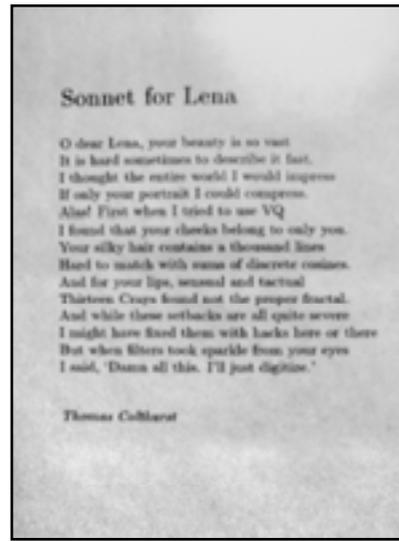
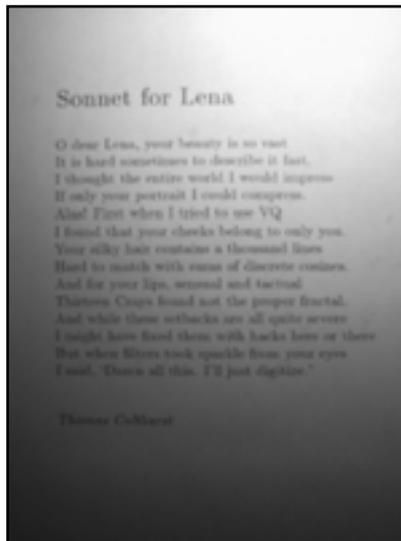




Обработка изображений





Общая информация

Этот курс
подготовлен и
читается при
поддержке

Microsoft
Research

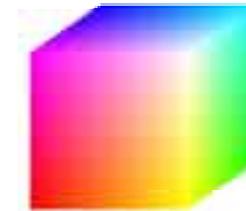
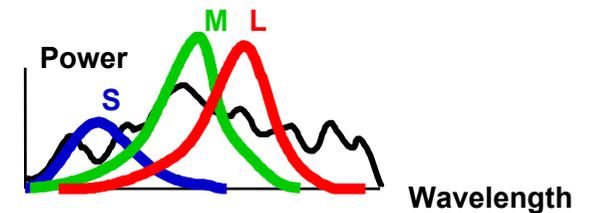
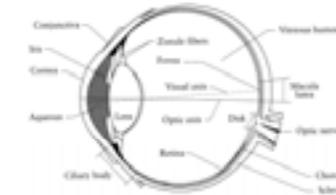
- Страница курса

<http://courses.graphicon.ru/main/vision>



На предыдущей лекции

- Камера-обскура
 - Фотоаппарат, глаз
- Цвет
 - Психологическое свойство человека
 - Свет описывается спектром
- Сетчатка глаза
 - Колбочки 3х видов
 - Трихроматическая теория
 - 3 канала для пиксель
- Адаптация зрения
 - Цветовой баланс, «баланс белого»
 - Серые карточки, «серый мир»





Обработка изображений

- Семейство методов и задач, где входной и выходной информацией являются изображения.
- Примеры :
 - Устранение шума в изображениях
 - Улучшение качества изображения
 - Усиления полезной и подавления нежелательной (в контексте конкретной задачи) информации

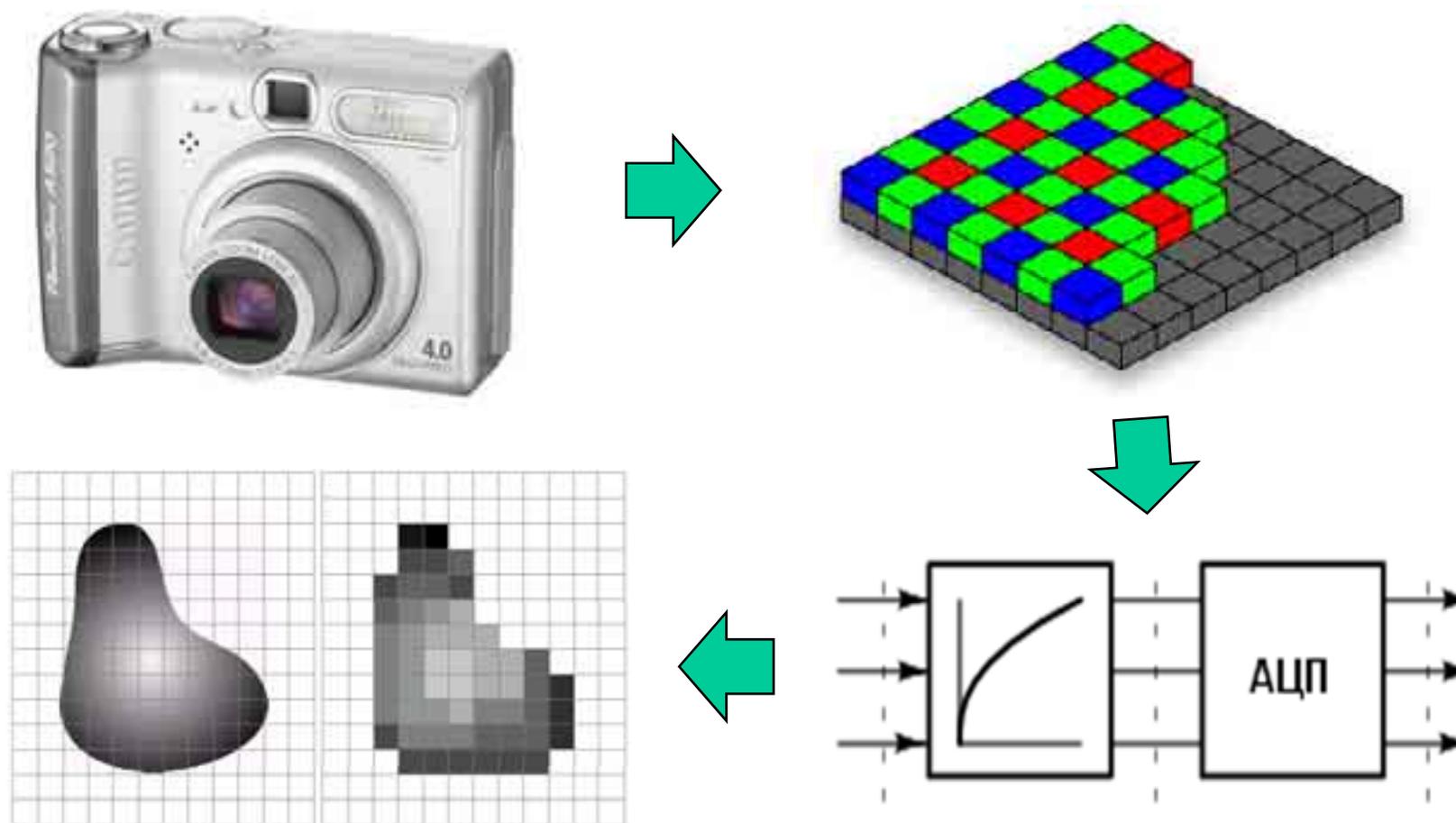


Обработка изображений

- Зачем обрабатывать?
 1. *Улучшение изображения для восприятия человеком*
 - цель – чтобы стало «лучше» с субъективной точки зрения человека
 2. *Улучшение изображения для восприятия компьютером*
 - цель – упрощение последующего распознавания
 3. *Развлечение (спецэффекты)*
 - цель – получить эстетическое удовольствие от красивого эффекта



Цифровое изображение

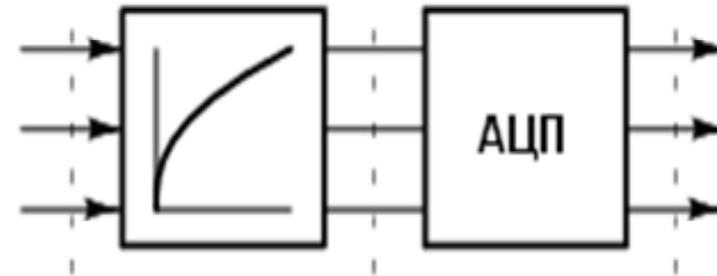


Вспоминаем процесс получения цифрового изображения...



Почему оно может получиться плохо?

- Ограниченный диапазон чувствительности датчика
- “Плохой” функции передачи датчика





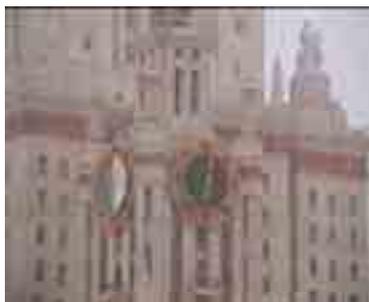
Что такое гистограмма?

Гистограмма – это график распределения яркостей на изображении. На горизонтальной оси - шкала яркостей тонов от белого до черного, на вертикальной оси - число пикселей заданной яркости.



0

255



0

255



Изменение контраста изображения

Что может не устраивать в полученном изображении:

- Узкий или смещенный диапазон яркостей пикселей (тусклое или «пересвеченное» изображение)
- Концентрация яркостей вокруг определенных значений, неравномерное заполнение диапазона яркостей (узкий диапазон - тусклое изображение)

Коррекция - к изображению применяется преобразование яркостей, компенсирующий нежелательный эффект:

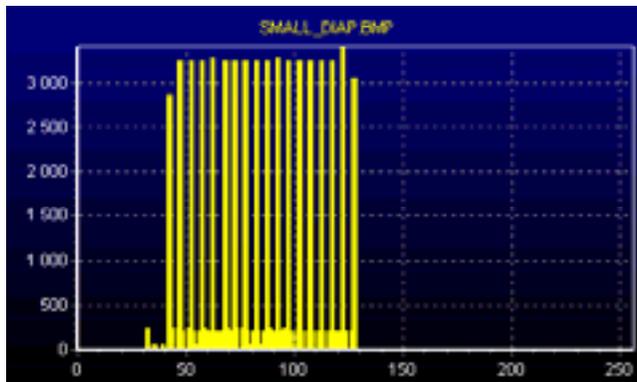
$$f^{-1}(y) = x$$

y – яркость пикселя на исходном изображении,
 x – яркость пикселя после коррекции.



Линейная коррекция

Компенсация узкого диапазона яркостей –
линейное растяжение:



$$f^{-1}(y) = (y - y_{\min}) * \frac{(255 - 0)}{(y_{\max} - y_{\min})}$$

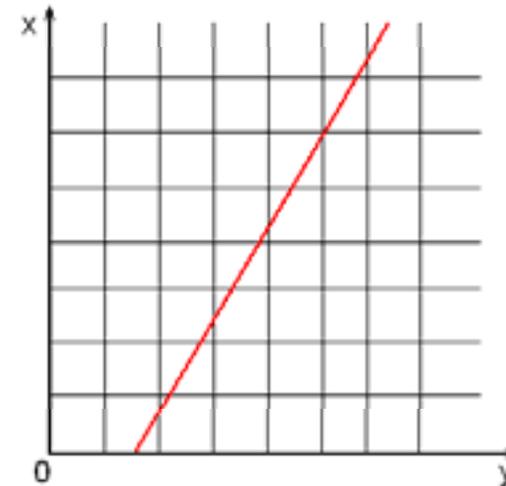
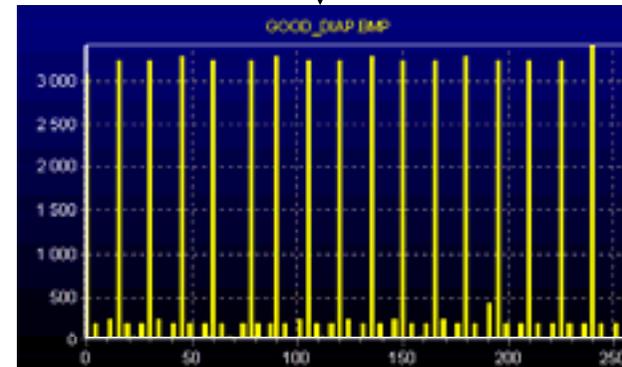
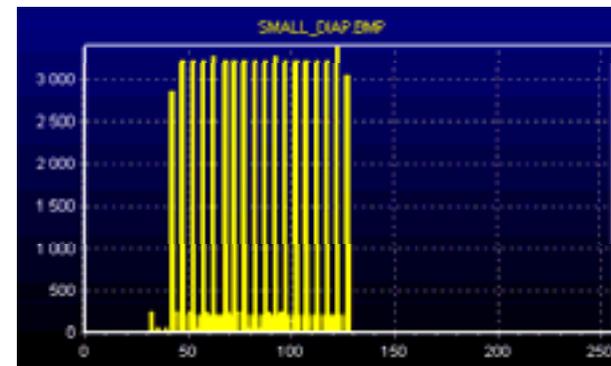


График функции $f^{-1}(y)$



Линейная коррекция

Компенсация узкого диапазона яркостей – линейное растяжение:





Линейная коррекция

Линейное растяжение – «как AutoContrast в Photoshop»





Линейная коррекция

Линейная коррекция помогает не всегда!





Нелинейная коррекция

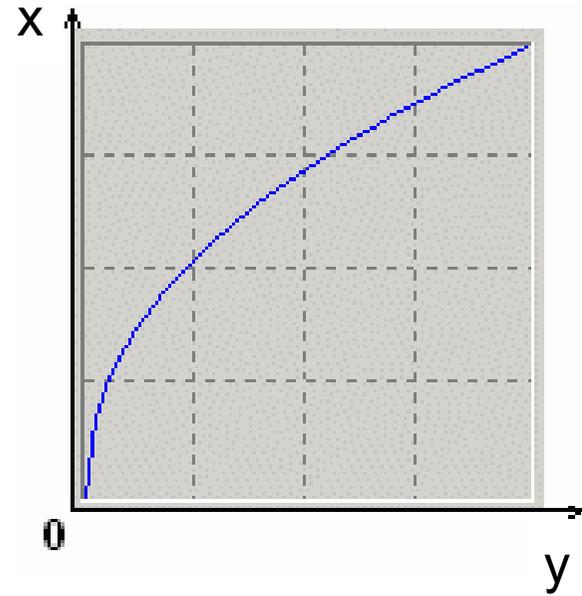


График функции $f^{-1}(y)$



Нелинейная коррекция

Нелинейная компенсация недостаточной контрастности

Часто применяемые функции:

- Гамма-коррекция
 - Изначальная цель – коррекция для правильного отображения на мониторе.

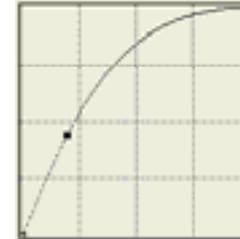
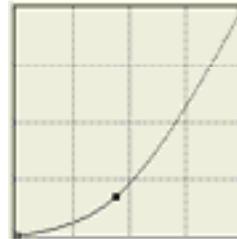
$$y = c \cdot x^\gamma$$

- Логарифмическая
 - Цель – сжатие динамического диапазона при визуализации данных

$$y = c \cdot \log(1 + x)$$



Гамма-коррекция



Графики функции $f^{-1}(y)$



Нелинейная коррекция

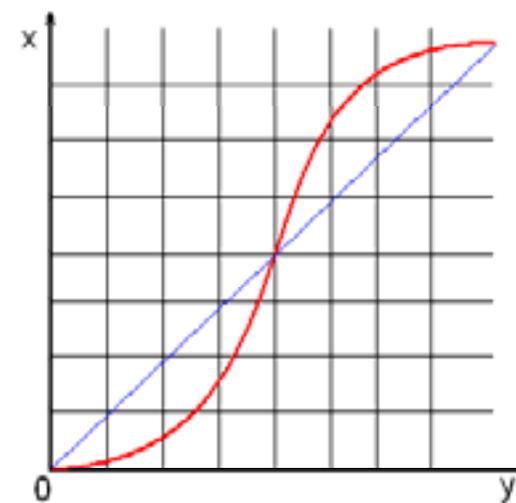
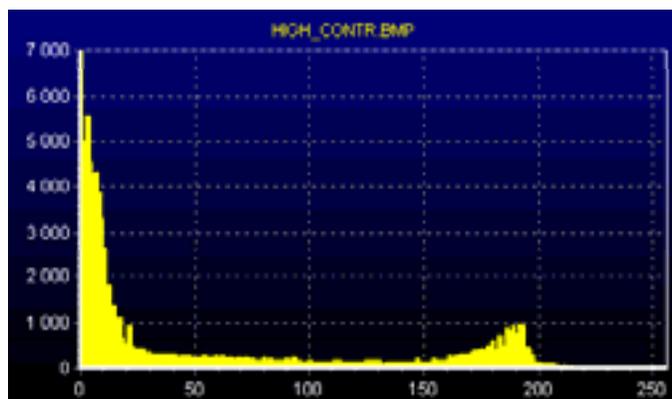
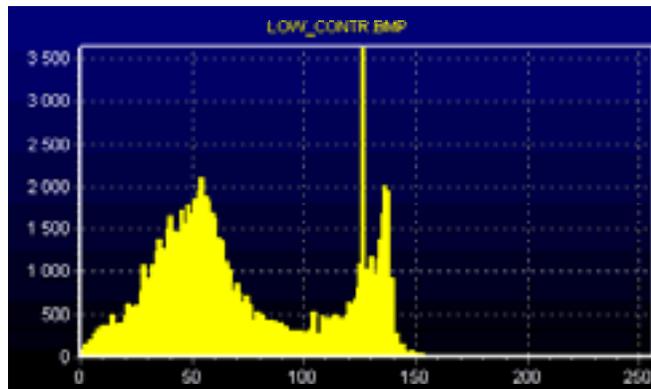


График функции $f^{-1}(y)$



Цветовая коррекция

- Изменение цветового баланса
 - Компенсация:
 - Неверного цветовосприятия камеры
 - Цветного освещения
- Ряд алгоритмов рассмотрели на предыдущей лекции





Цветовая коррекция изображений

- Растяжение контрастности (“autolevels”)
 - Идея – растянуть интенсивности по каждому из каналов на весь диапазон;
- Метод:
 - Найти минимум, максимум по каждому из каналов:

$$R_{\min}, R_{\max}, G_{\min}, G_{\max}, B_{\min}, B_{\max}$$

- Преобразовать интенсивности:

$$(R - R_{\min}) * \frac{(255 - 0)}{(R_{\max} - R_{\min})}; \quad (G - G_{\min}) * \frac{(255 - 0)}{(G_{\max} - G_{\min})};$$

$$(B - B_{\min}) * \frac{(255 - 0)}{(B_{\max} - B_{\min})};$$

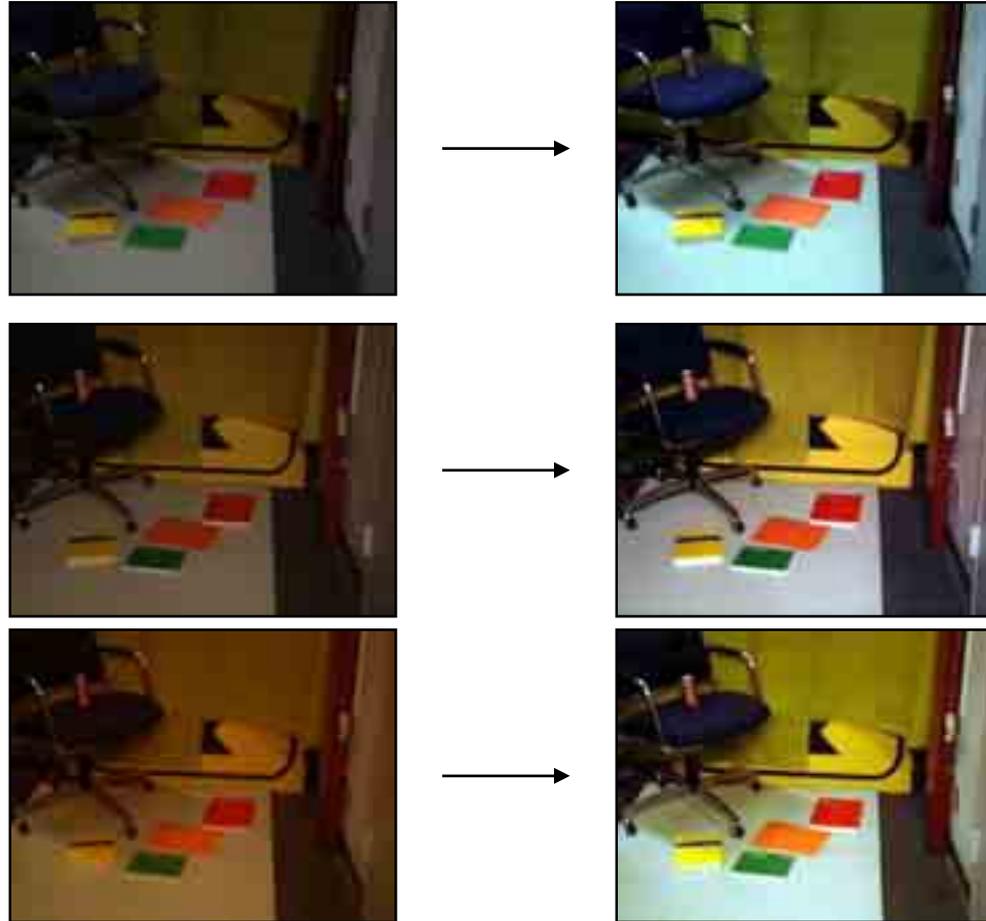


Растяжение контрастности





Растяжение контрастности



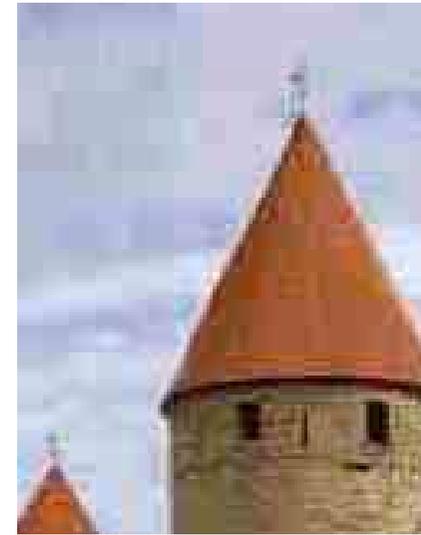


Шумоподавление

- Причины возникновения шума:
 - Несовершенство измерительных приборов
 - Хранение и передача изображений с потерей данных



Шум фотоаппарата



Сильное сжатие JPEG



Цель: подавление шума

Пусть дана камера и статичная сцена, требуется подавить шум.



Простейший вариант: усреднить несколько кадров



Усреднение

- Заменяем каждый пиксель взвешенным средним по окрестности
- Веса обозначаются как *ядро фильтра*
- Веса для усреднения задаются так:

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

1	1	1
1	1	1
1	1	1

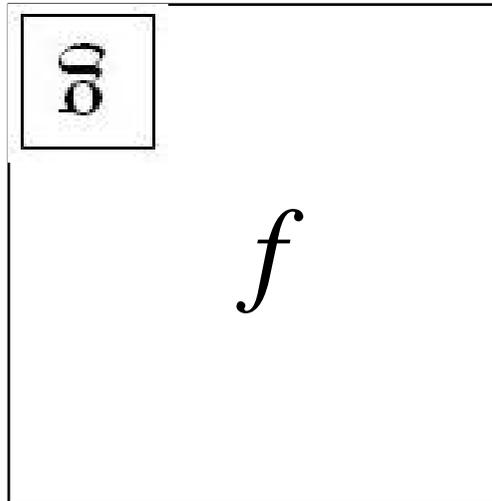
“box filter”



Определение свертки

- Пусть f – изображение, g – ядро. Свертка изображения f с помощью g обозначается как $f * g$.

$$(f * g)[m, n] = \sum_{k, l} f[m - k, n - l] g[k, l]$$



- Соглашение: ядро “перевернуто”
- MATLAB: conv2 vs. filter2 (also imfilter)



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: $\text{filter}(\text{shift}(f)) = \text{shift}(\text{filter}(f))$
- Теория: любой линейный оператор, инвариантный к сдвигу, может быть записан в виде свертки



Свойства

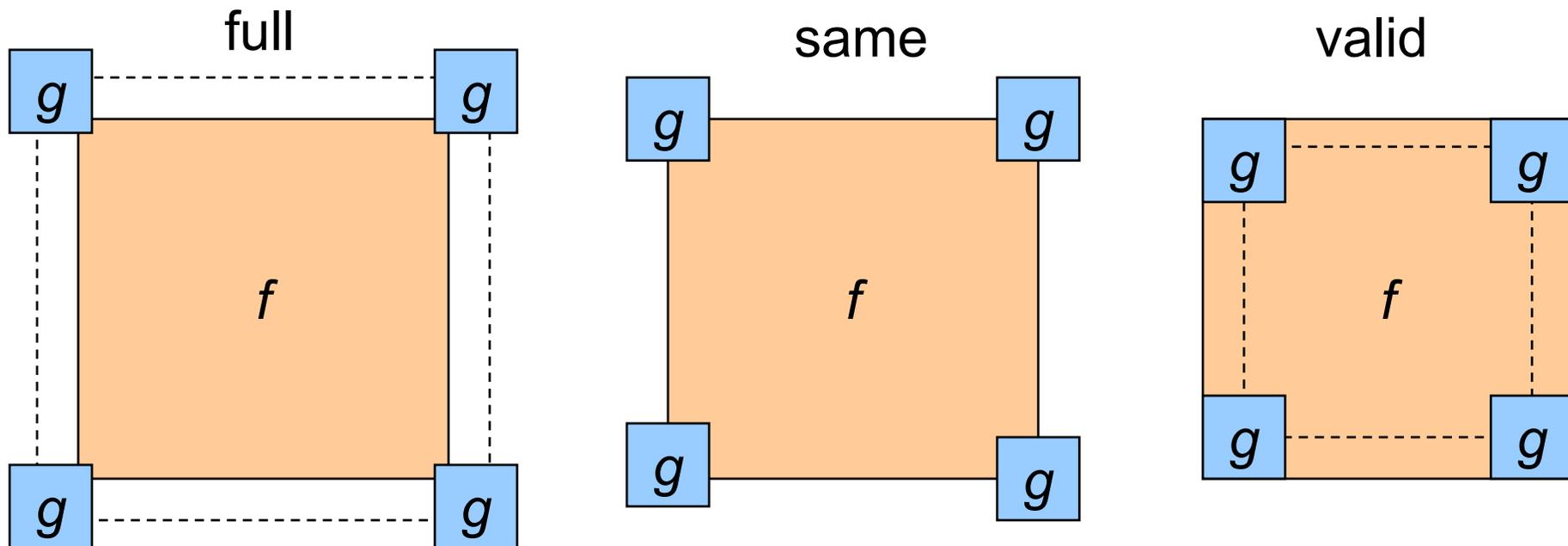
- Коммутативность: $a * b = b * a$
 - Нет никакой разницы между изображением и ядром фильтра
- Ассоциативность: $a * (b * c) = (a * b) * c$
 - Последовательное применение фильтров: $((a * b_1) * b_2) * b_3$
 - Эквивалентно применению такого фильтра: $a * (b_1 * b_2 * b_3)$
- Дистрибутивность по сложению:
$$a * (b + c) = (a * b) + (a * c)$$
- Домножение на скаляр можно вынести за скобки: $ka * b = a * kb = k(a * b)$
- Единица: $e = [\dots, 0, 0, 1, 0, 0, \dots]$,
 $a * e = a$



Детали реализации

Размер результирующего изображения?

- MATLAB: `filter2(g, f, shape)`
 - *shape* = 'full': output size is sum of sizes of *f* and *g*
 - *shape* = 'same': output size is same as *f*
 - *shape* = 'valid': output size is difference of sizes of *f* and *g*





Детали реализации

Как происходит фильтрация по краям?

- Окно фильтра выходит за границы изображения
- Необходимо экстраполировать изображение
- Варианты:
 - clip filter (black)
 - wrap around
 - copy edge
 - reflect across edge





Простейшие фильтры



Original

0	0	0
0	1	0
0	0	0

?



Простейшие фильтры



Original

0	0	0
0	1	0
0	0	0



Filtered
(no change)



Простейшие фильтры



Original

0	0	0
0	0	1
0	0	0

?



Простейшие фильтры



Original

0	0	0
0	0	1
0	0	0



Shifted left
By 1 pixel



Простейшие фильтры



Original

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

1	1	1
1	1	1
1	1	1

?



Простейшие фильтры



Original

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

1	1	1
1	1	1
1	1	1

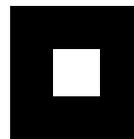


Blur (with a
box filter)



Сглаживание с box-фильтром

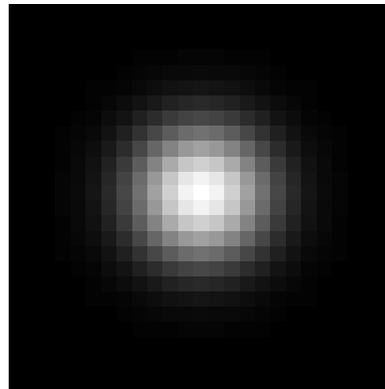
- Результат сглаживания с помощью усреднения отличается от разфокусированного изображения
- Точка света, наблюдаемая с расфокусированного объектива, выглядит как кружок света, а усреднение дает квадратик





Сглаживание

- Точка света, наблюдаемая с расфокусированного объектива, выглядит как кружок света, а усреднение дает квадратик
- Другой способ: взвешивает вклад пикселей по окрестности с учетом близости к центру:

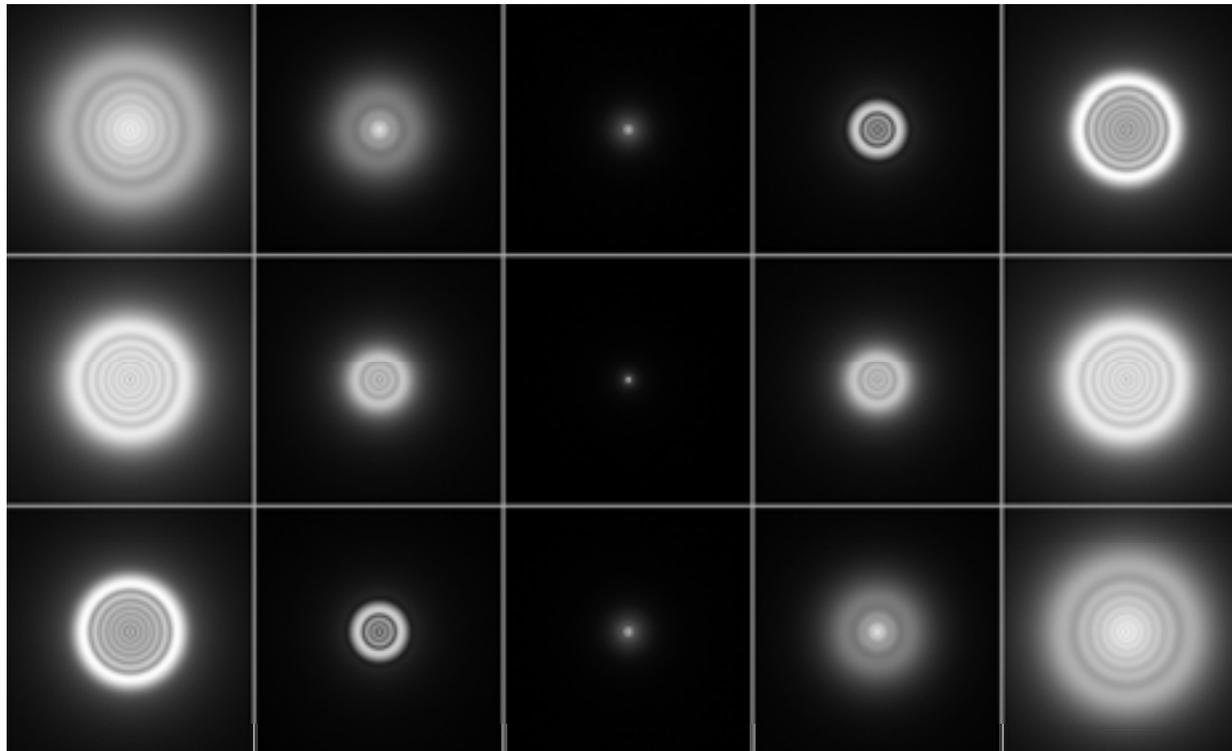


“fuzzy blob”



Point Spread Function (PSF)

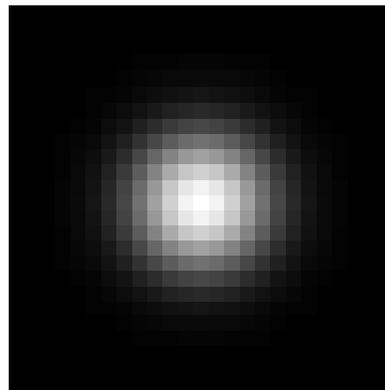
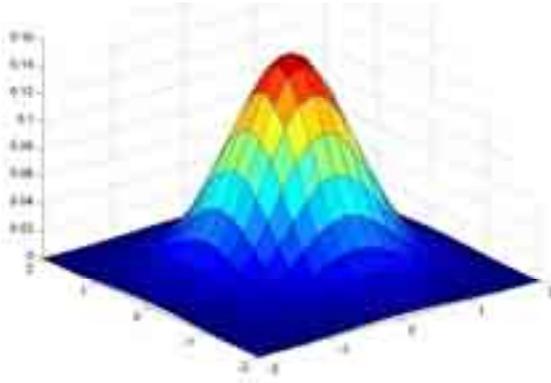
- Point spread function (PSF) – отклик оптической системы на точечный источник света (объект)





Ядро фильтра гаусса

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



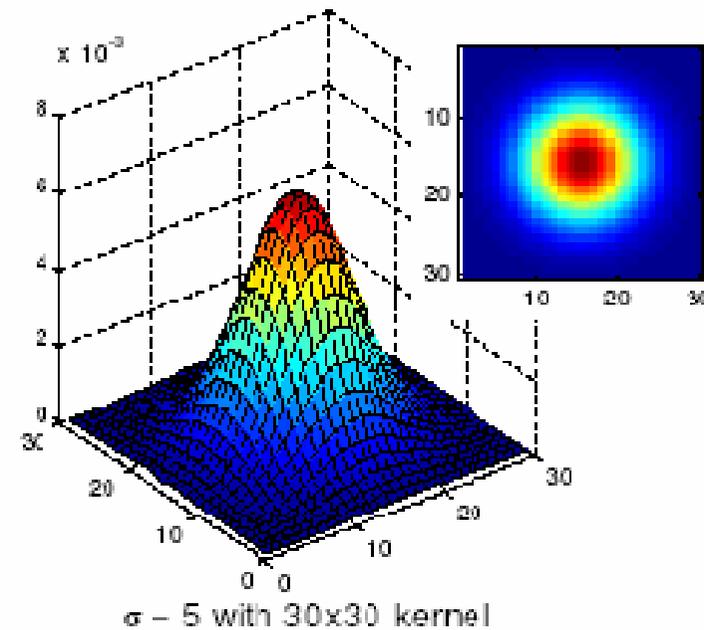
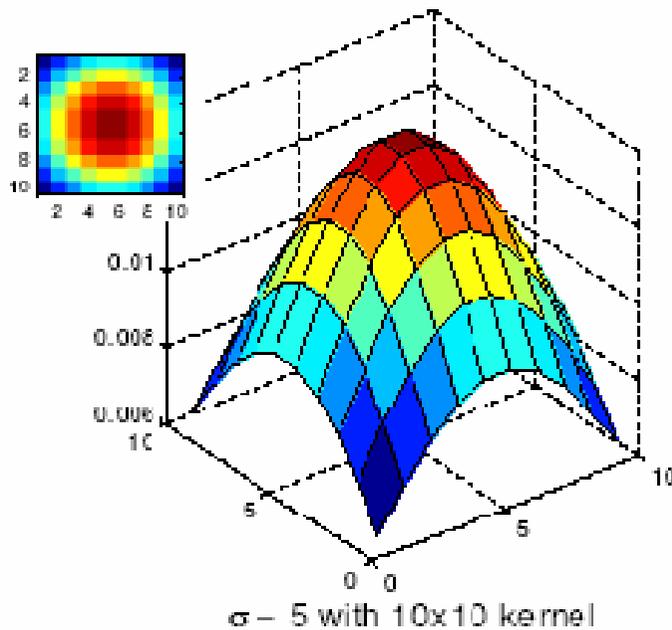
0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, $\sigma = 1$



Выбор размера ядра

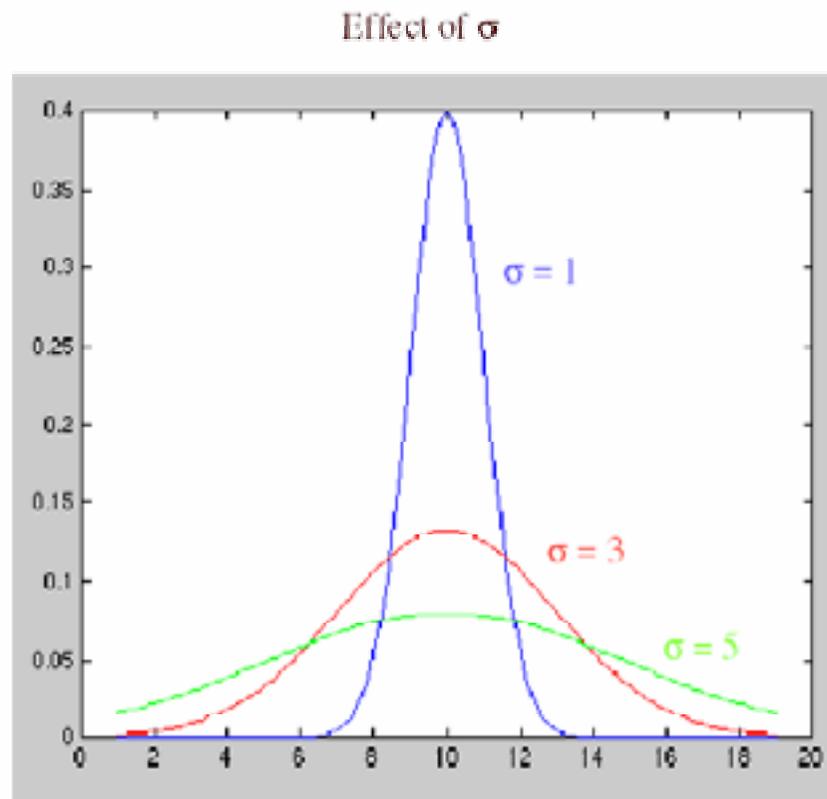
- Размер ядра дискретного фильтра ограничен





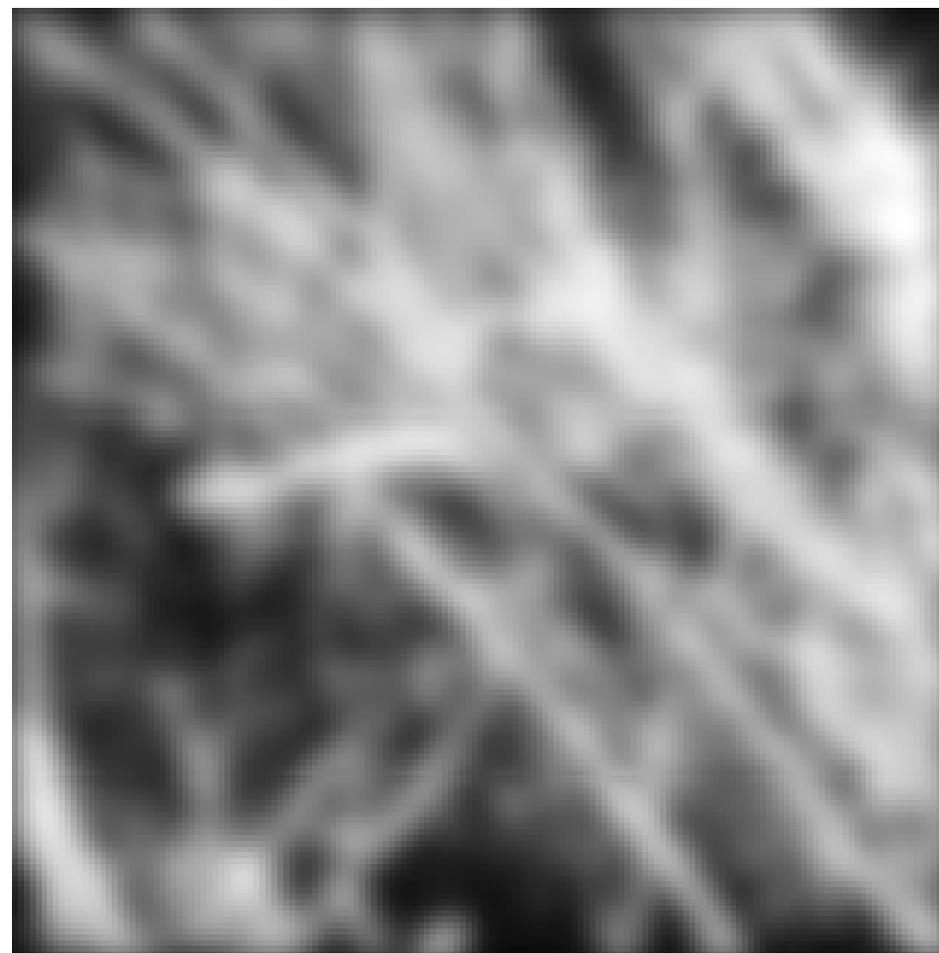
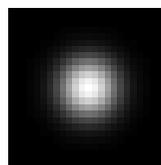
Выбор размера ядра

- Эмпирика: полуразмер фильтра равен 3σ



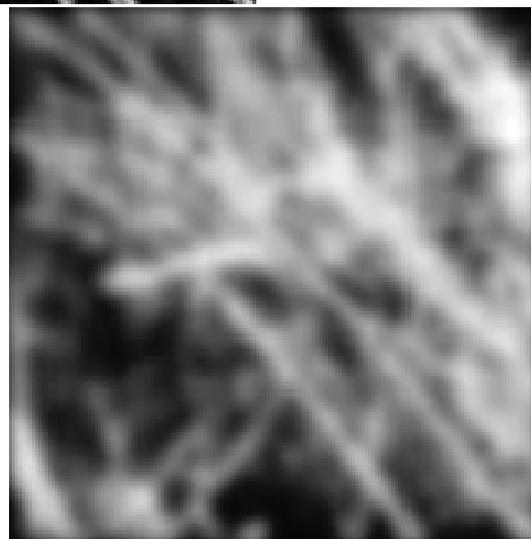
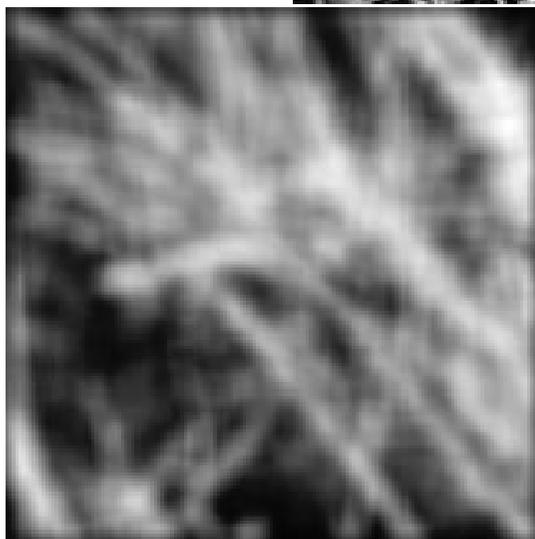


Сглаживание фильтром гаусса





Сравнение



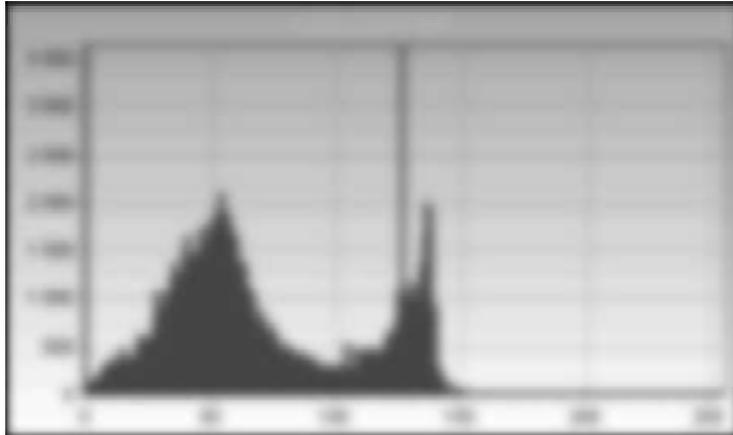


Свойства фильтра Гаусса

- Свертка с сами собой дает тоже фильтр гаусса
 - Сглаживание несколько раз фильтром с маленьким ядром дает результат, аналогичный свертке с большим ядром
 - Свертка 2 раза с фильтром радиуса σ дает тот же результат, что с фильтром радиуса $\sigma\sqrt{2}$

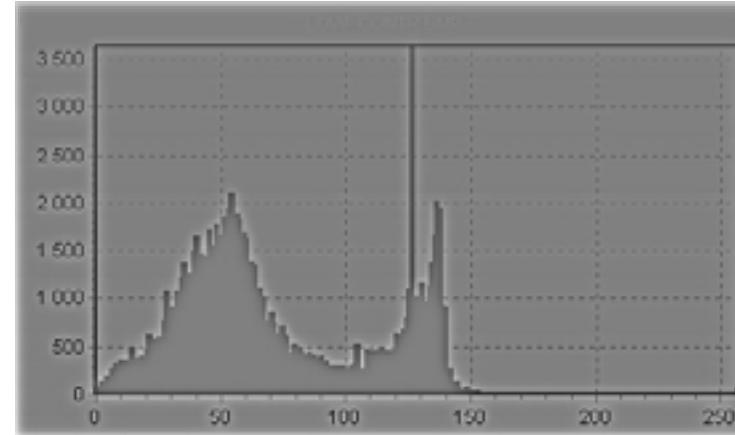


Маленькая экскурсия к Фурье

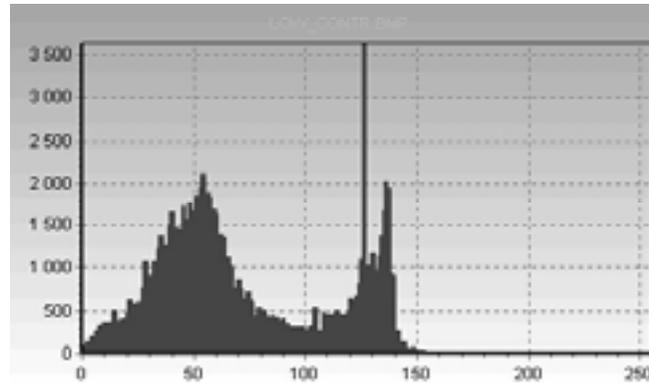


Низкие частоты

+



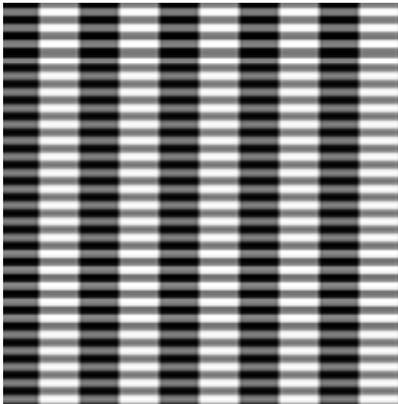
Высокие частоты



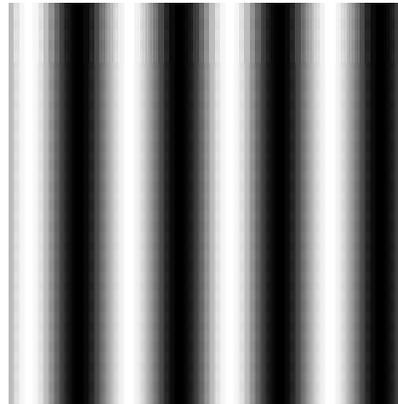


Фильтр Гаусса (gaussian blurring)

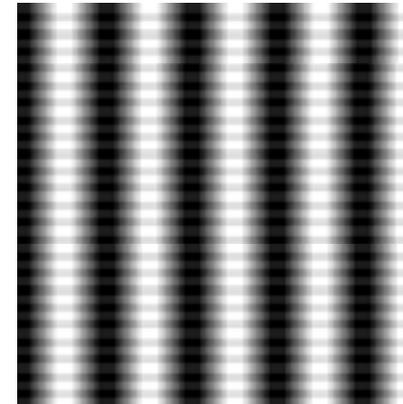
Результат свертки фильтром гаусса и усреднения



Исходное изображение



Фильтр Гаусса с
Sigma = 4



Усреднение по 49
пикселям (7x7)

Важное свойство фильтра Гаусса – он по сути является фильтром низких частот.



Сепарабельность

$$\begin{aligned} G_{\sigma}(x, y) &= \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \\ &= \left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)\right) \left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y^2}{2\sigma^2}\right)\right) \end{aligned}$$

Сепарабельное ядро

Раскладывается в произведение двух одномерных фильтров гаусса



Пример

2D свертка

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 2 & 3 & 3 \\ 3 & 5 & 5 \\ 4 & 4 & 6 \end{bmatrix}$$

Фильтр раскладывается
в произведение двух 1D
фильтров:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

Свертка по строкам:

$$\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 2 & 3 & 3 \\ 3 & 5 & 5 \\ 4 & 4 & 6 \end{bmatrix} = \begin{bmatrix} & 11 & \\ & 18 & \\ & 18 & \end{bmatrix}$$

Затем свертка по столбцу:



Сепарабельность

- Почему сепарабельность полезна на практике?



Виды шума



Original



Salt and pepper noise



Impulse noise



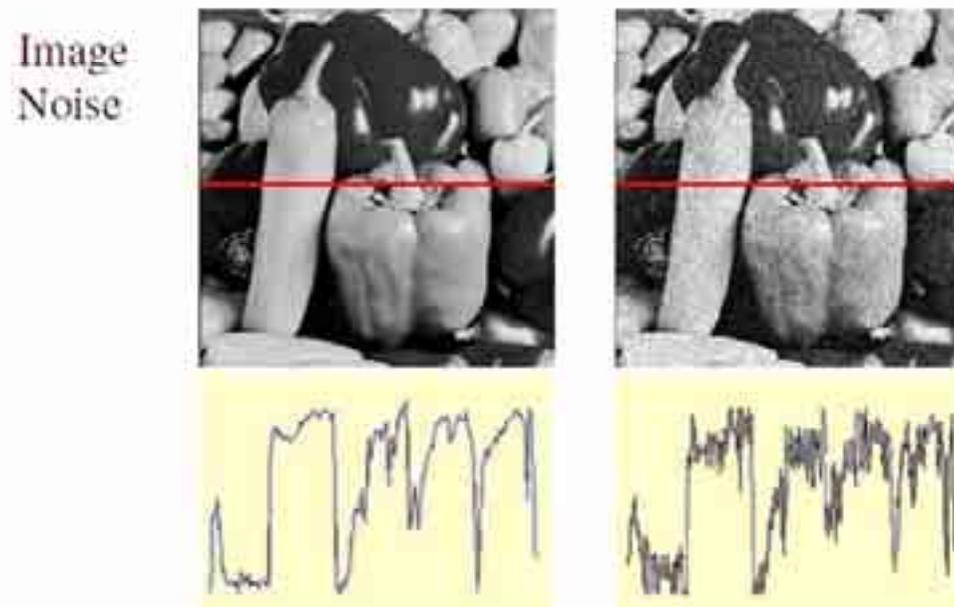
Gaussian noise

- **Соль и перец:** случайные черные и белые пиксели
- **Импульсный:** случайные белые пиксели
- **Гауссов:** колебания яркости, распределенные по нормальному закону



Гауссов шум

- Мат. модель: сумма множества независимых факторов
- Подходит при маленьких дисперсиях
- Предположения: независимость, нулевое матожидание

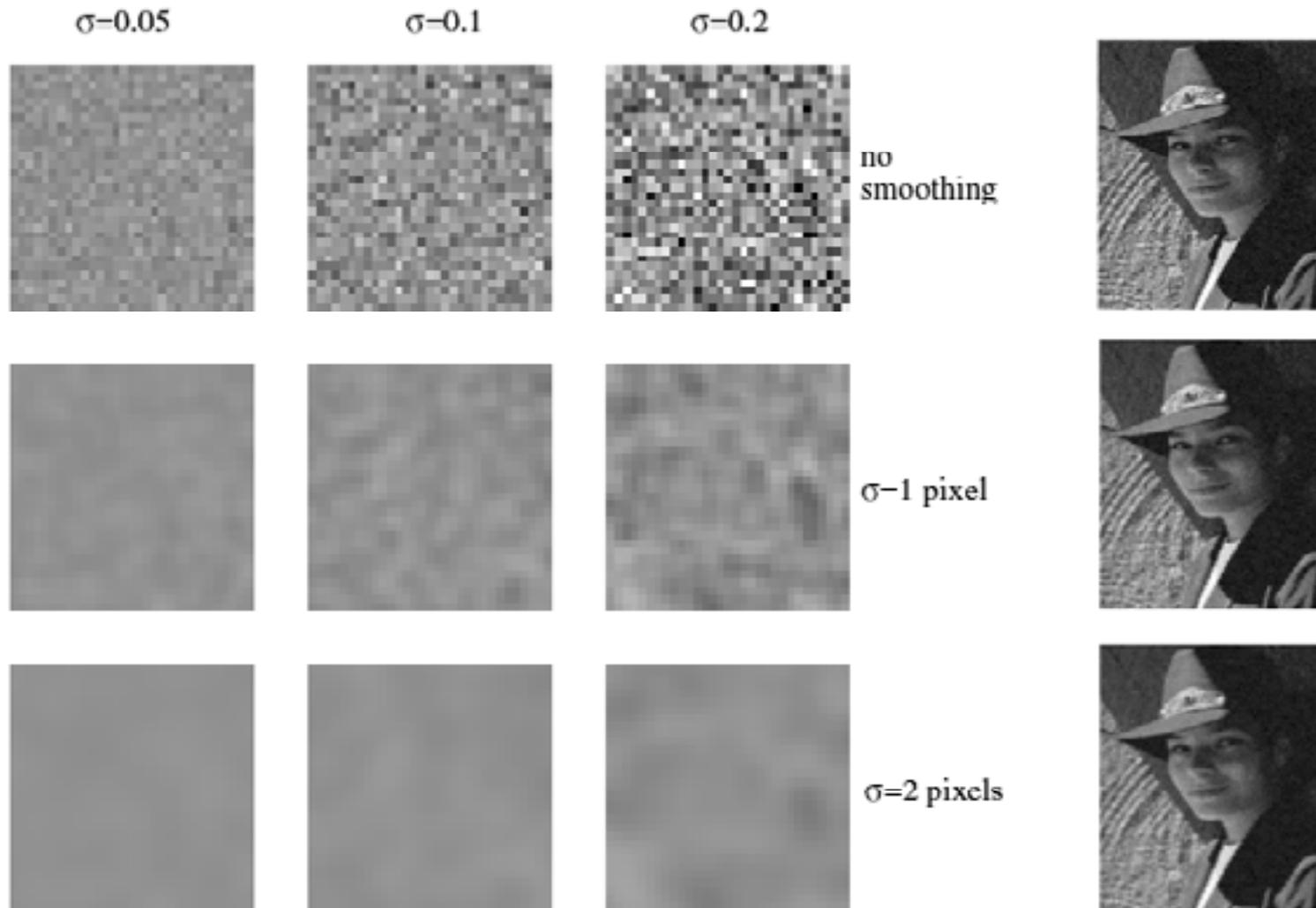


$$f(x, y) = \overbrace{f(x, y)}^{\text{Ideal Image}} + \overbrace{\eta(x, y)}^{\text{Noise process}}$$

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



Подавление гауссова шума



Сглаживание фильтрами большого радиуса подавляет шум, но размывает изображение



Подавление шума «соль и перец»

3x3



5x5



7x7

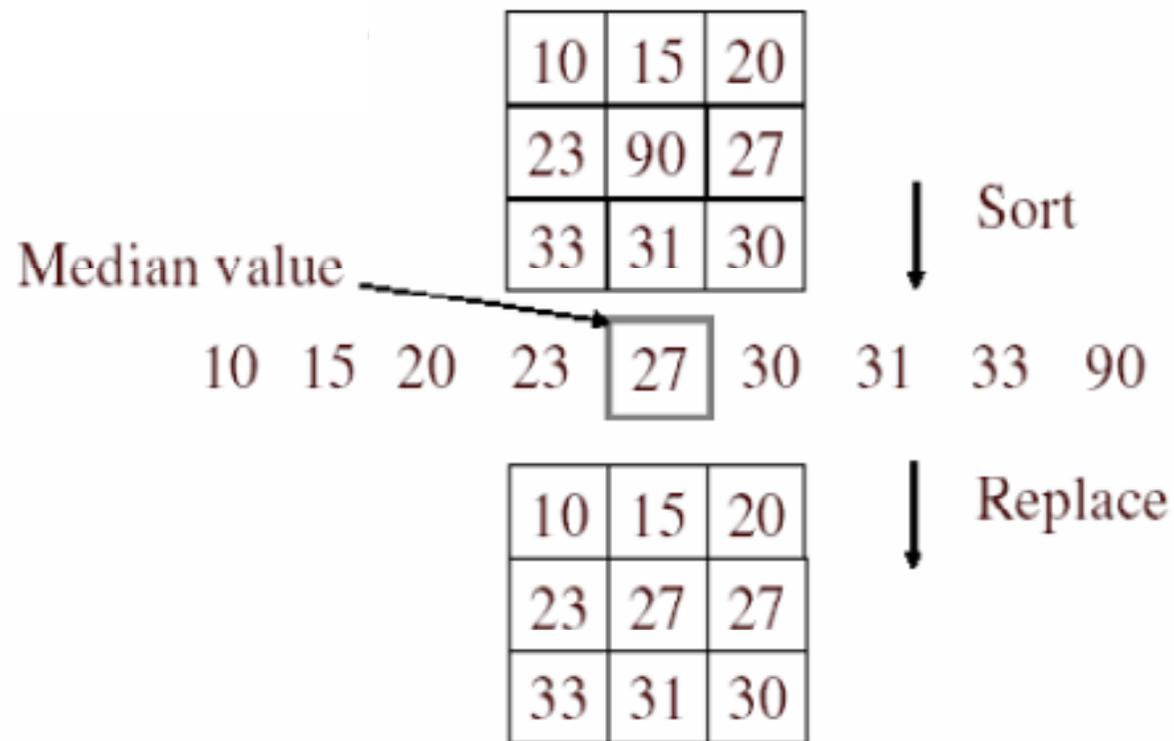


Чем результат плох?



Медианный фильтр

- Выбор медианы из выборки пикселей по окрестности данного



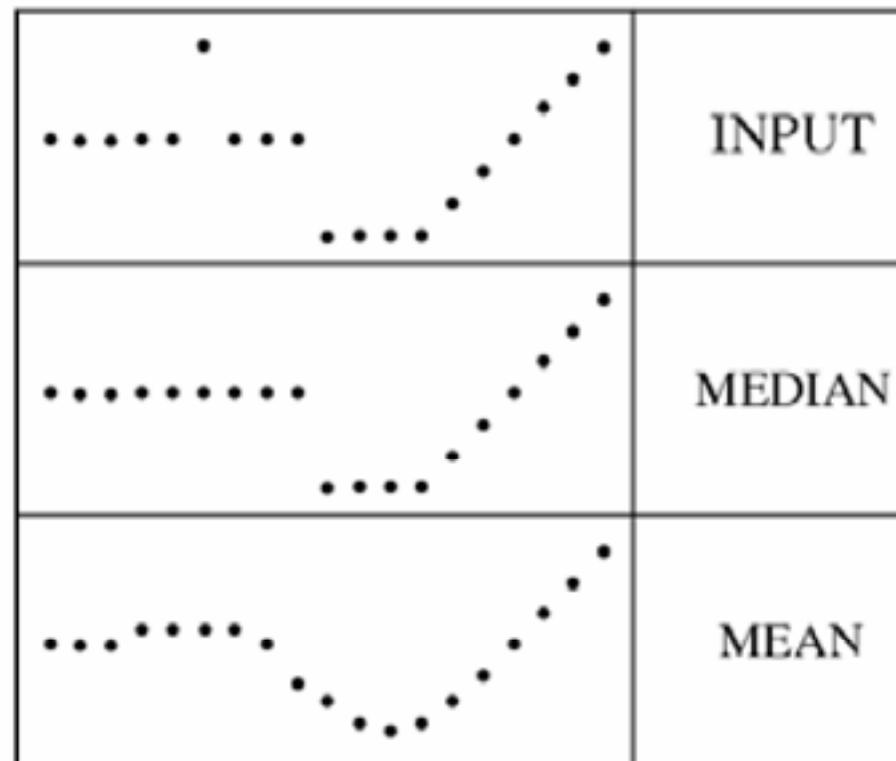
- Является ли фильтр линейным?



Медианный фильтр

- В чем преимущество медианного фильтра перед фильтром гаусса?
 - Устойчивость к выбросам (outliers)

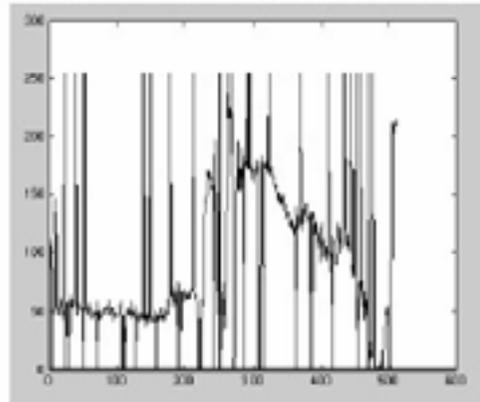
filters have width 5 :



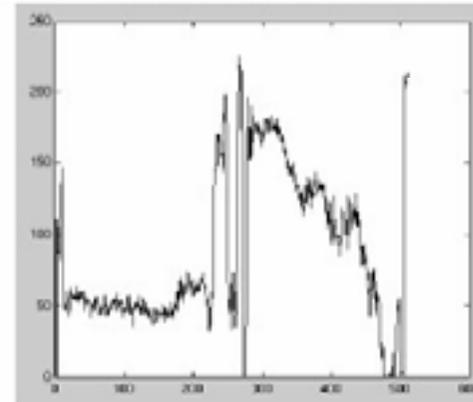


Медианный фильтр

Salt-and-pepper noise



Median filtered

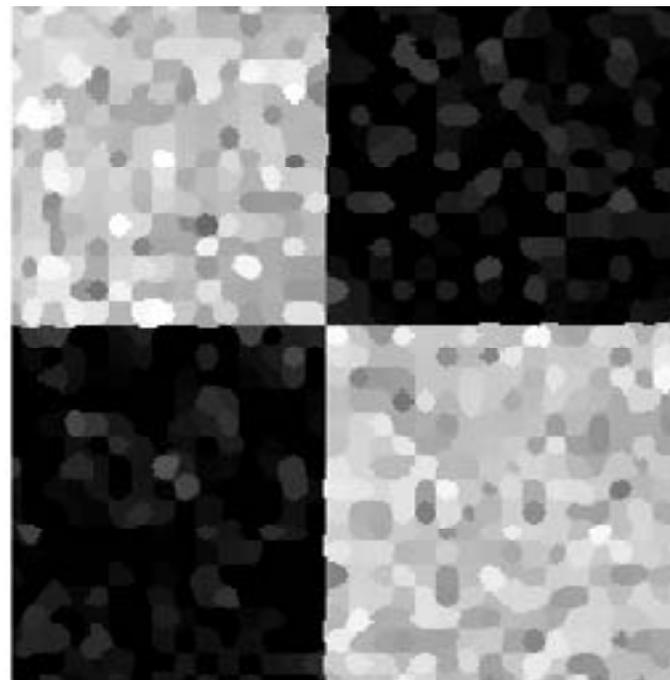
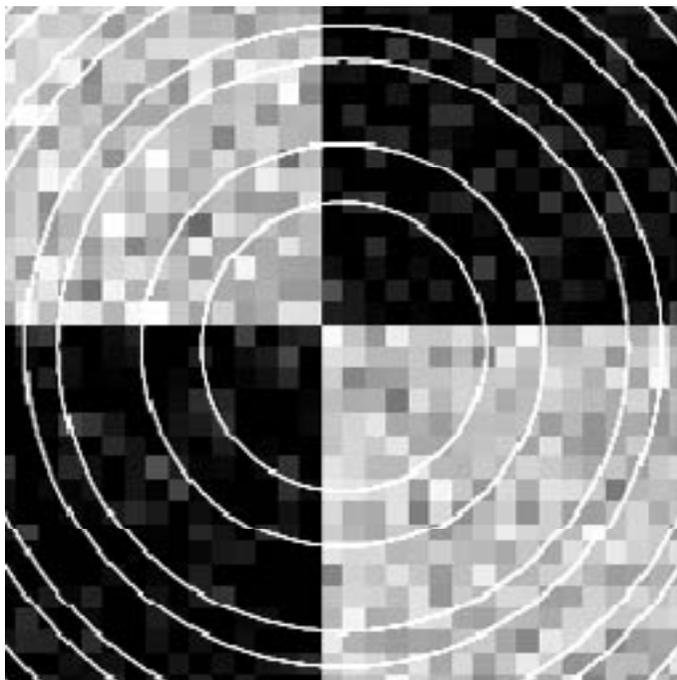


MATLAB: `medfilt2(image, [h w])`



Медианный фильтр

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





Сравнение фильтров

3x3

5x5

7x7

Гауссов



Медианный





Повышение резкости

Что теряется при сглаживании?



Добавим дополнительно высокие частоты:





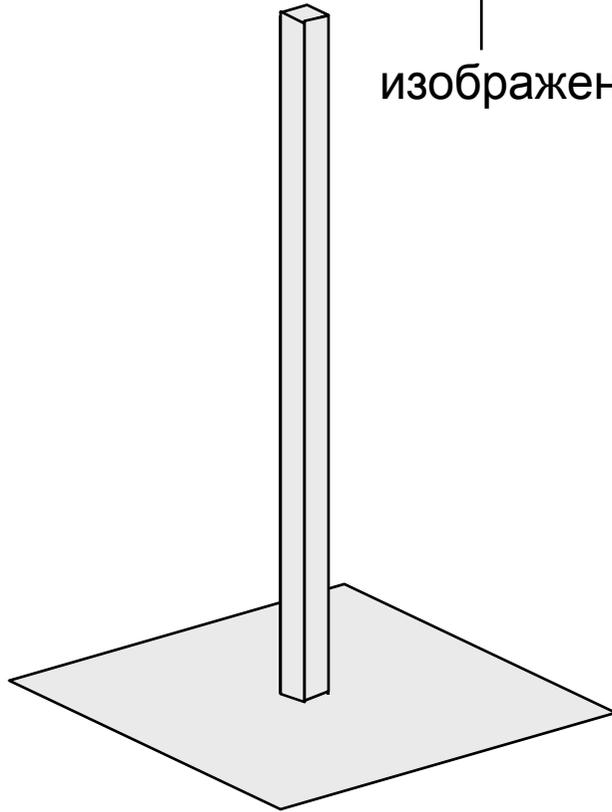
Фильтр Unsharp

$$f + \alpha(f - f * g) = (1 + \alpha)f - \alpha f * g = f * ((1 + \alpha)e - \alpha g)$$

↑
изображение

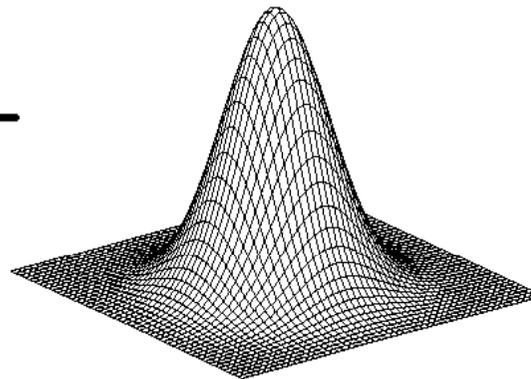
↑
сглаженное
изображение

↑
Единичный
Фильтр



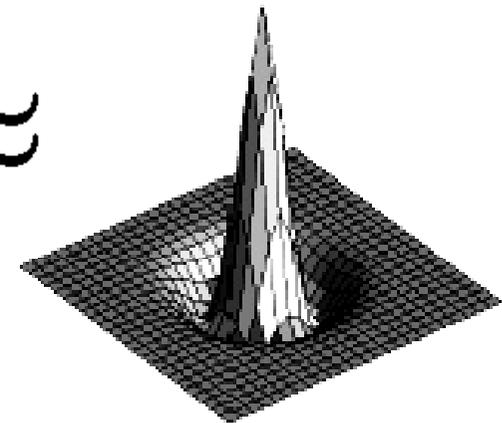
Единичный фильтр

—



Гауссов

≈



Лапласиан гауссиана



Пример

Ядро
свертки

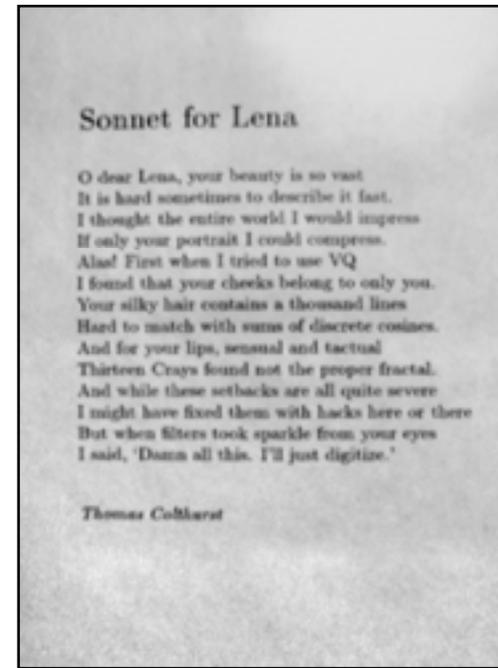
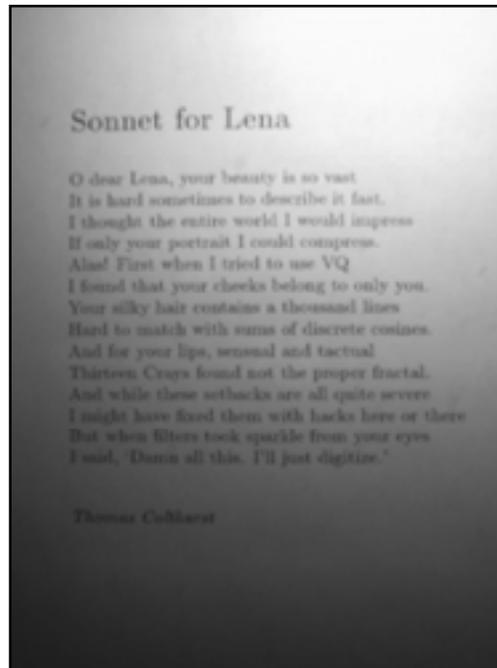
$$\frac{1}{10} \cdot \begin{vmatrix} -1 & -2 & -1 \\ -2 & 22 & -2 \\ -1 & -2 & -1 \end{vmatrix}$$





Компенсация разности освещения

Пример





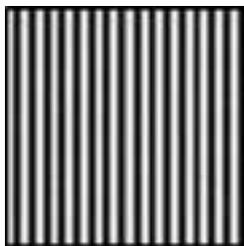
Компенсация разности освещения

Идея:

Формирование изображения:

$$I(i, j) = l(i, j) \cdot r(i, j)$$

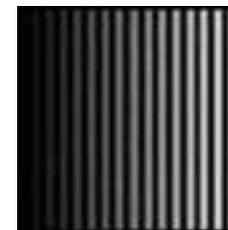
Плавные изменения яркости относятся к освещению,
резкие - к объектам.



объект $r(i, j)$



освещение $l(i, j)$



Изображение
освещенного
объекта $I(i, j)$



Выравнивание освещения

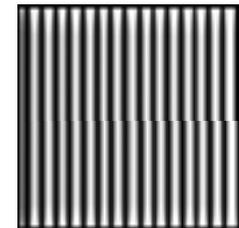
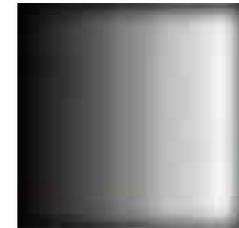
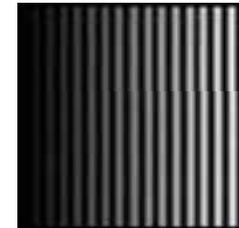
- Алгоритм Single scale retinex (SSR)
 - Получить приближенное изображение освещения путем низочастотной фильтрации

$$\hat{l}(i, j) = G * I(i, j)$$

- Восстановить изображение по формуле

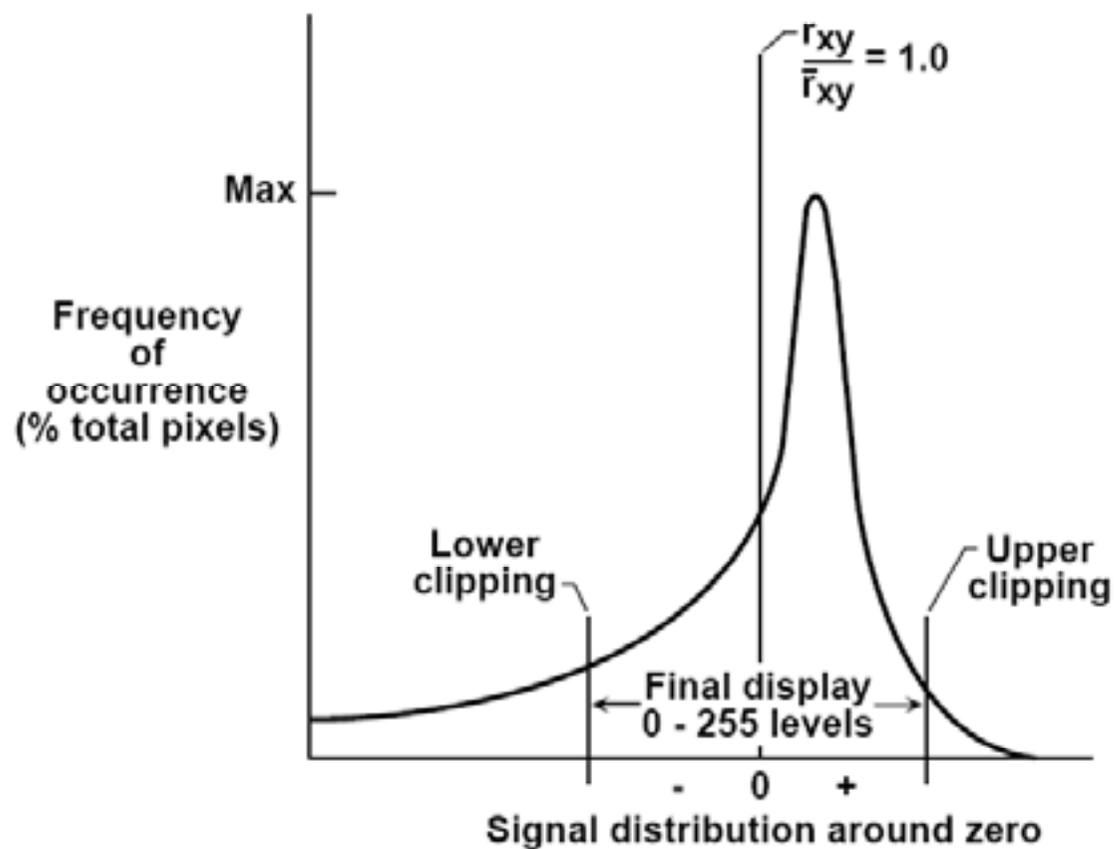
$$\hat{r}(i, j) = \log \frac{I(i, j)}{\hat{l}(i, j)}$$

$$\hat{r}(i, j) = \log I(i, j) - \log \hat{l}(i, j)$$





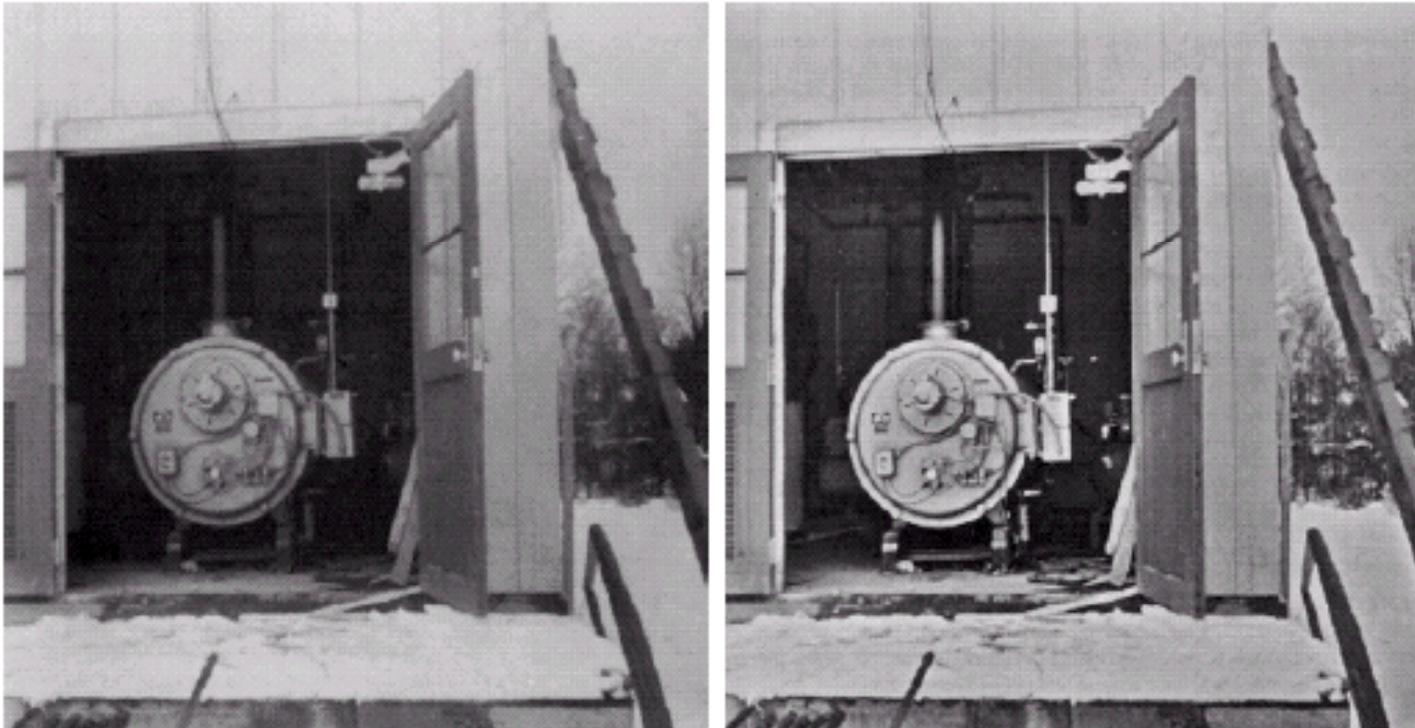
Обрезание по порогу





Выравнивание освещения

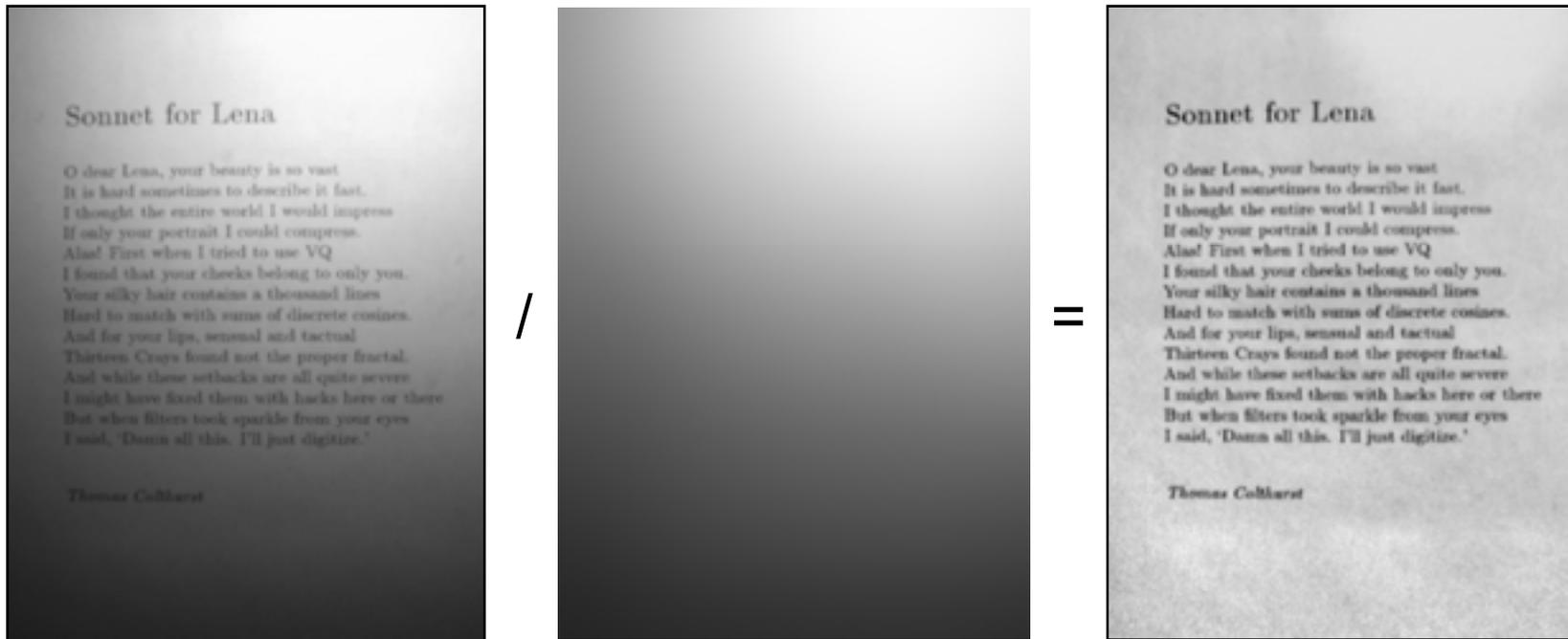
Пример





Компенсация разности освещения

Пример



Gauss 14.7 пикселей



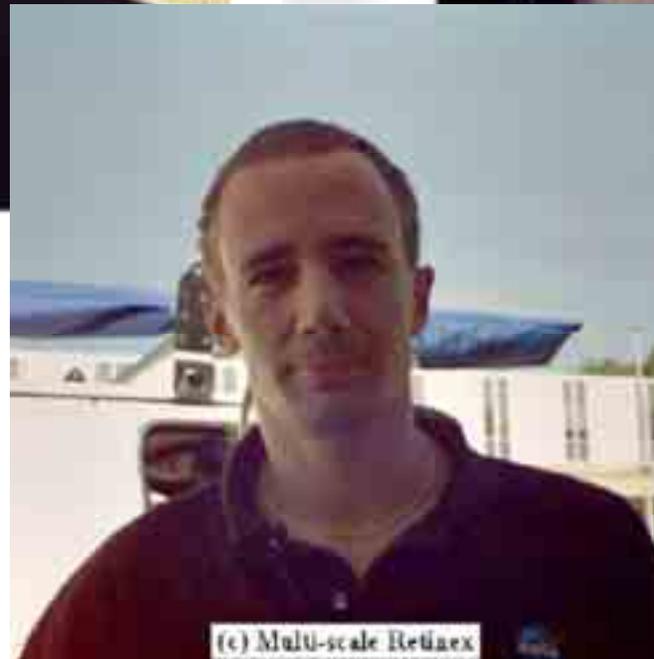
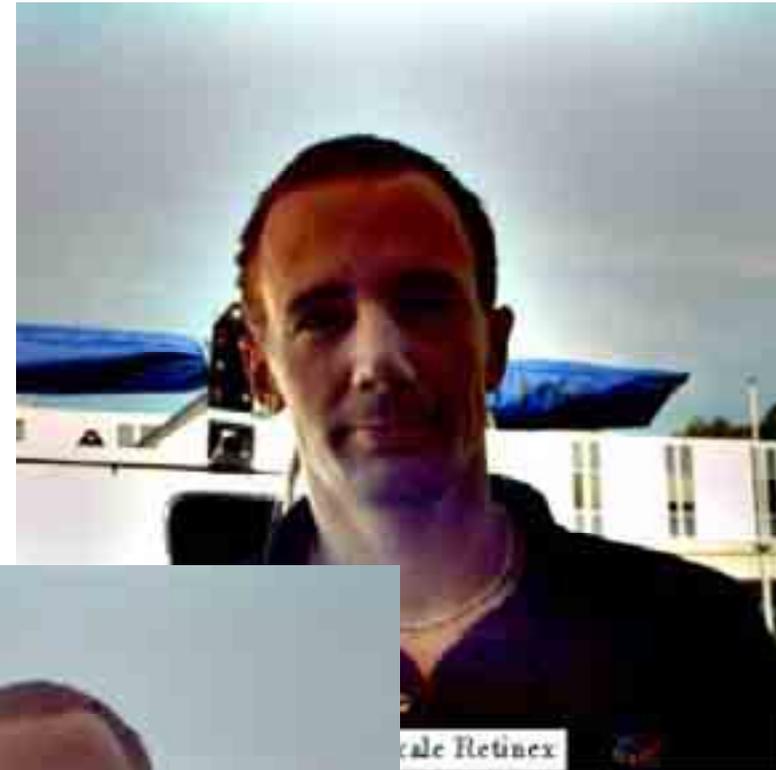
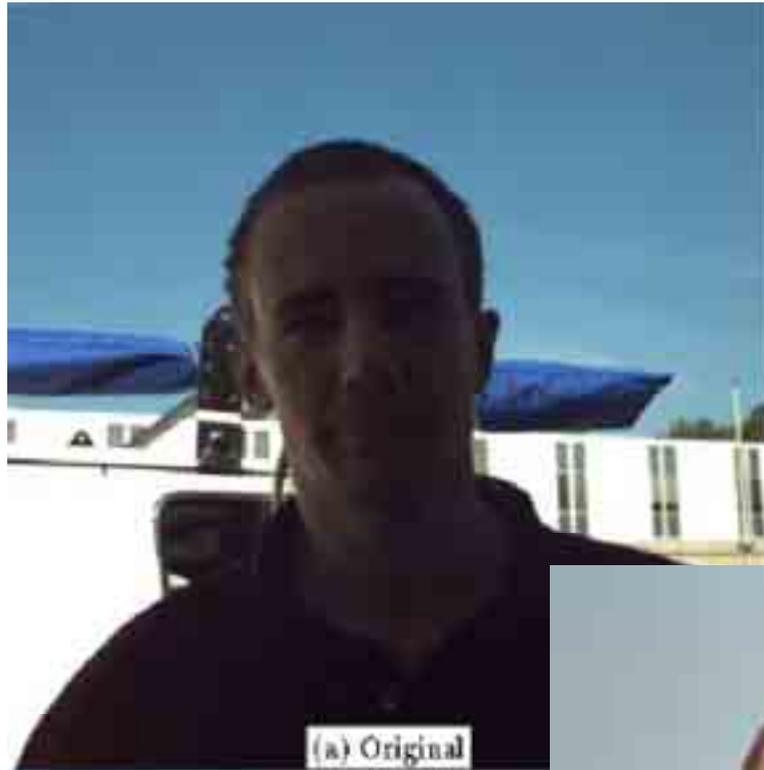
Многомасштабный вариант

$$\hat{r}(i, j) = \sum_k w_k (\log I(i, j) - \log g_k(i, j) \cdot I(i, j))$$

- Чаще всего выбирают 3 масштаба
- Веса одинаковые (1/3)



Multi-scale retinex





Метрики качества

- Как измерить похожесть двух изображений?
 - Для оценки качества подавления шума, например



исходное
изображение



искаженное
изображение



Метрики качества

- Среднеквадратичная ошибка (MSE)

$$MSE = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2$$

N – число пикселей

- Пиковое отношение сигнал/шум (PSNR)

$$PSNR_{dB} = 10 \lg \frac{M^2}{MSE}$$

M – максимальное значение пикселя



Метрики качества

- PSNR и MSE не учитывают особенности человеческого восприятия!



Оригинал



Метрики качества

- У этих изображений одинаковые PSNR с оригиналом (примерно 25 dB)



Повышена контрастность

Добавлен белый гауссов шум

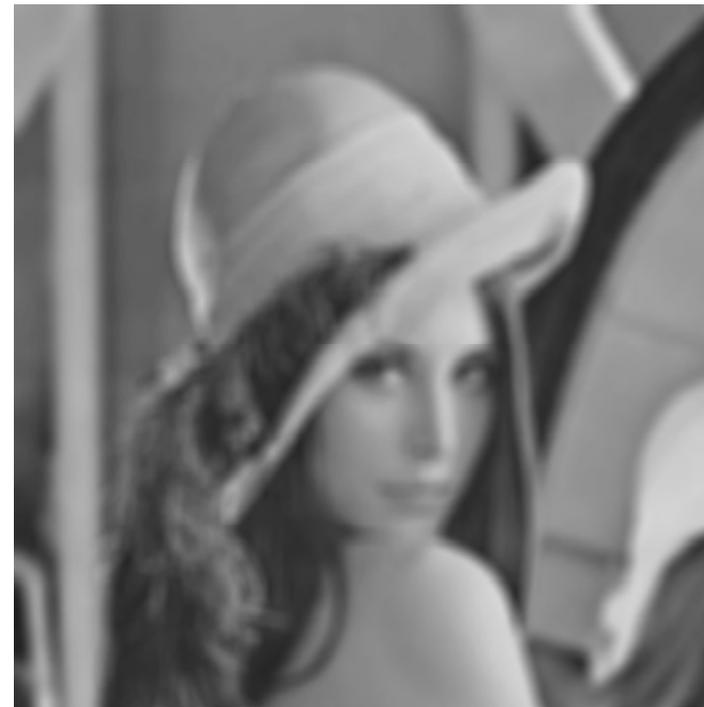


Метрики качества

- И у этих – тоже примерно 25 dB!



Добавлен импульсный шум



Размытие



Метрики качества

- И у этого – тоже!



Артефакт блочности после JPEG



Метрики качества

- Вывод: PSNR не всегда отражает реальный видимый уровень искажений.
- Как улучшить?

HVS models
(human visual system)

- Использовать функцию чувствительности глаза к различным частотам (CSF)
- Использовать свойство маскировки
- Использовать равномерные к восприятию цветовые пространства (CIE Lab, CIEDE2000)



Спецэффекты

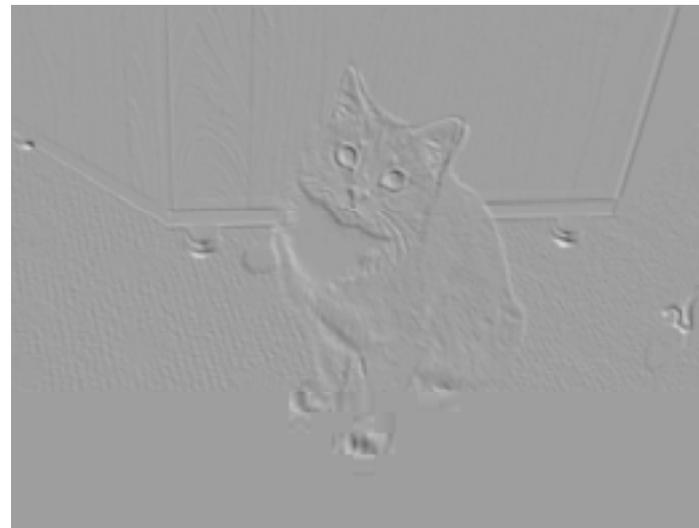
- Рассмотрим
 - Тиснение
 - Негатив
 - «Светящиеся» края
 - Геометрические эффекты
 - Перенос/поворот
 - Искажение
 - «Эффект стекла»



Тиснение

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

Фильтр + сдвиг яркости, нормировка...





Цифровой негатив



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



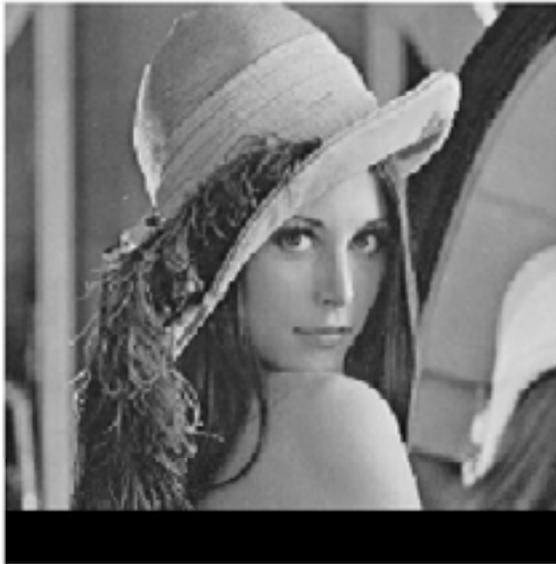
Светящиеся края



Медианный фильтра + выделение краев + фильтр
«максимума»



Перенос/поворот



Перенос:

$$x(k; l) = k + 50; y(k; l) = l;$$

Поворот:

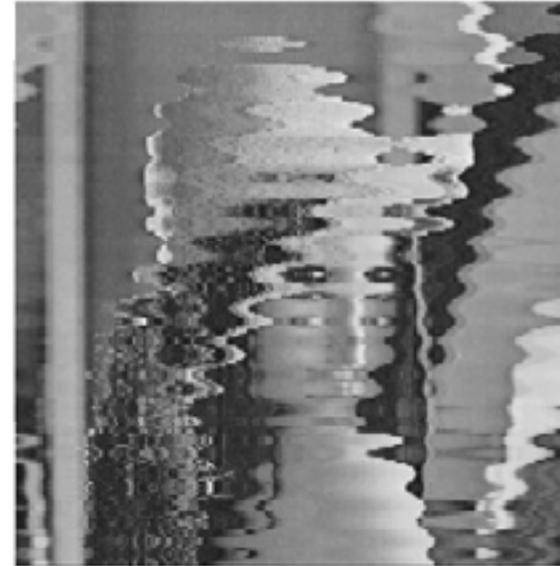
$$x(k; l) = (k \cdot x_0)\cos(\mu) + (l \cdot y_0)\sin(\mu) + x_0;$$

$$y(k; l) = (k \cdot x_0)\sin(\mu) + (l \cdot y_0)\cos(\mu) + y_0;$$

$$x_0 = y_0 = 256.5 \text{ (центр поворота), } \mu = \pi/6$$



«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»


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



На следующей лекции

- Старые-добрые методы распознавания объектов
- Сопоставление шаблонов
- Выделение краёв
- Выделение контрастных объектов
- Геометрические и фотометрические инварианты