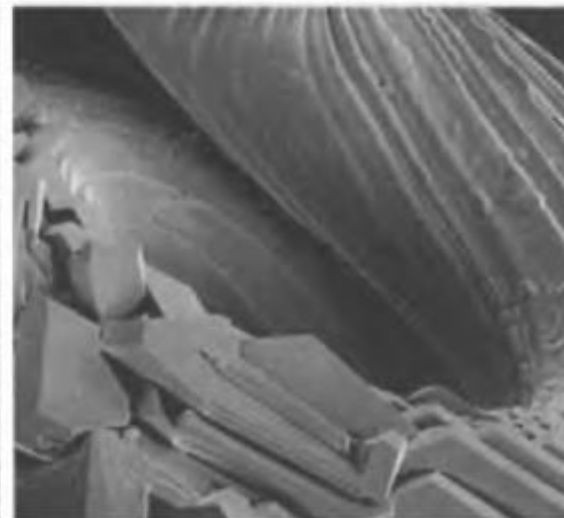
0	-1	0
-1	5	-1
0	-1	0

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

Вольфрамовая нить накала

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

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



Результаты диагональной маски более резкие

Ещё о подходах к повышению резкости

- Нерезкое маскирование
- Фильтрация с подъемом высоких частот

Что теряется при сглаживании?



Добавим дополнительно высокие частоты:



Source: S. Lazebnik

Нерезкое маскирование

Для повышения резкости

Применяется в полиграфии

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

Создание нерезкого изображения

Вычитание из изображения его расфокусированной копии
(результат - маска)

Прибавление маски к исходному изображению

Фильтрация с подъемом высоких частот

Некоторое обобщение нерезкого маскирования

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

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

Типичное приложение фильтрации с подъемом высоких частот

Исходное изображение темнее чем требуется



а) исходное

б) маска

в) $A=1$

г) $A=1,7$



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

Увеличение среднего уровня яркости

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

Улучшение изображений с использованием первых производных: градиент

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

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

$$|\nabla f| \approx |G_x| + |G_y|$$

Перекрестный градиентный оператор Робертса

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

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

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

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

1965г.

Оператор Собела

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

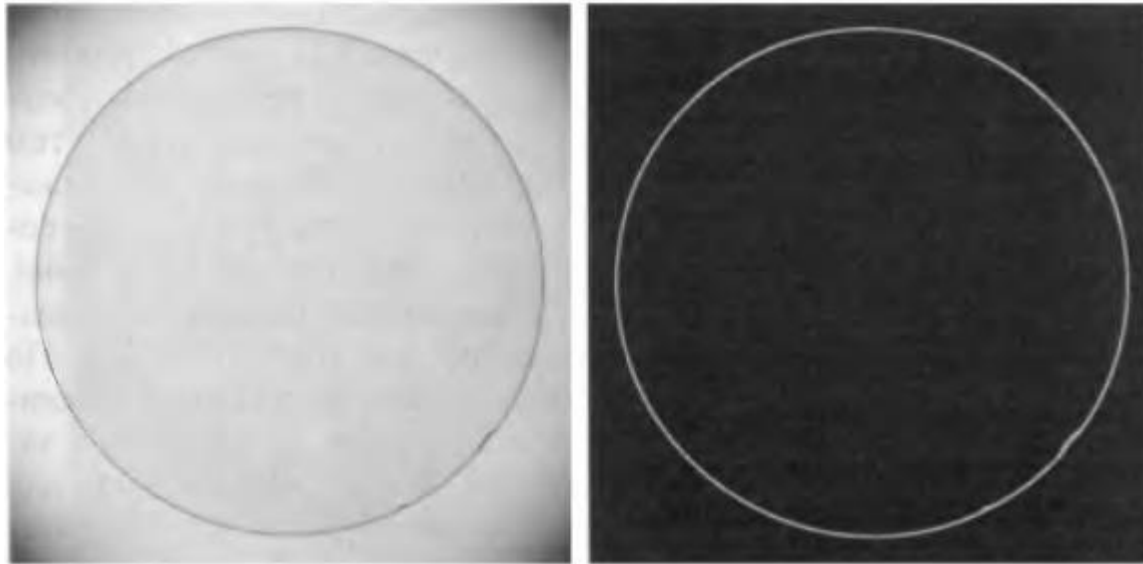
z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

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

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

Пример. Применение градиента для улучшения контуров – оператор Собела

Для дальнейшего автоматического контроля



Определение оператора Превитт (Джуди Превитт)

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

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

Примеры



Робертса



Превитт



Собея

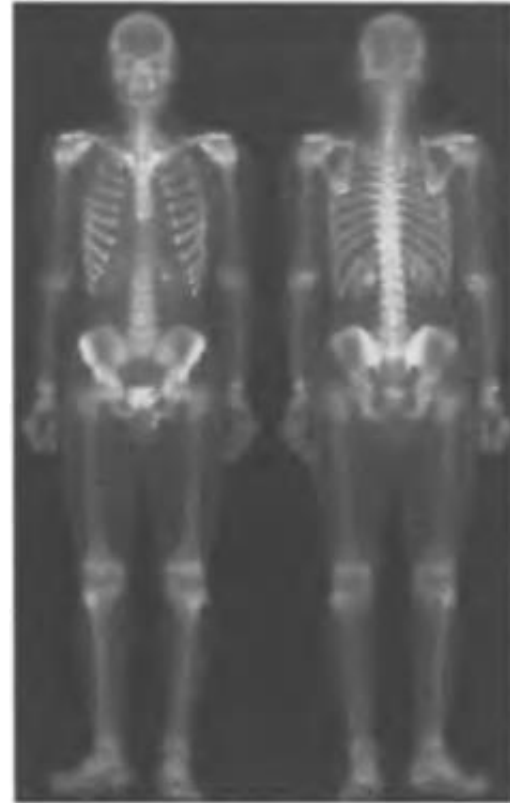
Свойства масок для вычисления производных

- 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 сглаживающих масок

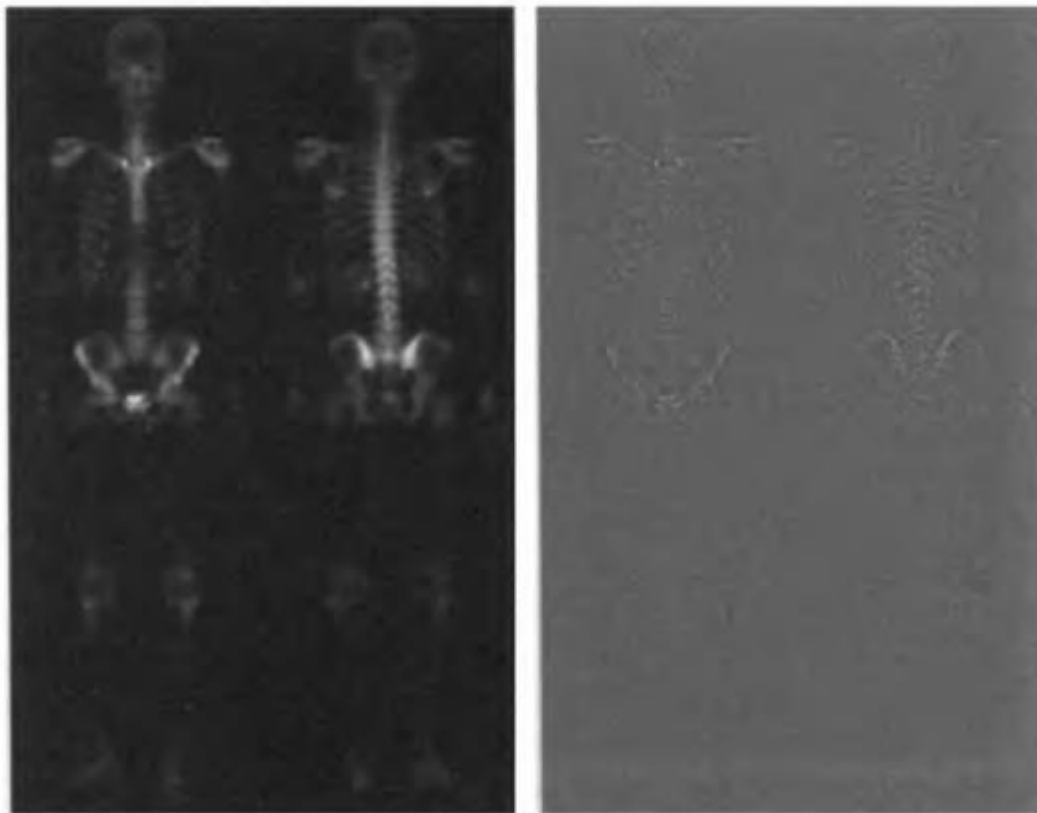
- Элементы сглаживающих масок имеют **положительные значения** и их **сумма равна 1**
- Отклик масок на **областях постоянных значений совпадает с входными значениями**
- **Степень сглаживания** и снижения шума **пропорциональна размеру** маски
- Ступенчатые края размываются **пропорционально** размеру маски

Было – будет



Снимок с помощью гамма-лучей

Комбинирование методов пространственного улучшения



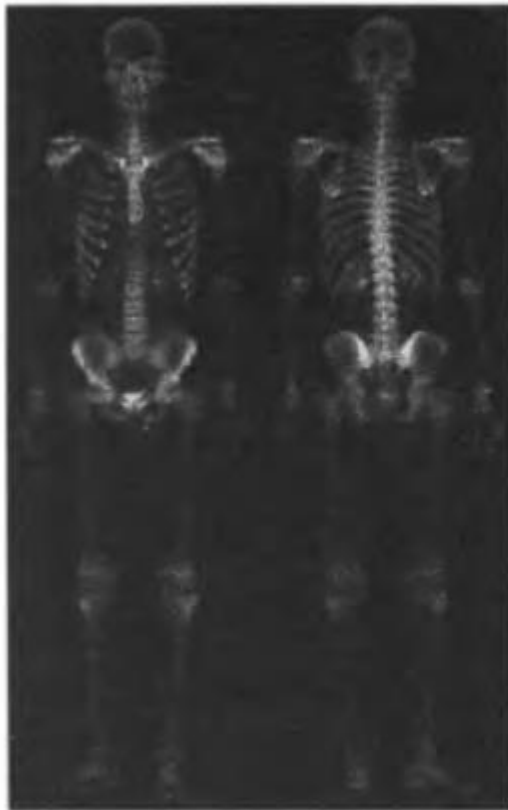
1

2

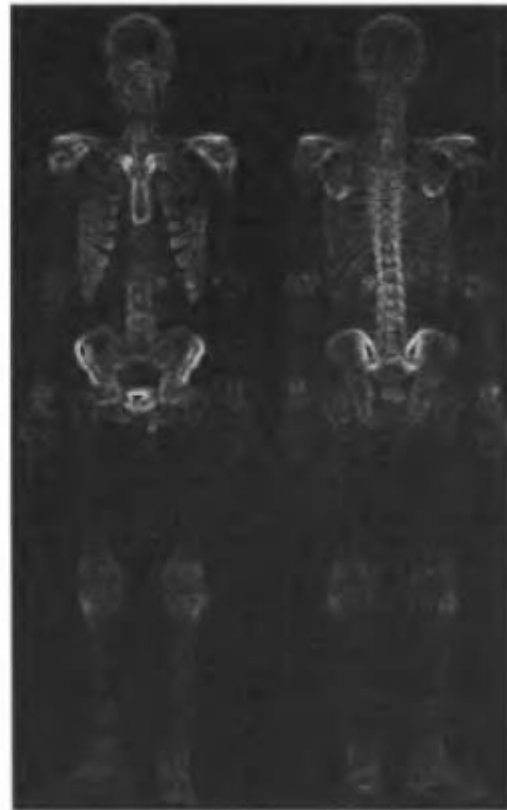
Применение Лапласиана для выявления мелких деталей

Комбинирование методов пространственного улучшения

Повышение
резкости
сложением
двух
предыдущих
изображений



3

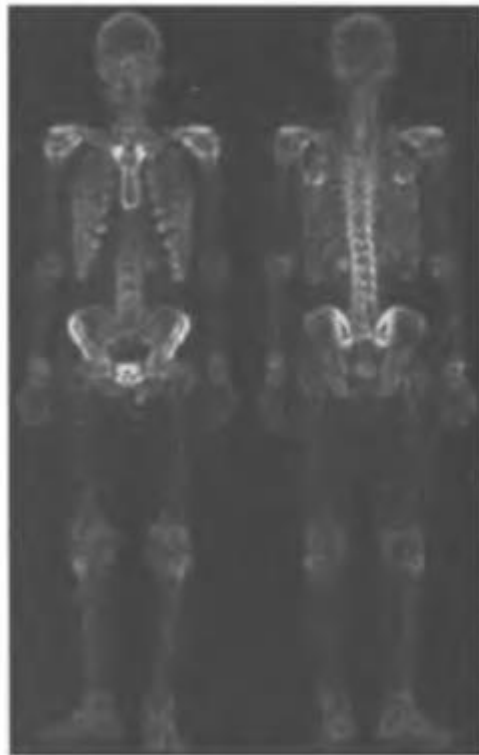


4

Применение
оператора
Собела – для
улучшения
выступающих
краев

Комбинирование методов пространственного улучшения

Сглаживание
градиента
усредняющим
фильтром 5x5



5

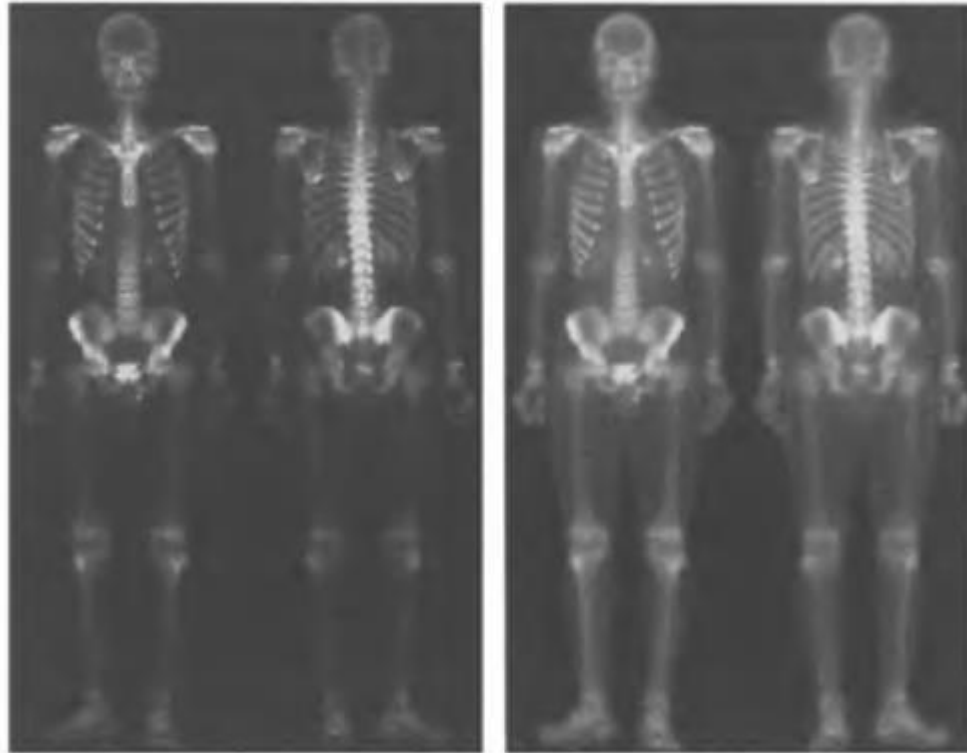


6

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

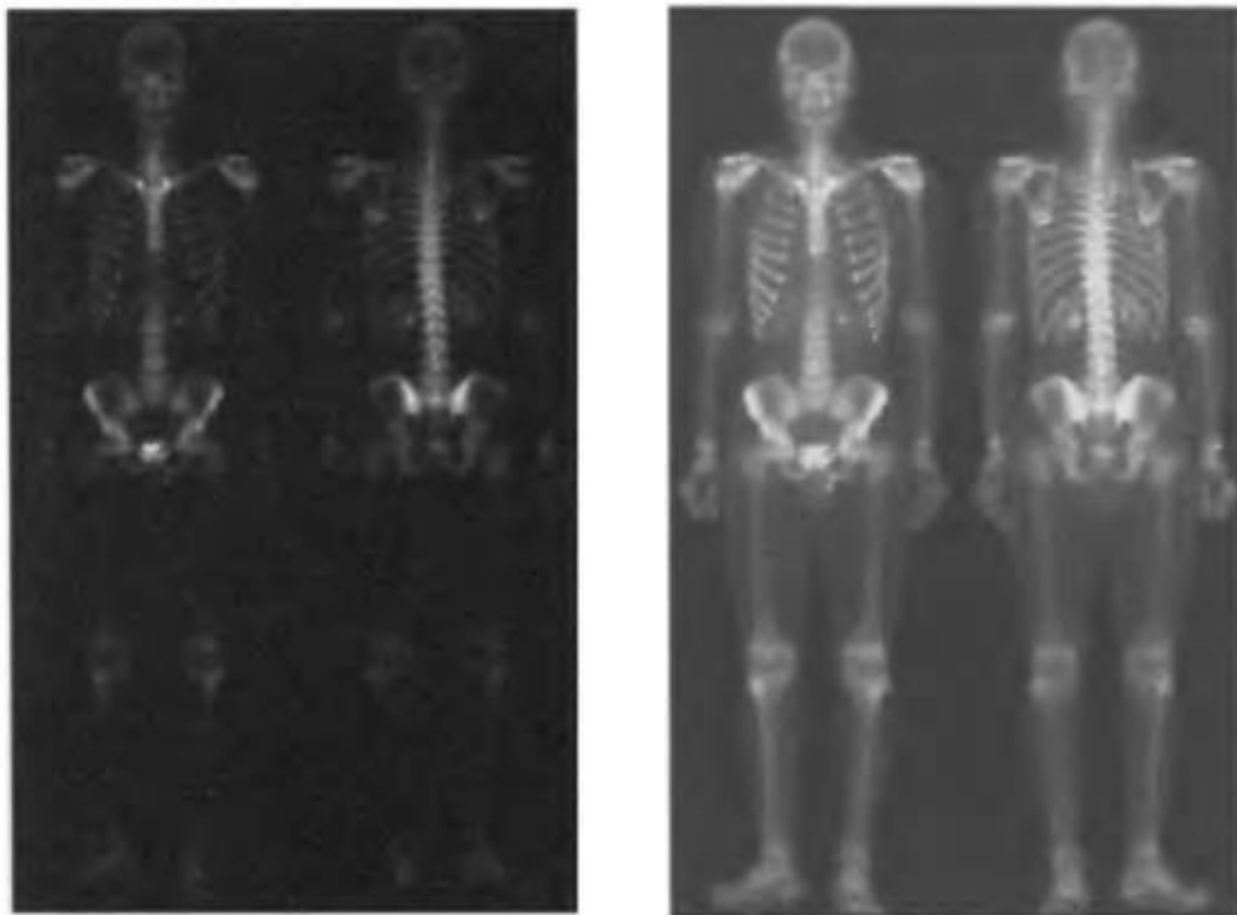
Комбинирование методов пространственного улучшения

Повышение
резкости
сложением
изображений
1 и 6



Применение
градациионной
 γ -коррекции
($\gamma=0,5$ $c=1$)

Было – стало



Предварительная обработка изображений

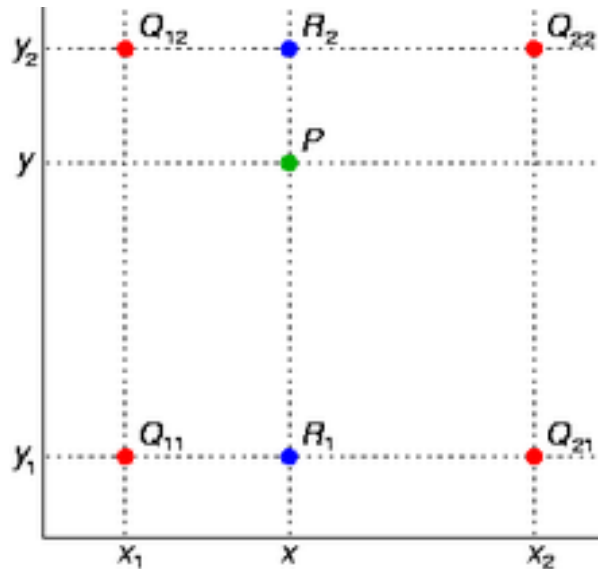
Примеры задач

- увеличение фрагментов
- геометрические преобразования
- повышение контрастного разрешения
- выделение наиболее информативных фрагментов
- улучшение качества изображений

Увеличение цифровых изображений

- Интерполяция по ближайшему соседу
- Метод дублирования пикселей – частный случай
 - Эффект пикселизации
- Билинейная интерполяция

Билинейная интерполяция



$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$$

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$

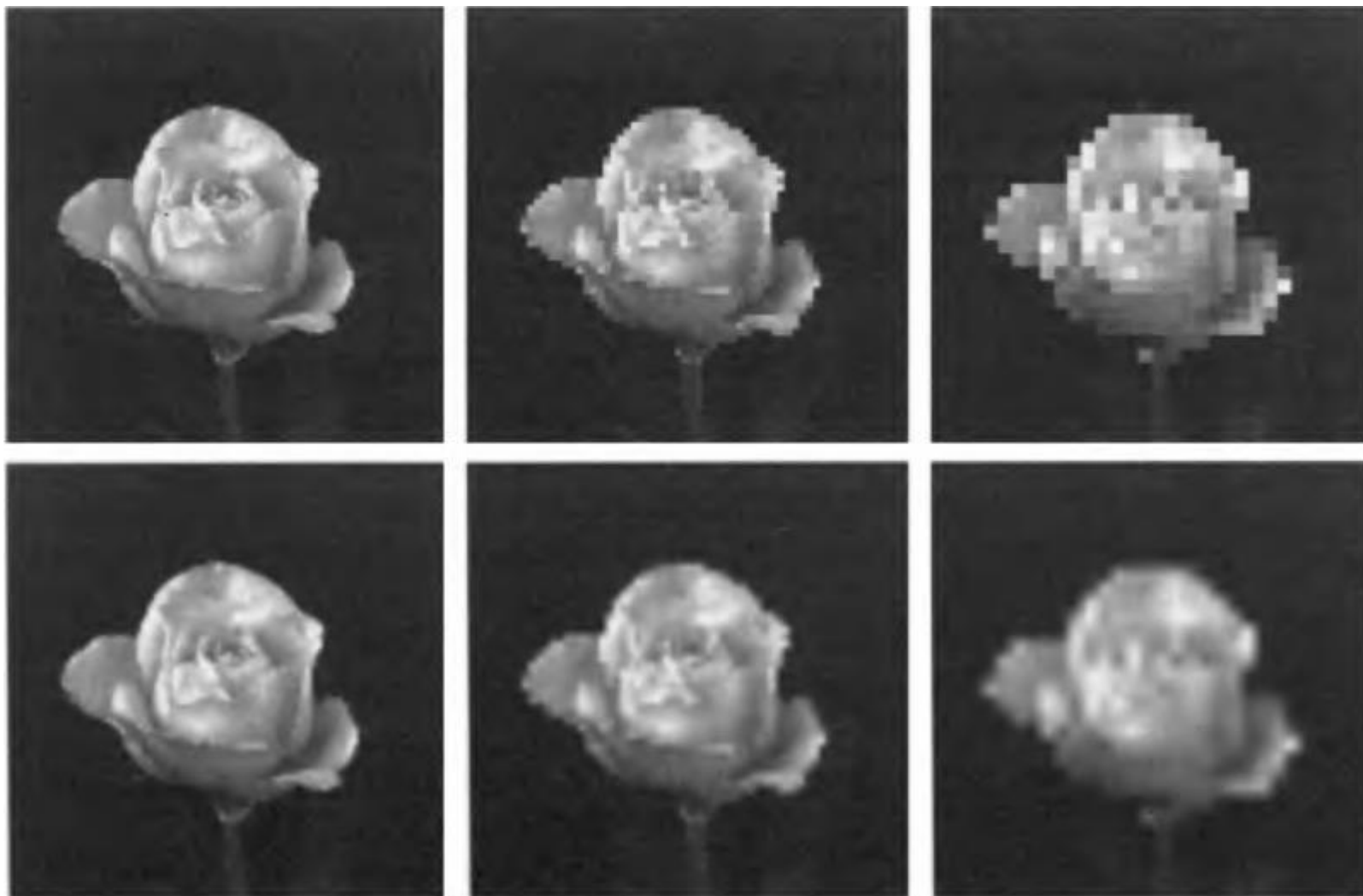
$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2).$$

Четыре красные точки представляют собой известные значения функции. Значение в зеленой точке должно быть интерполировано.

Пример увеличения



Результаты: по ближайшему соседу и билинейная



Недостаток метода билинейной интерполяции

Исходное изображение $W \times H$ пикселей

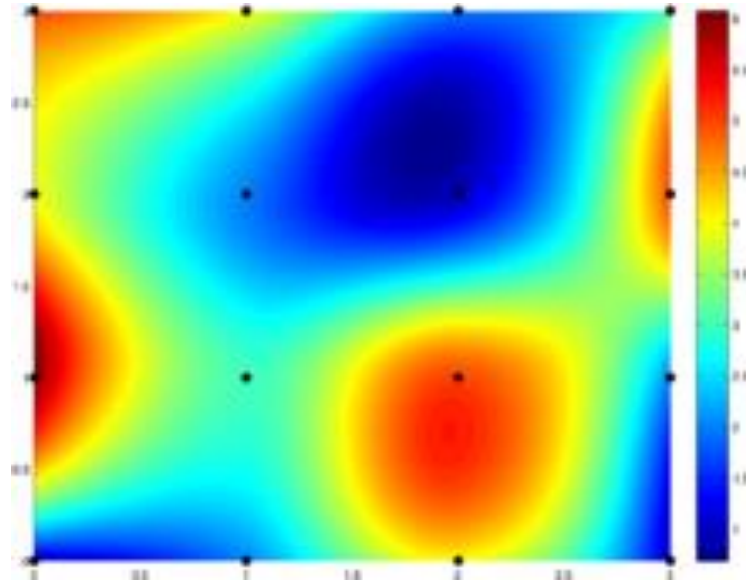
Увеличение в N раз

Результирующее изображение не $NW \times NH$ пикселей, а $(N(W-1)+1) \times (N(H-1)+1)$ пикселей.

Интерполяция с большим числом соседей

- 3-D графика
- Обработка медицинских изображений

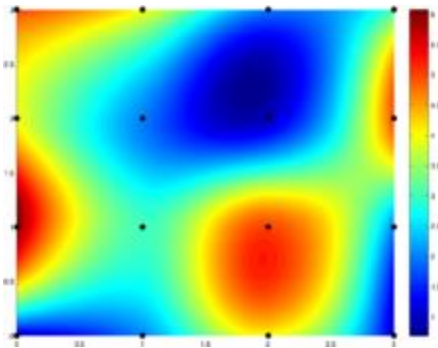
Бикубическая интерполяция



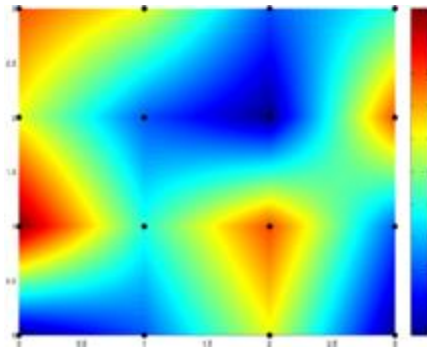
Сетка $[0,3] \times [0,3]$

16 соседей

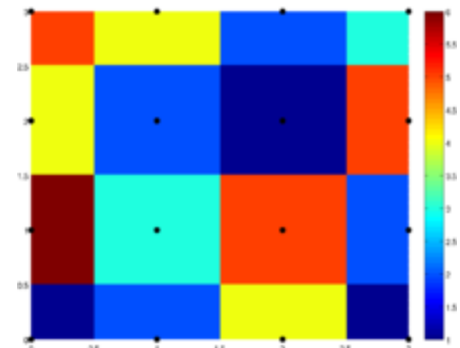
Сравнение



Бикубическая








Билинейная



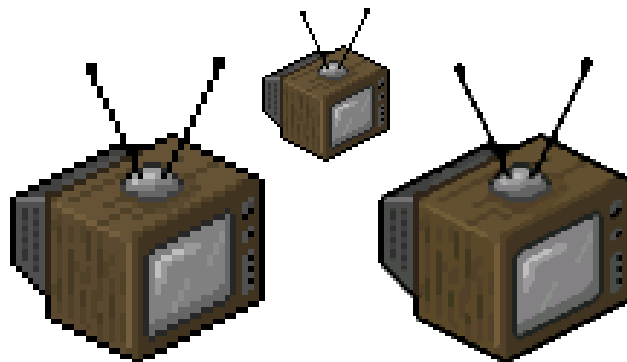
Ближайшего
соседа

Алгоритмы масштабирования

Алгоритм	Изображение
(Исходные изображения)	
Super-xBR 4x	
SuperEagle	
SuperSal	
Sal 2x	
Scale 2x	

Изображение, увеличенное при помощи

метода ближайшего соседа (слева) и 2xSaI (справа)



EPX/Scale2x/AdvMAME2x

- EPX («Eric's Pixel eXpansion», пиксельное увеличение Эрика) — алгоритм, разработанный Эриком Джонстоном из [LucasArts](#) приблизительно в 1992 году

такие как AdvMAME2x и Scale2x, разработанные приблизительно в 2001 году, имеют более эффективную, но функционально идентичную, реализацию:

```
  А      --\ 1 2
С Р В   --/ 3 4
  D
1=P; 2=P; 3=P; 4=P;
Если С==А и С!=D и А!=В => 1=А
Если А==В и А!=С и В!=D => 2=В
Если В==D и В!=А и D!=С => 4=D
Если D==С и D!=В и С!=А => 3=С
```

Семейство $hq3x$



Изображение, увеличенное в 3
раза методом ближайшего
соседа



Изображение, полученное при
использовании алгоритма $hq3x$.

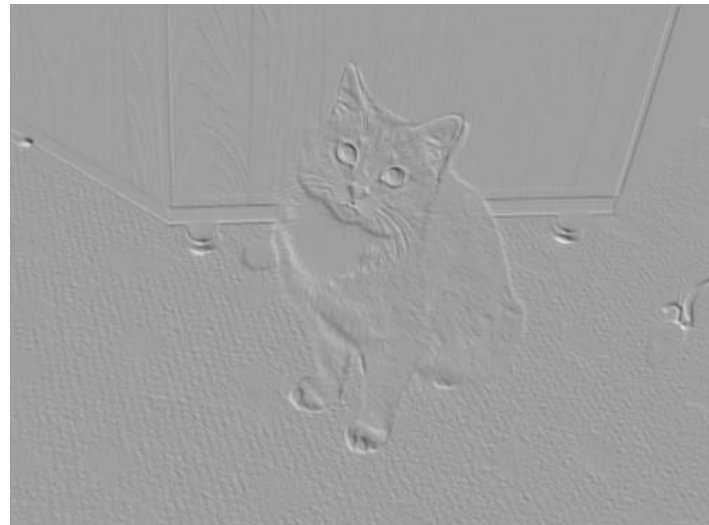
Спецэффекты

- Рассмотрим
 - Тиснение
 - Негатив
 - «Светящиеся» края
 - Геометрические эффекты
 - Перенос/поворот
 - Искажение
 - «Эффект стекла»

Тиснение

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

Фильтр + сдвиг яркости, нормировка...

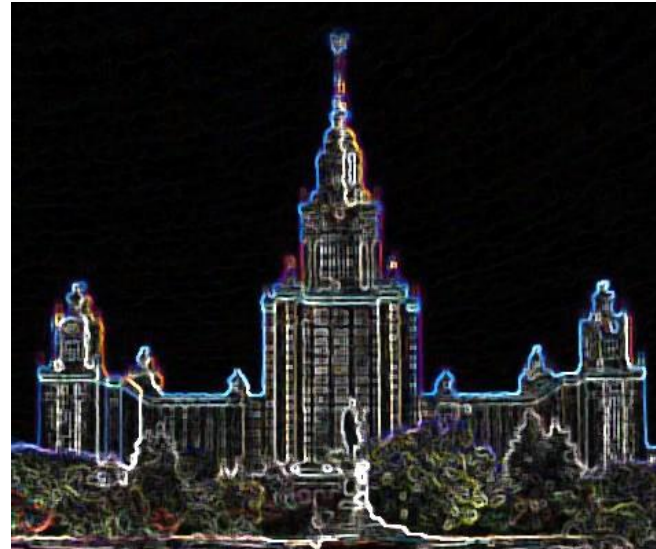


Цифровой негатив



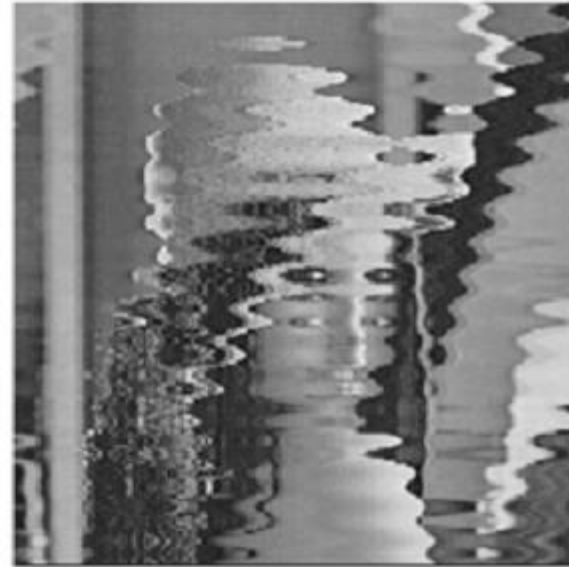
$$R' = 255 - R; \quad G' = 255 - G; \quad B' = 255 - B;$$

Светящиеся края



Медианный фильтр + выделение краев + фильтр «максимума»

«Волны»



Волны 1:

$$x(k; l) = k + 20\sin(2\pi l / 128); y(k; l) = l;$$

Волны 2:

$$x(k; l) = k + 20\sin(2\pi k / 30); y(k; l) = l;$$

«Эффект стекла»



$$x(k; l) = k + (\text{rand}(1, 1) - 0.5) * 10;$$
$$y(k; l) = l + (\text{rand}(1, 1) - 0.5) * 10;$$

Современные методы

- Сейчас методы фильтрации изображений продолжают активно развиваться
- Можно выделить подходы:
 - Вейвлет и т.д. представления
 - Разреженные представления
 - Марковские случайные поля

Резюме лекции

- Линейная фильтрация (свёртка) изображения позволяет решать целый ряд задач – шумоподавление, повышение резкости, оценка градиента
- Выделение краёв изображения в простом случае можно достичь поиском локальных максимумов градиента яркости
- Всё это открытые задачи и сейчас активно продолжают исследоваться!