



Основы обработки изображений



Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sense of discrete cosines.
And for your lips, sensual and tactical
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Down all this. I'll just digitize.'

Thomas Collier

Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sense of discrete cosines.
And for your lips, sensual and tactical
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Down all this. I'll just digitize.'

Thomas Collier



Антон Конушин

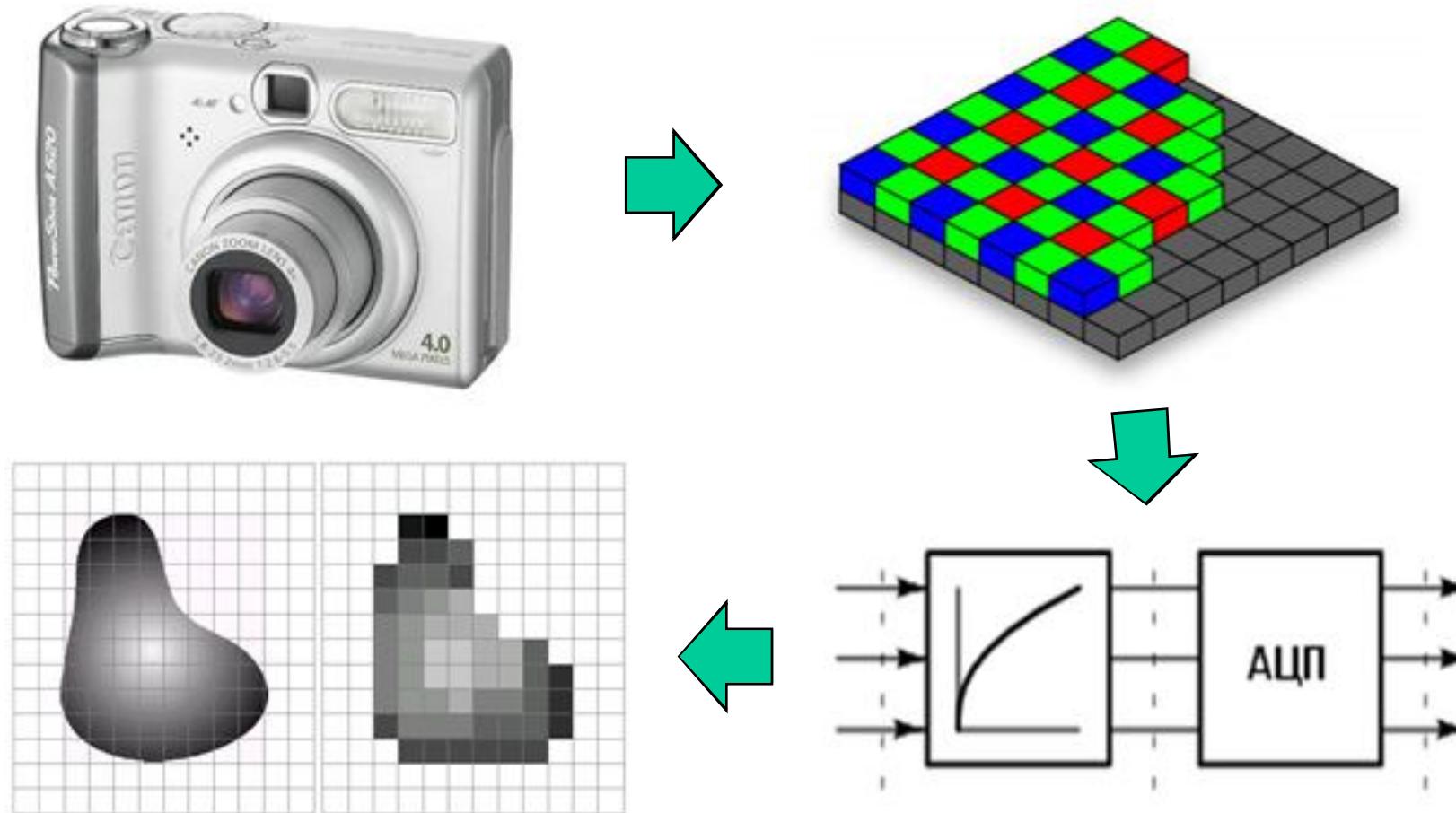


Обработка изображений

- Семейство методов и задач, где входной и выходной информацией являются изображения.
- 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



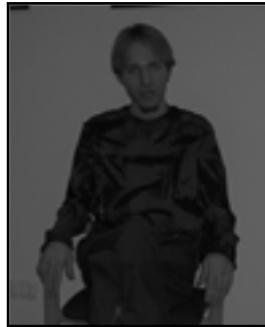
Цифровое изображение



Вспоминаем процесс получения цифрового изображения...



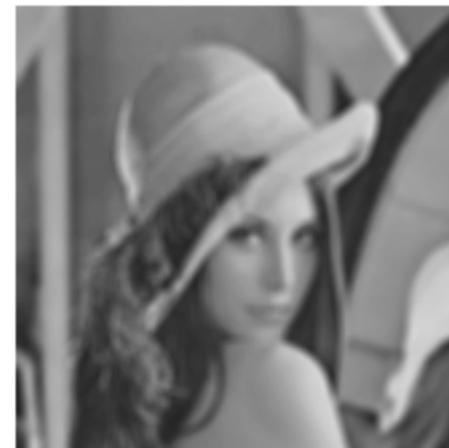
Что может получиться плохо?



Темное или слабоконтрастное

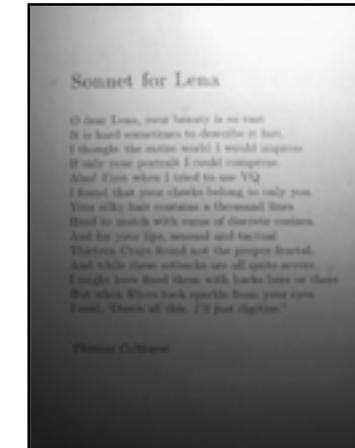


Неправильные цвета



Шумное

Нерезкое



Неравномерно
освещённое



Постоянство цвета и освещенности



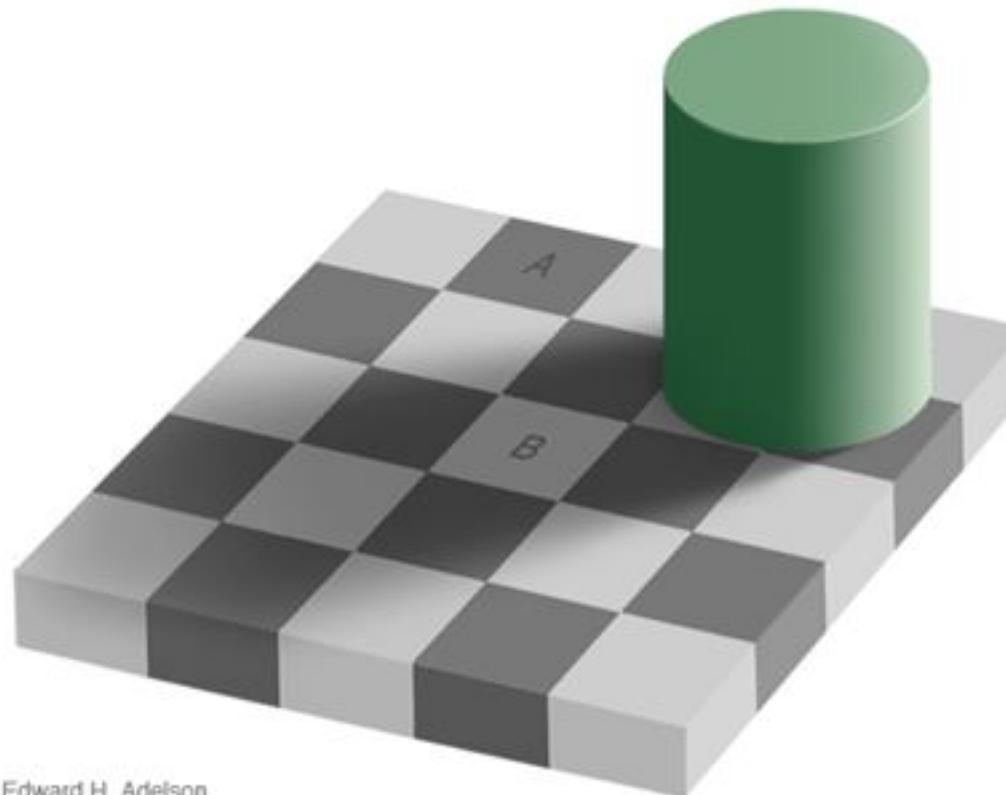
Способность
зрительной системы
человека оценивать
собственные
отражательные
свойства
поверхностей в не
зависимости от
условий
освещенности

Пример:
Белый цвет на
свету и в тени

J. S. Sargent, The Daughters of Edward D. Boit, 1882



ПОСТОЯНСТВО ЯРКОСТИ



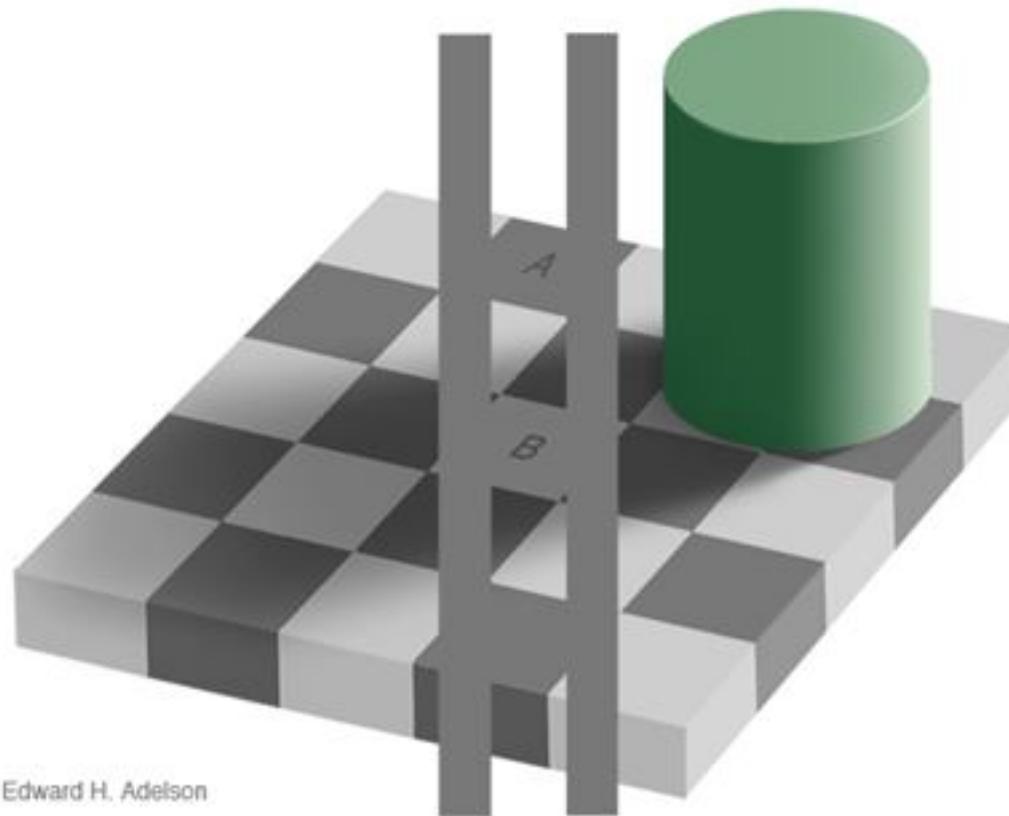
Edward H. Adelson

http://web.mit.edu/persci/people/adelson/checkershadow_illusion.html

Slide by S. Lazebnik



ПОСТОЯНСТВО ЯРКОСТИ



Edward H. Adelson

http://web.mit.edu/persci/people/adelson/checkershadow_illusion.html

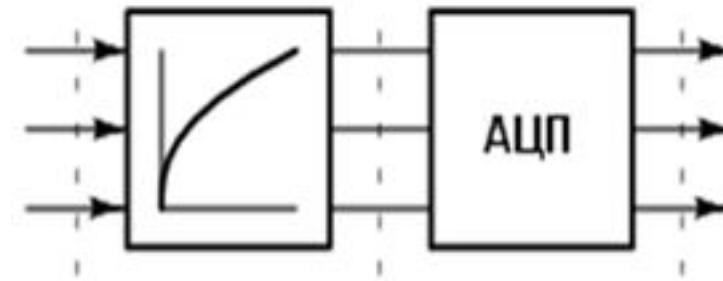
Slide by S. Lazebnik



Коррекция контраста изображения

Почему изображение может быть слабоконтрастным?

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



Как мы можем численно оценить, контрастным получилось изображение или нет?



Гистограмма яркости

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 «пересвеченое» изображение)
- Концентрация яркостей вокруг определенных значений, неравномерное заполнение диапазона яркостей (узкий диапазон - тусклое изображение)

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

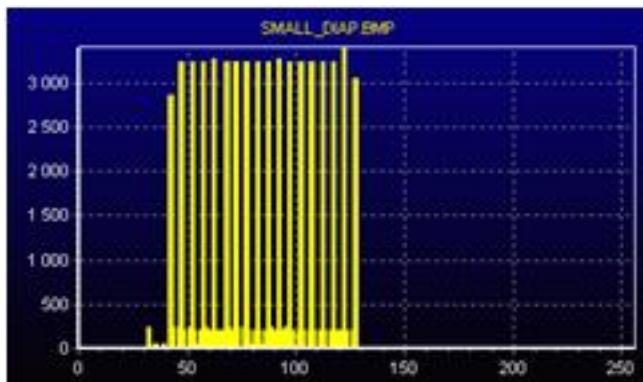
$$f^{-1}(y) = x \quad y - \text{яркость пикселя на исходном изображении}, \\ x - \text{яркость пикселя после коррекции}.$$





Линейная коррекция

Компенсация узкого диапазона яркостей –
линейное растяжение:



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

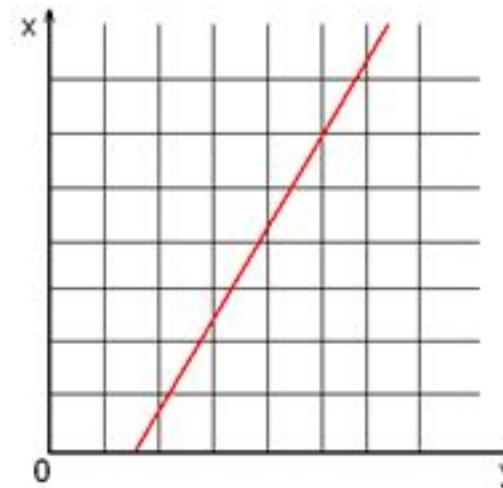
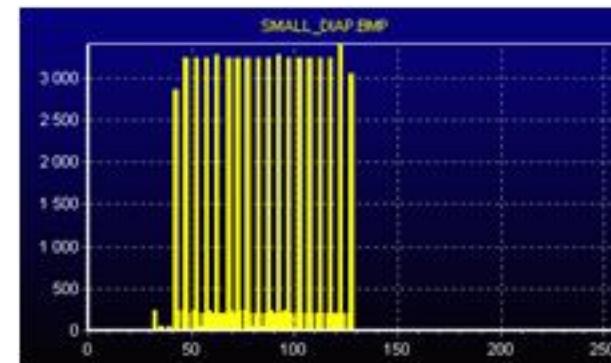


График функции $f^{-1}(y)$

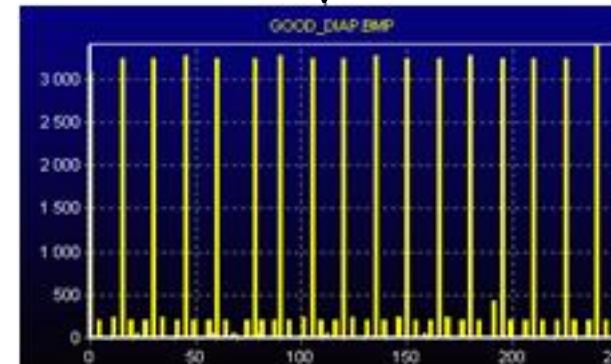


Линейная коррекция

Компенсация узкого диапазона яркостей – линейное растяжение:



SMALL_DIAPIBM





Робастная линейная коррекция

Что будет при применении линейной коррекции к такой картинке?



Робастная (устойчивая) версия метода:

- Вычислим такую линейную коррекцию, чтобы 5% самых темных пикселов стали черными и 5% самых светлых стали белыми



Линейная коррекция

Линейное растяжение – «как AutoContrast в Photoshop»





Линейная коррекция



Линейная коррекция помогает не всегда!



Нелинейная коррекция

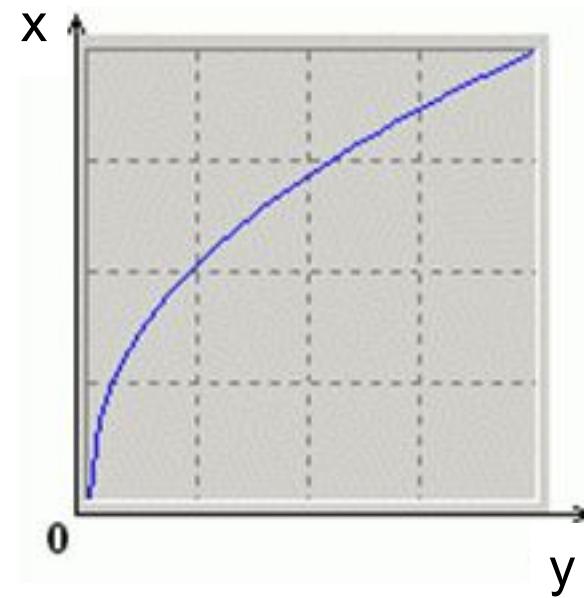


График функции $f^{-1}(y)$



Нелинейная коррекция

Нелинейная компенсация недостаточной контрастности

Часто применяемые функции:

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

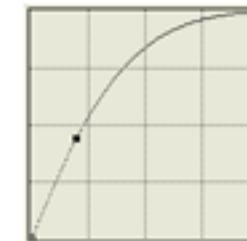
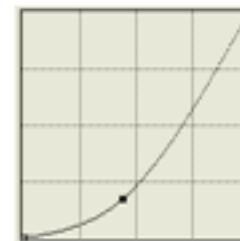
$$y = c \cdot x^\gamma$$

- Логарифмическая
 - Цель – сжатие динамического диапазона при визуализации данных

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



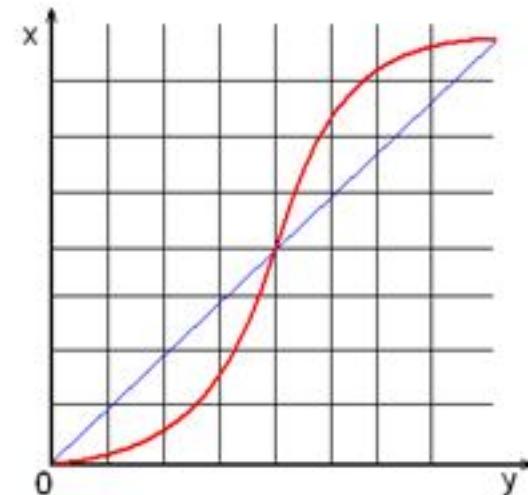
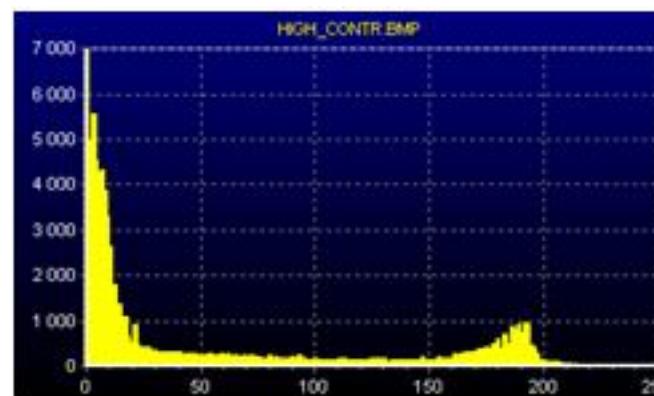
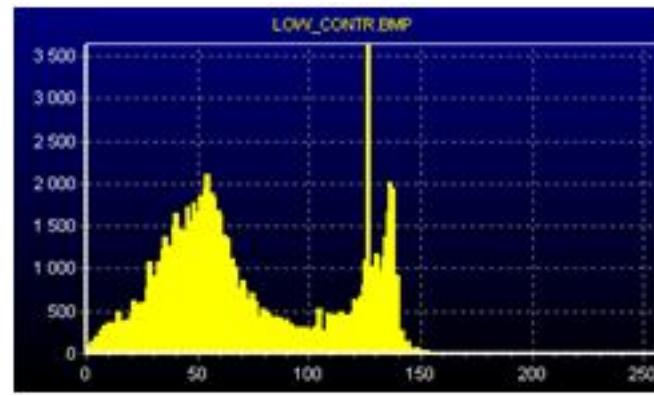
Гамма-коррекция



Графики функции $f^{-1}(y)$



Нелинейная коррекция





Цветовой баланс («баланс белого»)

- Когда мы смотрим на фотографию или монитор, глаза адаптируются к освещению в комнате, а не к освещению сцены на фотографии
- Если «баланс белого» неточен, цвета фотографии кажутся неестественными

Неправильный баланс



Правильный баланс



<http://www.cambridgeincolour.com/tutorials/white-balance.htm>



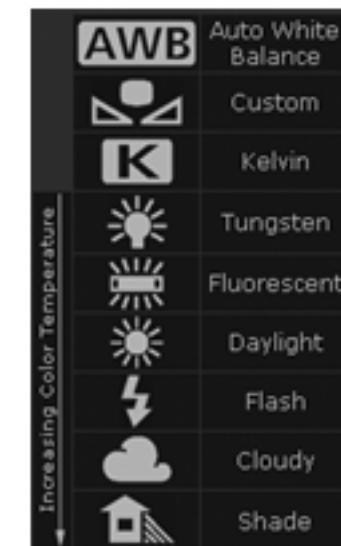
Баланс белого

Пленочные камеры:

- Разные виды пленки и светофильтры применяются для разных сцен

Цифровые камеры:

- Автоматический баланс белого
- Предустановки баланса белого для типичных условий съемки



Задача очень непростая!



Серые карточки

- Простейший способ:
 - Сфотографировать объект с известным цветом
 - Домножить каждый канал на коэффициент, чтобы цвет объекта на фотографии совпал с нужным
- Пример: серые (белые) карточки
 - Фотографируем нейтральный объект (белый)
 - Оцениваем вес каждого канала
 - Если цвет объект записывается как r_w, g_w, b_w тогда веса $1/r_w, 1/g_w, 1/b_w$

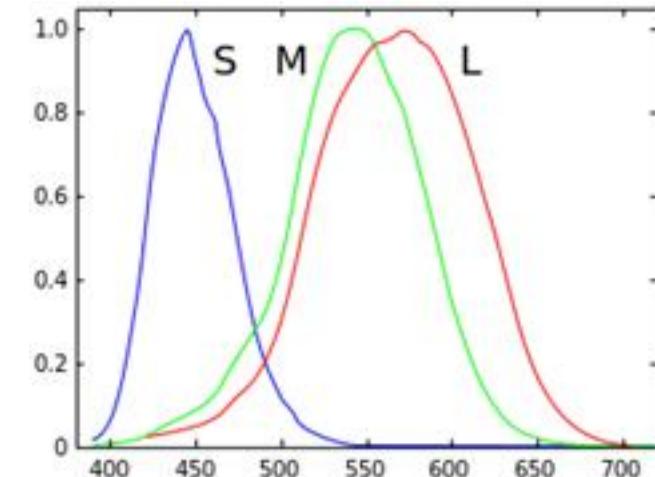


Корректно ли это с точки зрения человеческого зрения и камеры?



Адаптация «Von Kries»

- Модель коррекции, основанная на человеческом восприятии
- У человека может меняться чувствительность каждого типа колбочек
- Схема алгоритма:
 - Перевести из RGB в LMS пространство
 - Вычислить преобразование
 - Перевести обратно



- L(long)
- M(medium)
- S(short)

$$L = 0.3811 \cdot R + 0.5783 \cdot G + 0.0402 \cdot B$$

$$M = 0.1967 \cdot R + 0.7244 \cdot G + 0.0782 \cdot B$$

$$S = 0.0241 \cdot R + 0.1288 \cdot G + 0.8444 \cdot B$$

$$R = 4.4679 \cdot L - 3.5873 \cdot M + 0.1193 \cdot S$$

$$G = -1.2186 \cdot L + 2.3809 \cdot M - 0.1624 \cdot S$$

$$B = 0.0497 \cdot L - 0.2439 \cdot M + 1.2045 \cdot S$$



Профессиональная цветокоррекция



Source: The dark knight



Source: <http://x-rite.com>

Используем цветной шаблон с
многими цветами
Какое преобразование в камере?



Базы изображений



<http://vision.middlebury.edu/color/>

- Авторы собрали большую коллекцию разных изображений для оценки различных моделей преобразования в камере
- Полиномиальная модель (24 параметра)

$$y_i = g_i([M_D k]_i)$$

A. Chakrabarti, D. Scharstein, and T. Zickler. [An empirical camera model for Internet color vision](#). BMVC 2009



Оценка параметров цветокоррекции

Если нет цветовых шаблонов, тогда нам нужно угадать (или оценить) коэффициенты усиления

Модель «Серого мира» (Grayworld)

- Средний уровень («серый») по каждому каналу должен быть одинаков для всех каналов
- Если цветовой баланс нарушен, тогда «серый» в этом канале больше «серого» других каналов
- Вычислим коэффициенты усиления так, чтобы среднее в каждом канале стало одинаковым:

$$\bar{R} = \frac{1}{N} \sum R(x, y); \quad \bar{G} = \frac{1}{N} \sum G(x, y); \quad \bar{B} = \frac{1}{N} \sum B(x, y); \quad Avg = \frac{\bar{R} + \bar{G} + \bar{B}}{3};$$

$$R' = R \cdot \frac{Avg}{\bar{R}}; \quad G' = G \cdot \frac{Avg}{\bar{G}}; \quad B' = B \cdot \frac{Avg}{\bar{B}}$$



«Серый мир» - примеры





«Серый мир» - примеры





«Серый мир» - примеры





Варианты «угадывания»

Модель блика

- Цвета бликов обычно соответствуют цвету источника
- Ищем самый яркий пиксель, и веса берем обратно пропорционально его цветам

Сопоставление диапазона (Gamut)

- Gamut: выпуклая оболочка цветов всех пикселей
- Выбираем преобразование, которое переводит диапазон изображения к «стандартному» при дневном освещении



Цветовая коррекция изображений

- Растяжение контрастности (“autolevels”)
 - Идея – растянуть интенсивности по каждому из каналов на весь диапазон;
- Метод:
 - Найти minimum, maximum по каждому из каналов:
 $R_{\min}, R_{\max}, G_{\min}, G_{\max}, B_{\min}, B_{\max}$
 - Преобразовать интенсивности:

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

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

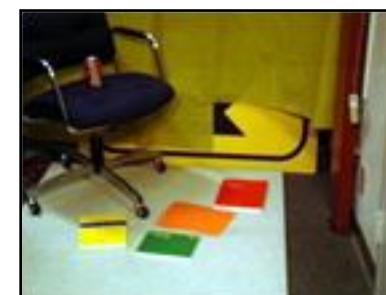


Растяжение контрастности



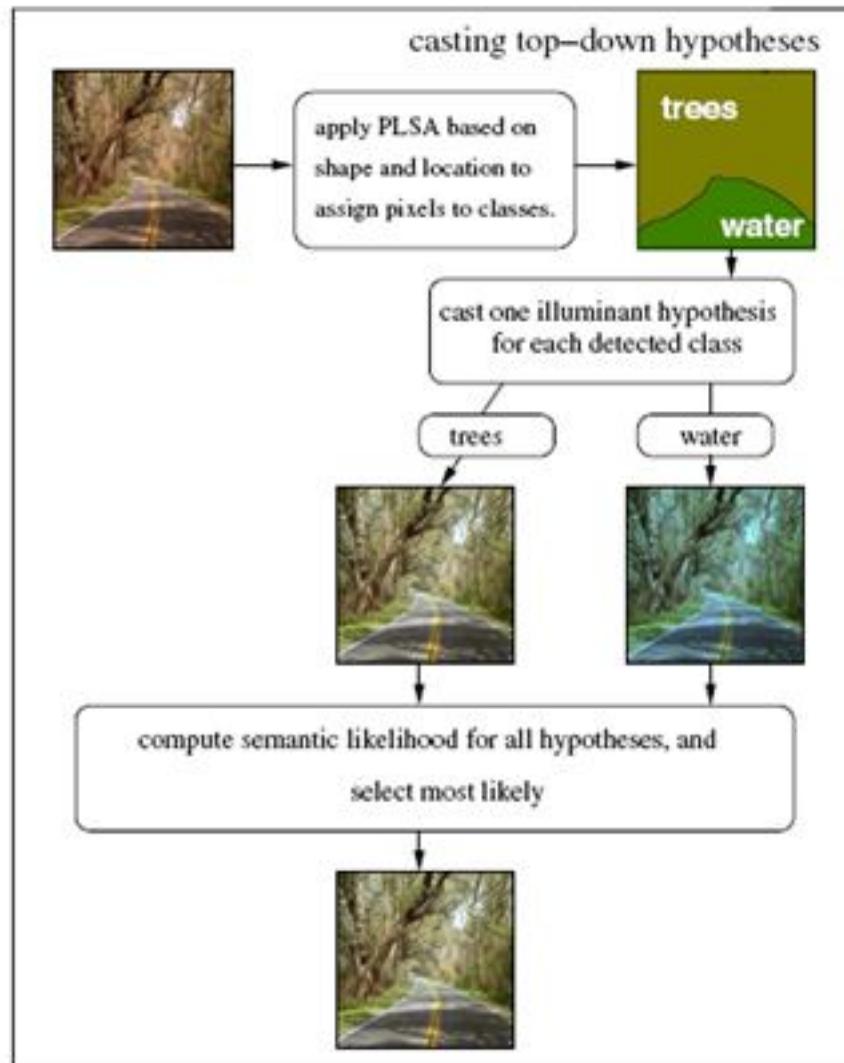


Растяжение контрастности





Распознавание баланса белого



Методы цветовой коррекции до сих пор развиваются.

Пример: Для каждого класса объектов, присутствующих в сцене, вычисляем преобразование таким образом, чтобы диапазон цветов объекта совпадал со средним диапазоном объектов этого класса на «типовых» изображениях



Шумоподавление

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



Шум фотоаппарата



Исходное изображение



Сильное сжатие JPEG



Виды шума



Original



Salt and pepper noise



Impulse noise



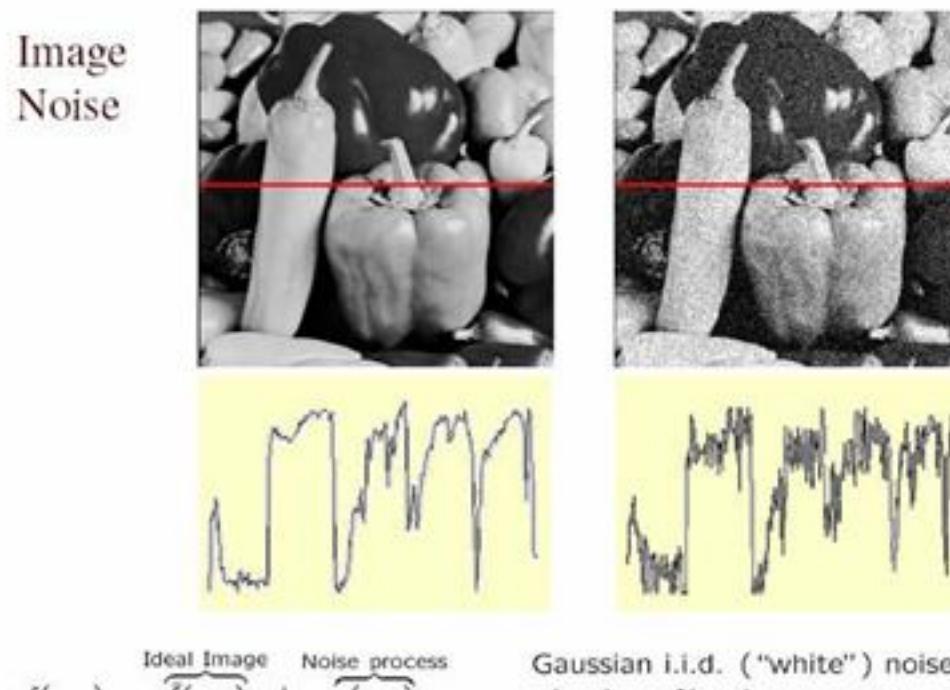
Gaussian noise

- **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



Source: M. Hebert



Усреднение нескольких кадров



Зашумленные изображения



Усреднение по 10 изображениям

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

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

$$E(\bar{I}(i, j)) = g_r(i, j);$$

Так работали камеры в некоторых сотовых телефонах



Усреднение

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

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

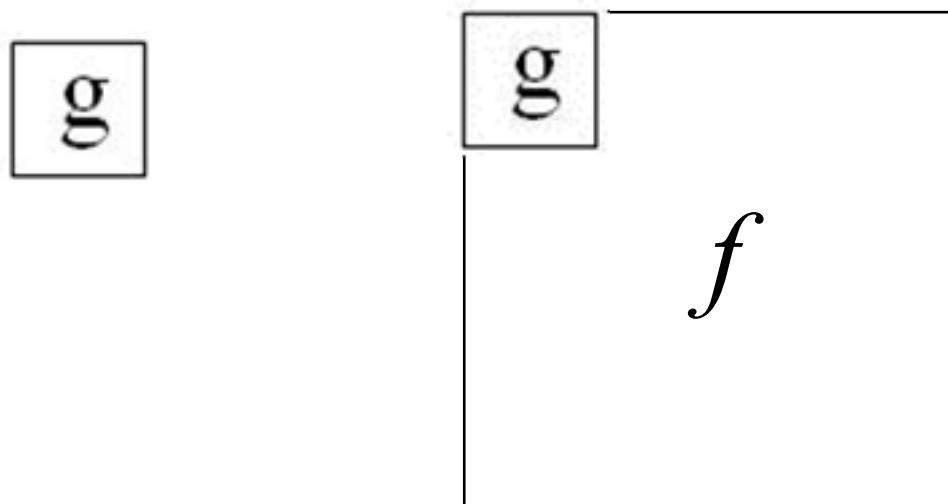
“box filter”



Определение свертки

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

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





Основные свойства

- **Линейность:** $\text{filter}(f_1 + f_2) = \text{filter}(f_1) + \text{filter}(f_2)$
- **Инвариантность к сдвигу:** не зависит от сдвига пикселя: $\text{filter}(\text{shift}(f)) = \text{shift}(\text{filter}(f))$
- Теория: любой линейный оператор, инвариантный к сдвигу, может быть записан в виде свертки
- Чтобы доказать нелинейность фильтра, можно воспользоваться основными свойствами, и показать их не выполнение на примере



Свойства

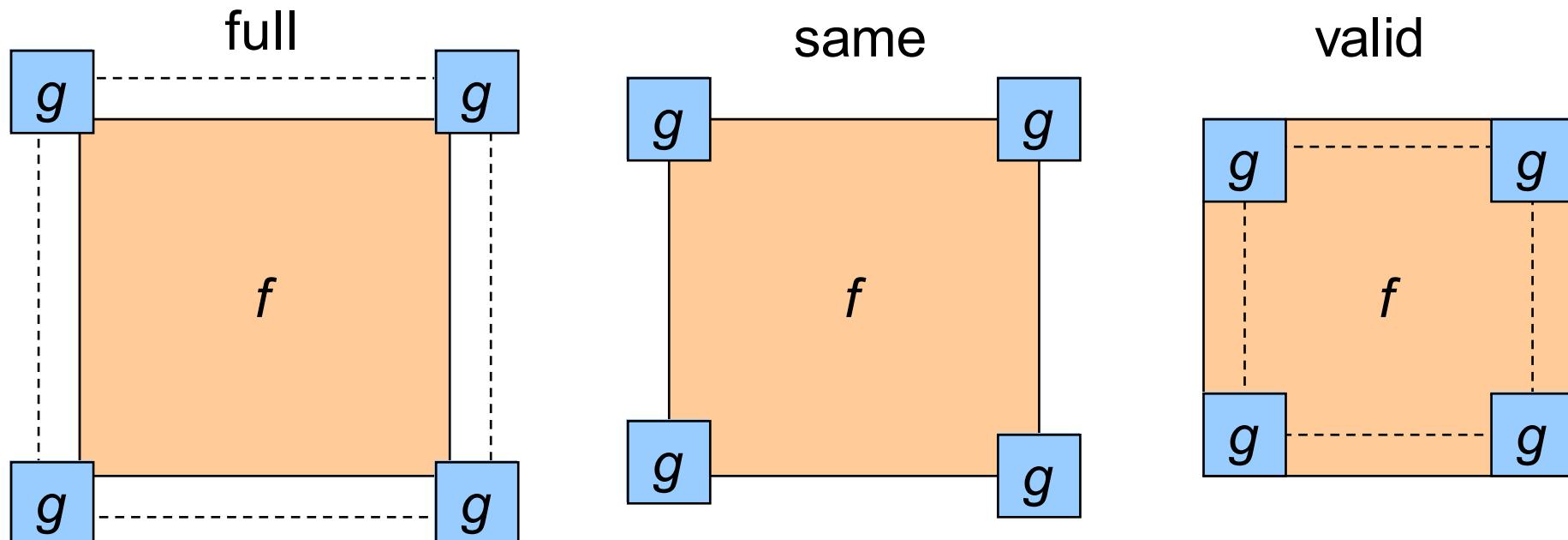
- Коммутативность: $a * b = b * a$
 - Нет никакой разницы между изображением и ядром фильтра
- Ассоциативность: $a * (b * c) = (a * b) * c$
 - Последовательное применение фильтров: $((a * b_1) * b_2) * b_3$
 - Эквивалентно применению такого фильтра: $a * (b_1 * b_2 * b_3)$
- Дистрибутивность по сложению:
$$a * (b + c) = (a * b) + (a * c)$$
- Умножение 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?

- Больше на размер фильтра -1
- Тот же, что у исходного
- Меньше на полуразмер фильтра

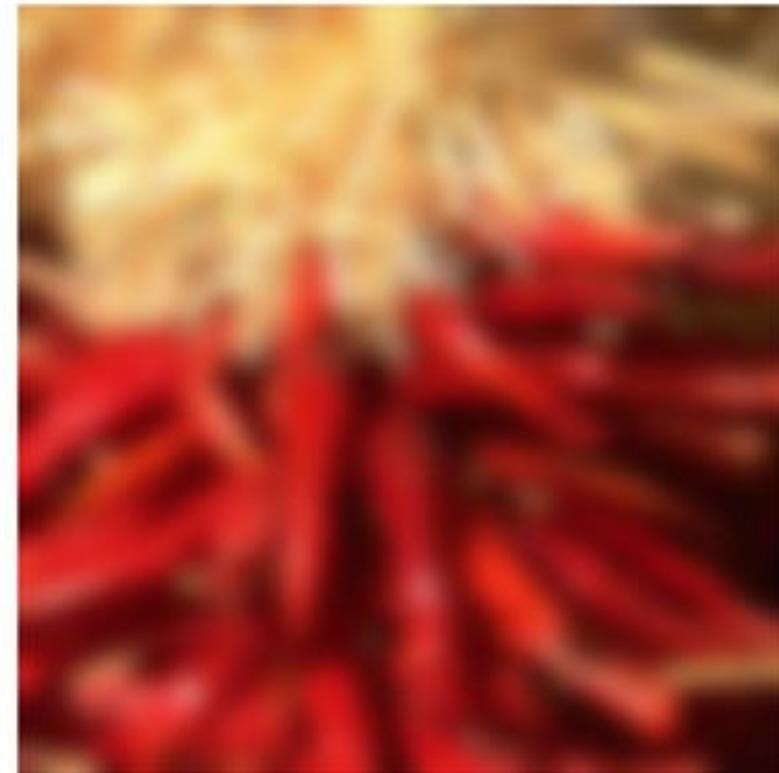




Детали реализации

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



Source: S. Marschner



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

?

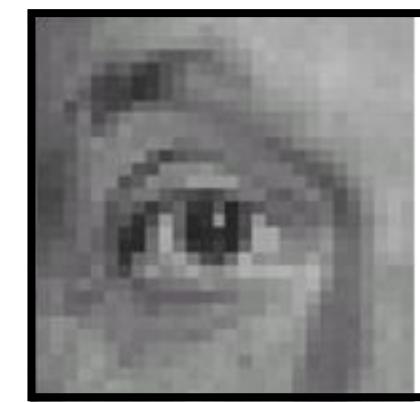


Простейшие фильтры



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

?

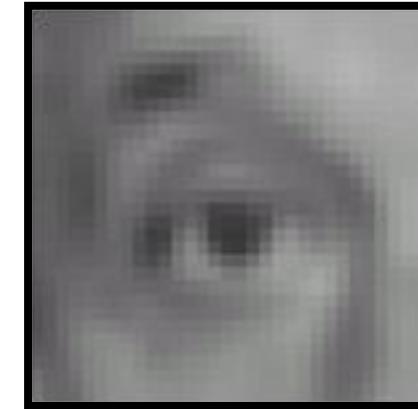


Простейшие фильтры



Original

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

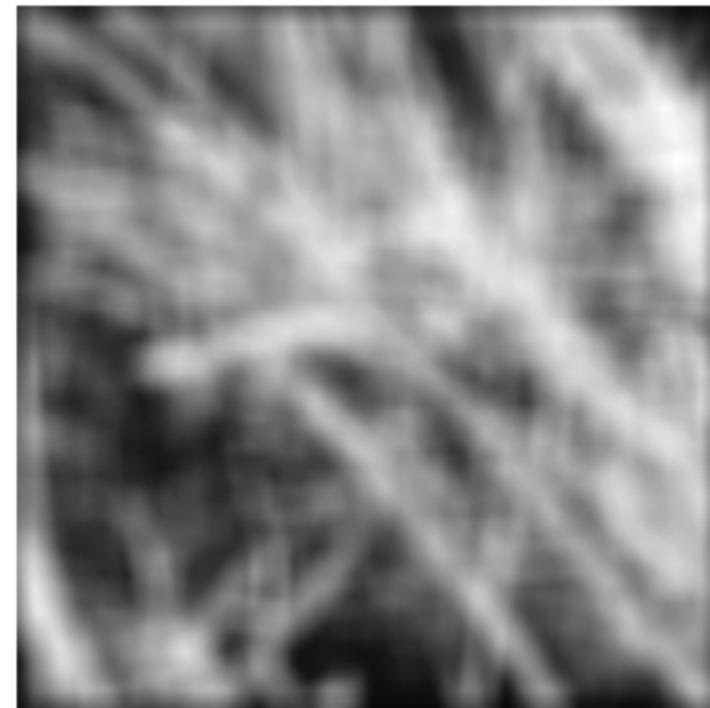


Blur (with a
box filter)



Сглаживание с box-фильтром

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



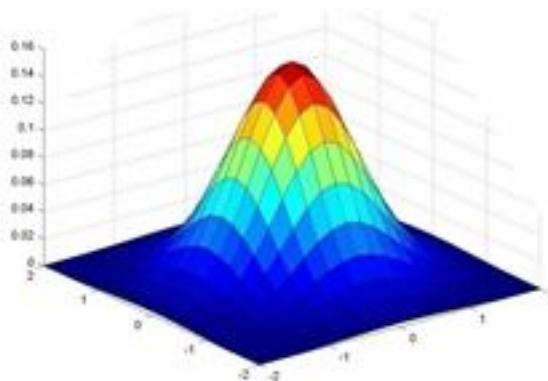
Source: D. Forsyth



Ядро фильтра гаусса

Взвешиваем вклад пикселей по окрестности с учетом близости к центру:

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



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

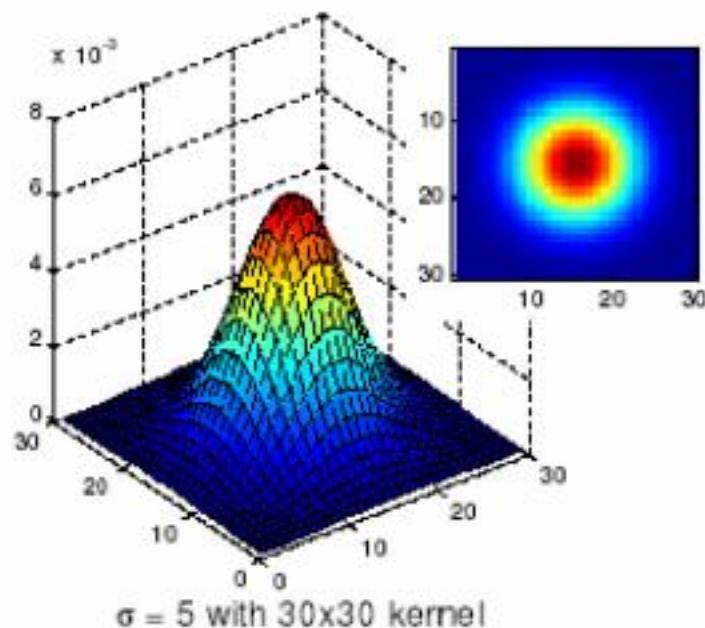
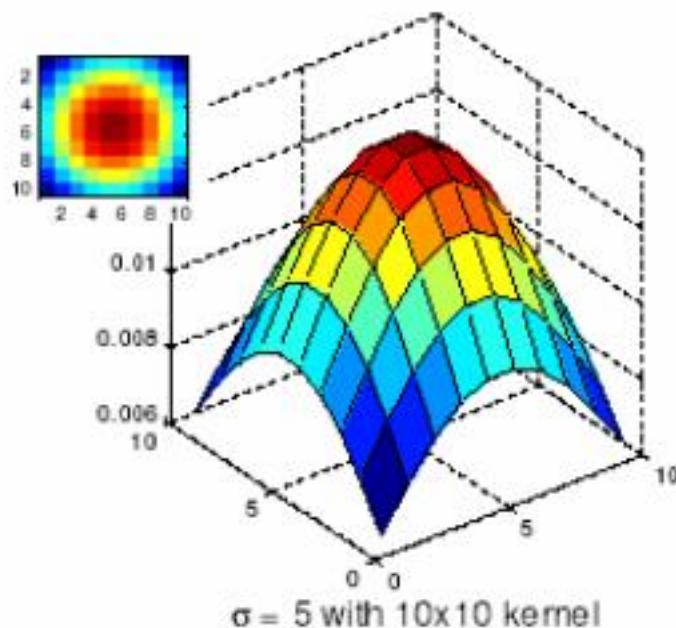
$5 \times 5, \sigma = 1$

Source: C. Rasmussen



Выбор размера ядра

- Размер ядра дискретного фильтра ограничен

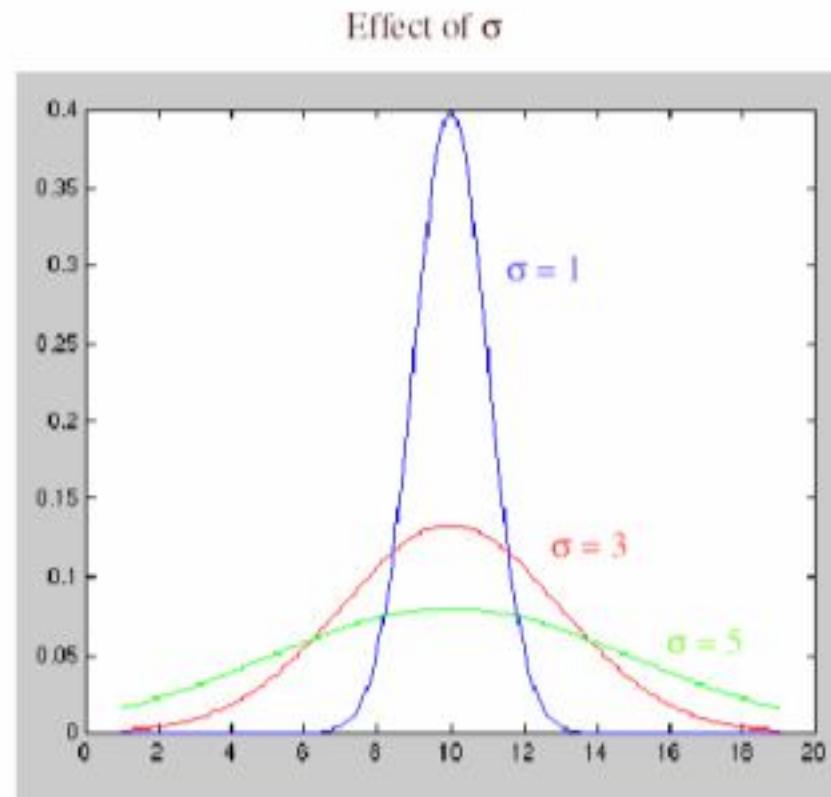


Source: K. Grauman



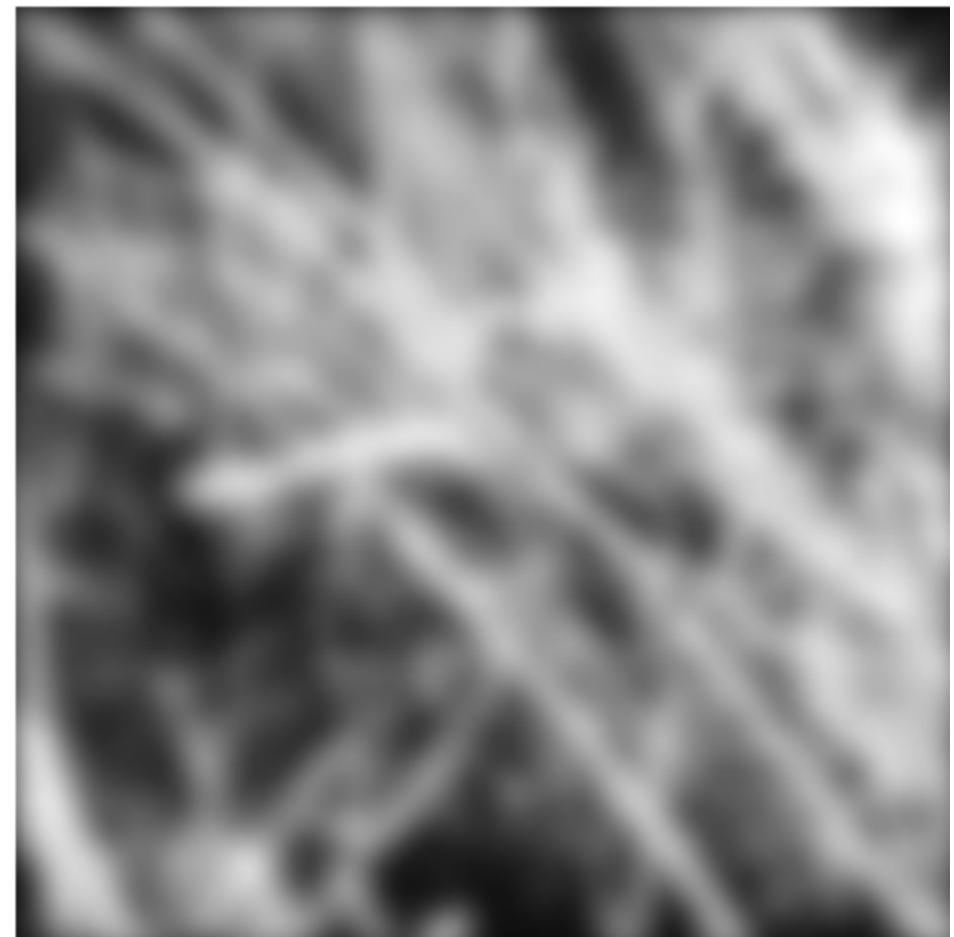
Выбор размера ядра

- Эмпирика: полуразмер фильтра равен 3σ



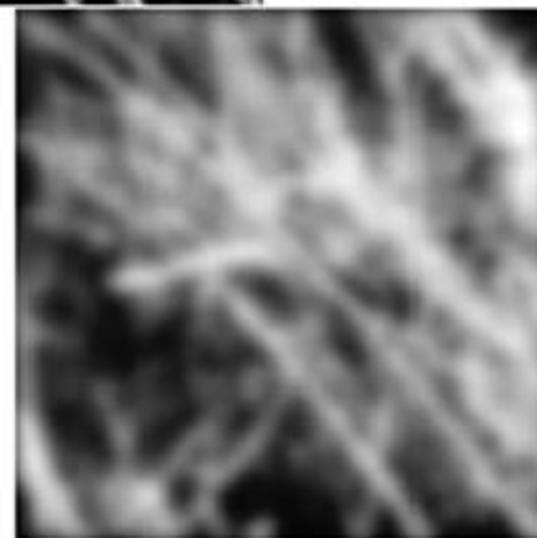
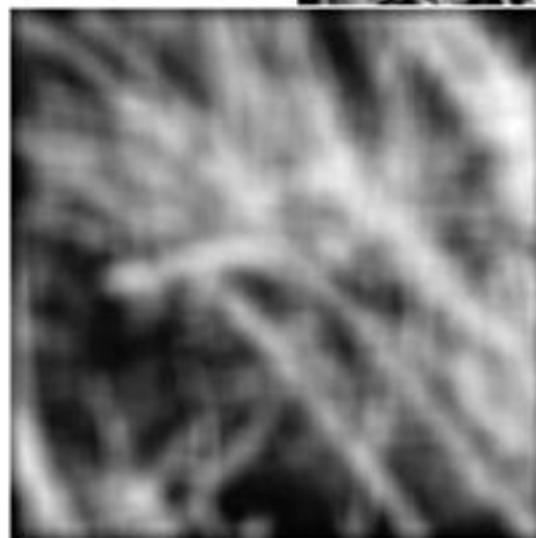


Сглаживание фильтром гаусса





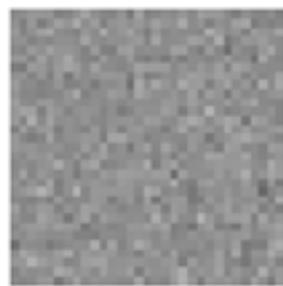
Сравнение



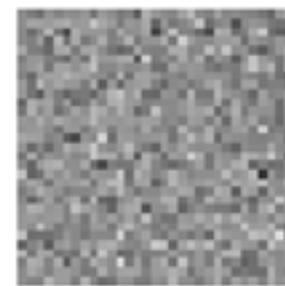


Подавление гауссова шума

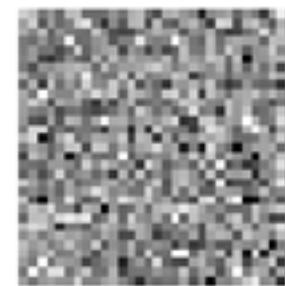
$\sigma=0.05$



$\sigma=0.1$



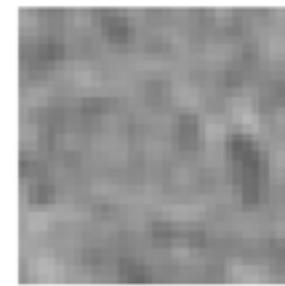
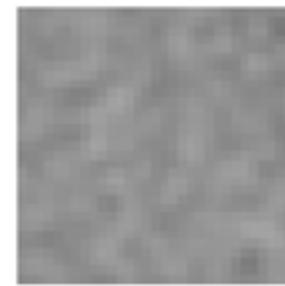
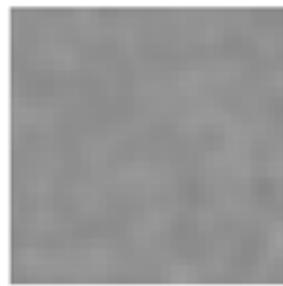
$\sigma=0.2$



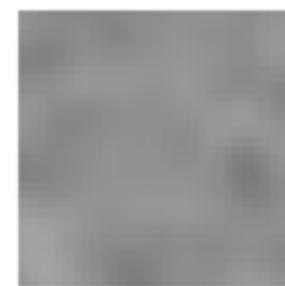
no
smoothing



$\sigma=1$ pixel



$\sigma=2$ pixels



Сглаживание фильтрами большого радиуса подавляет шум, но размывает изображение



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 $\sigma\sqrt{2}$



Сепарабельность

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

Сепарабельное ядро

Раскладывается в произведение двух одномерных фильтром 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:

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

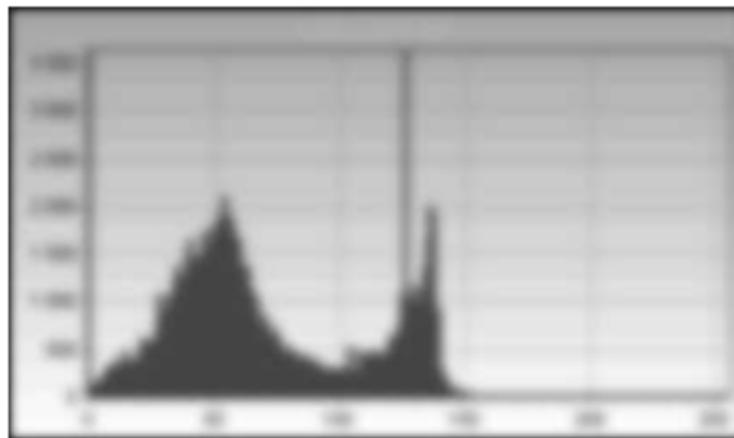
Затем свертка по столбцу:

Почему сепарабельность полезна на практике?

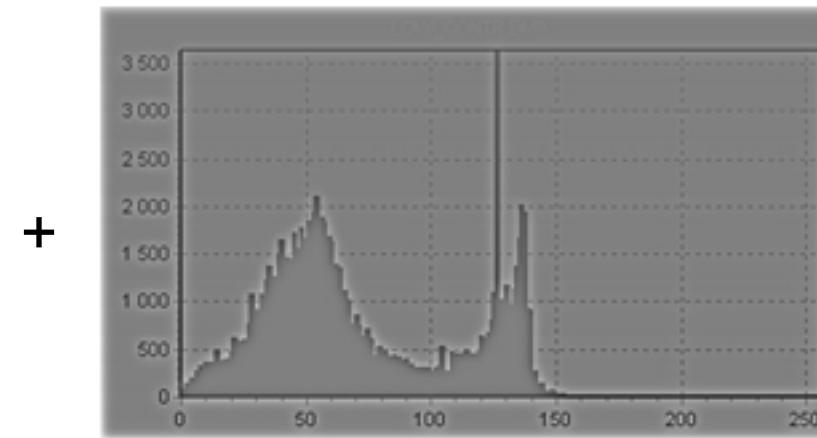
Source: K. Grauman



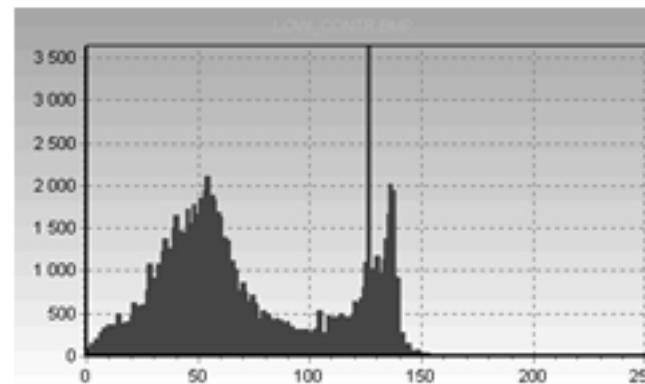
Маленькая экскурсия к Фурье



Низкие частоты



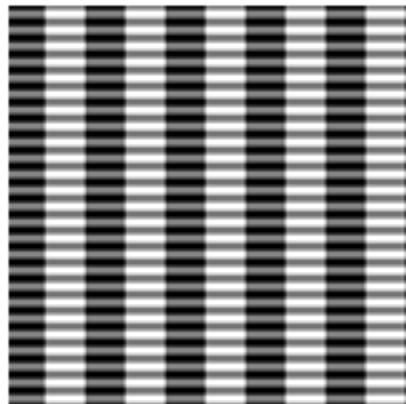
Высокие частоты



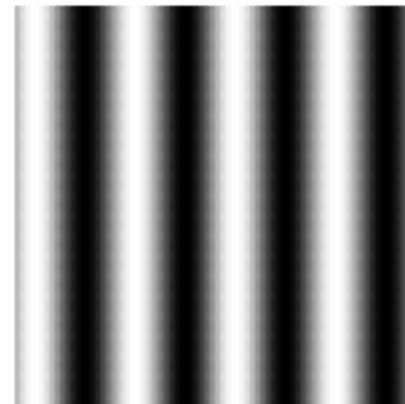


Фильтр Гаусса

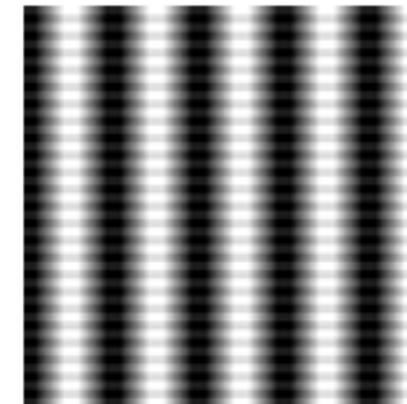
Результат свертки фильтром гаусса и усреднения



Исходное изображение



Фильтр Гаусса с
Sigma = 4



Усреднение по 49
пикселям (7x7)

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



Подавление шума «соль и перец»

3x3



5x5



7x7

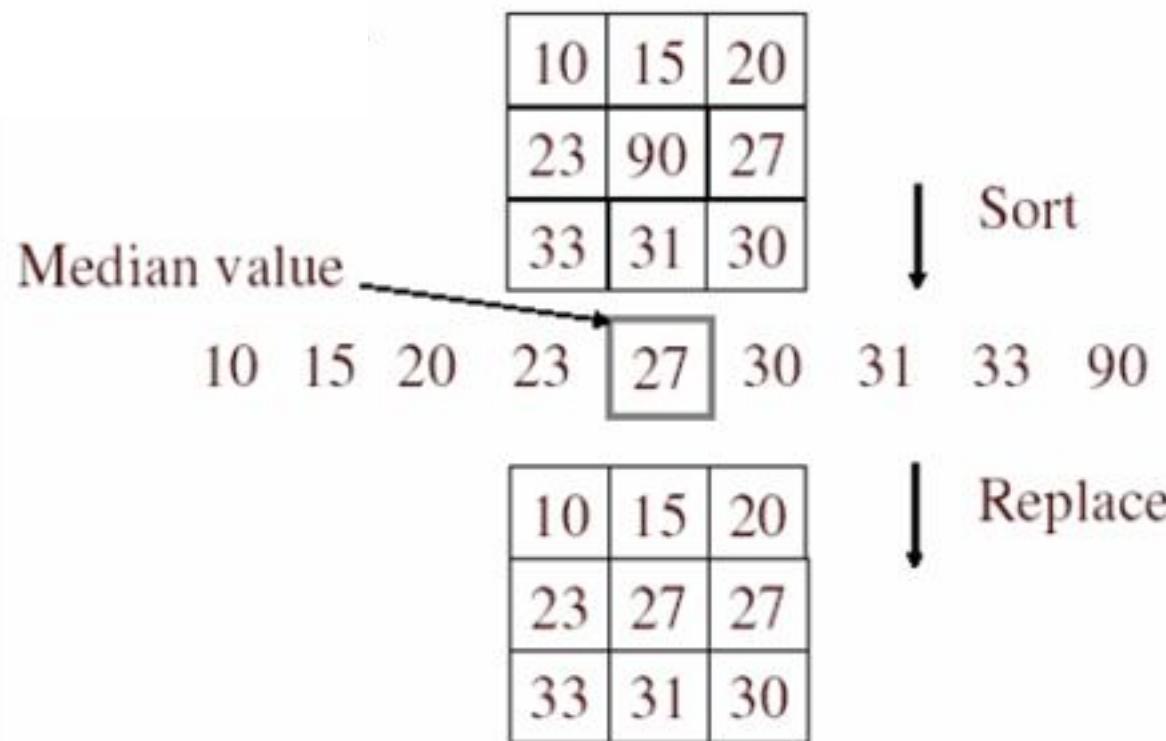


Применим фильтр Гаусса
Чем результат плох?



Медианный фильтр

- Выбор медианы из выборки пикселей по окрестности данного



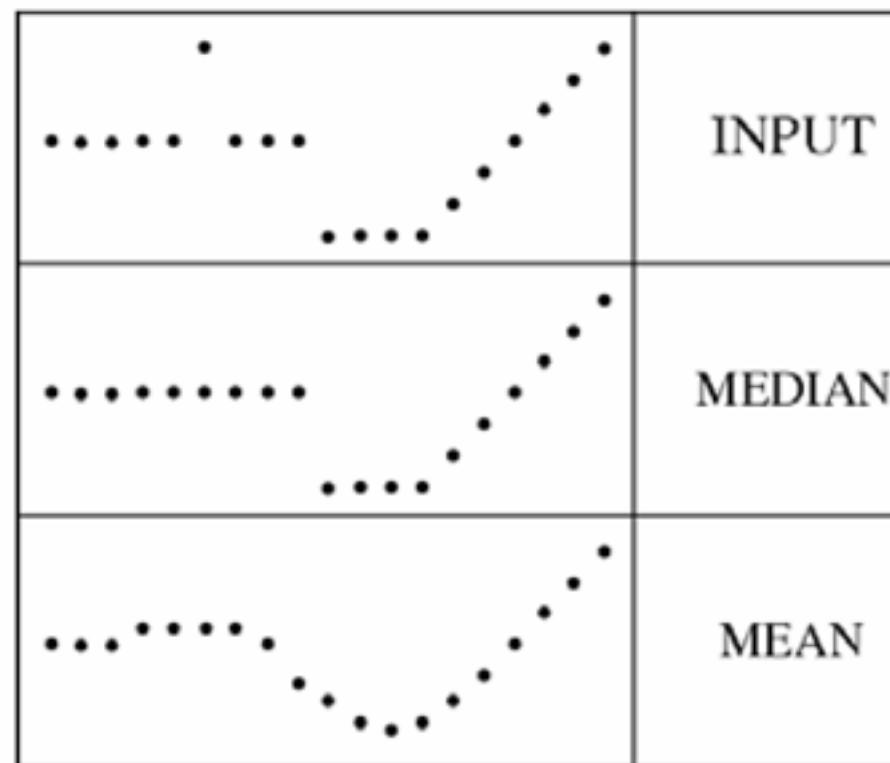
- Является ли фильтр линейным?



Медианный фильтр

- В чем преимущество медианного фильтра перед фильтром гаусса?
 - Устойчивость к выбросам (outliers)
 - Не появляется новых (промежуточных) значений

filters have width 5 :

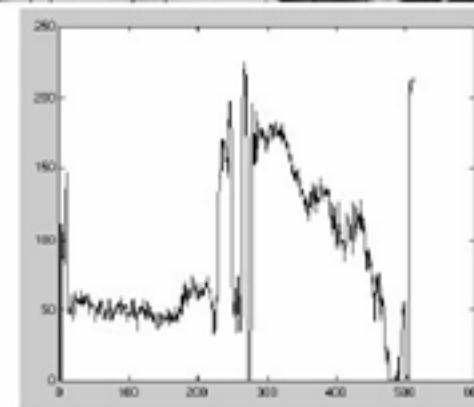
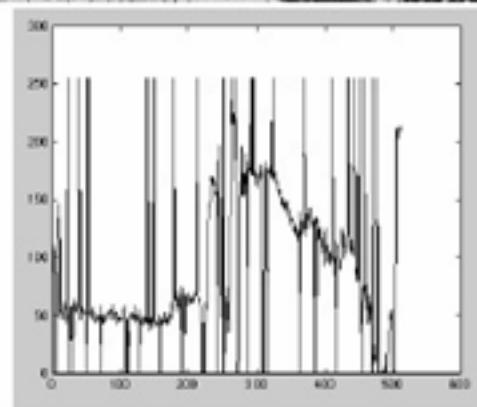


Source: K. Grauman



Медианный фильтр

Шум «соль и перец» Медианная фильтрация

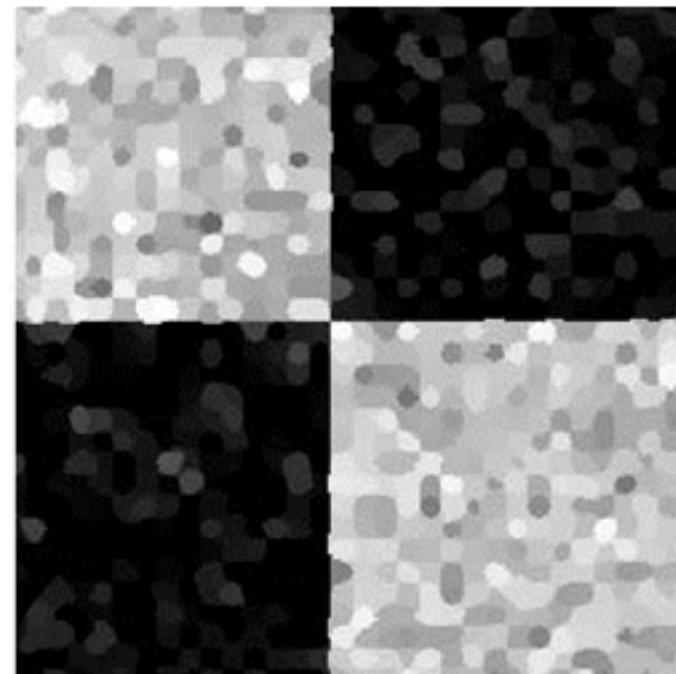
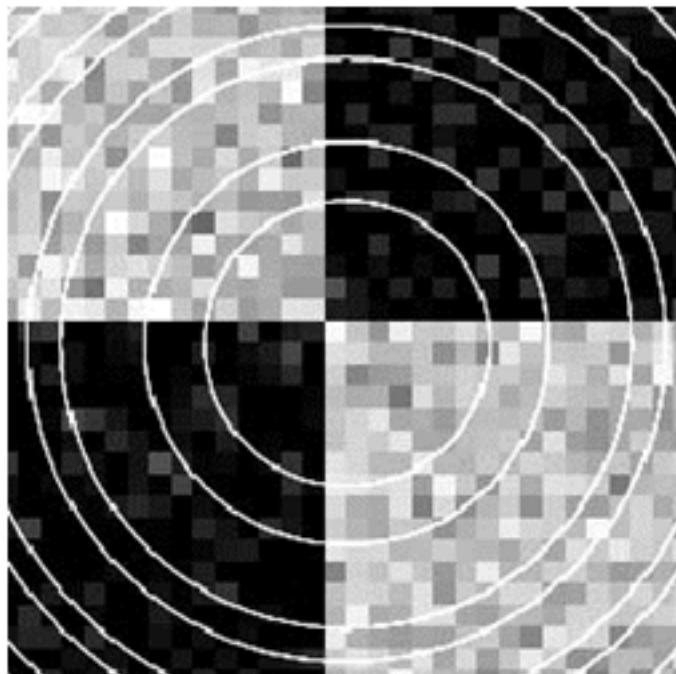


Source: M. Hebert



Медианный фильтр

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





Сравнение фильтров

3x3



Гауссов

5x5



7x7

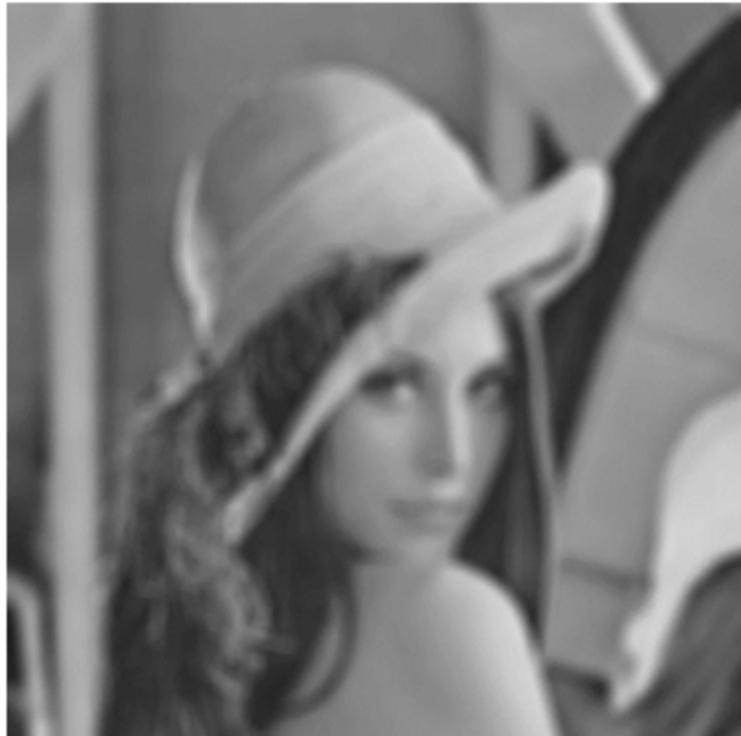


Медианный





Повышение резкости



Иногда с помощью
фильтрации можно немного
повысить резкость
изображения (подчеркнуть
края)



Повышение резкости

Что теряется при сглаживании?



-



=



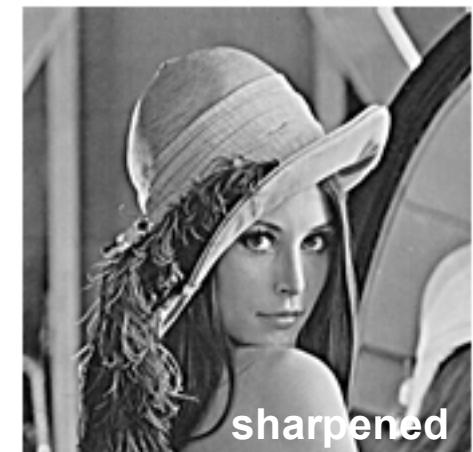
Добавим дополнительно высокие частоты:



+ α

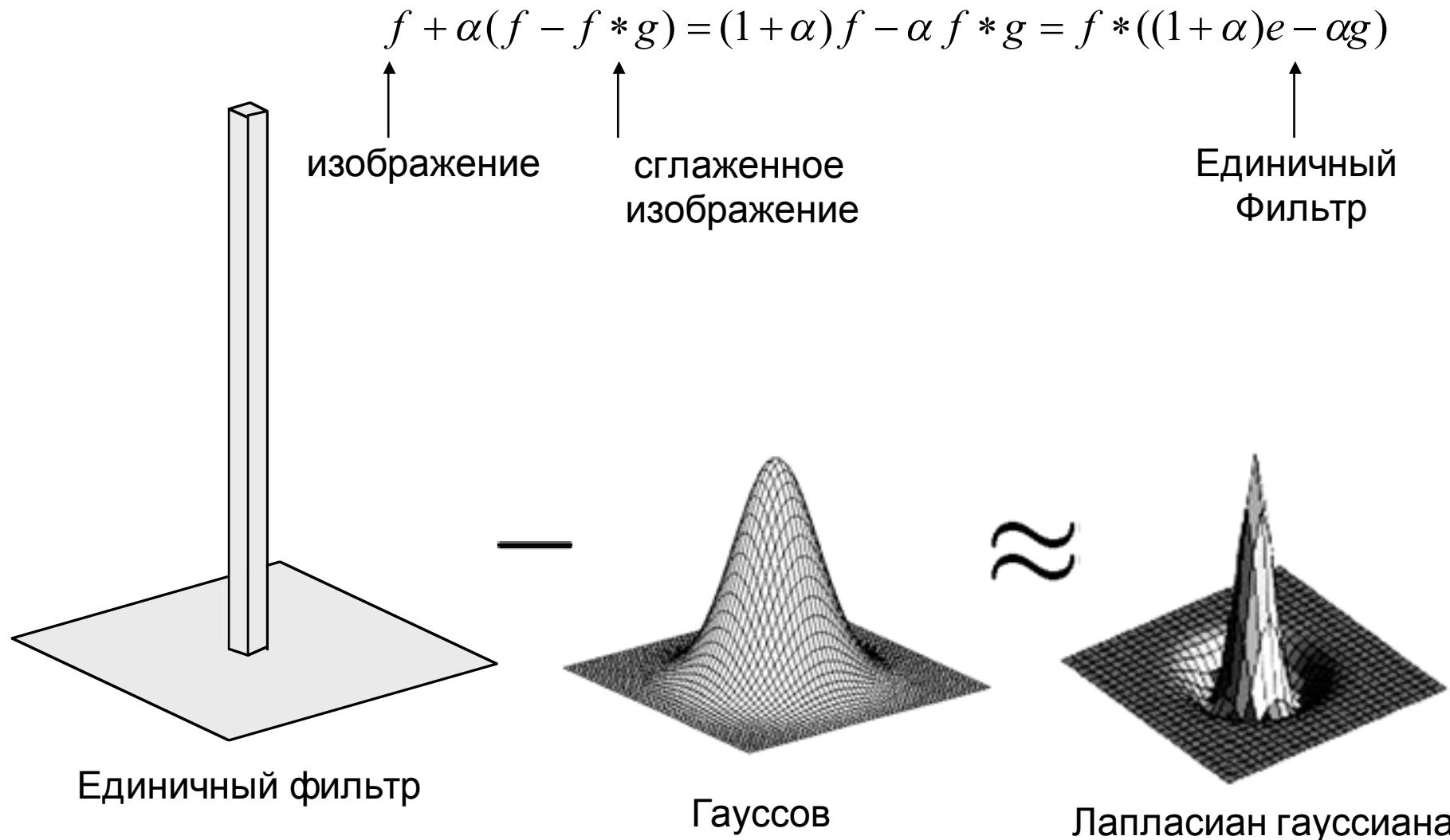


=





Фильтр Unsharp





Пример повышения резкости

Ядро
свертки

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



Центральный элемент – положительный, на границах – отрицательный. Сумма элементов равна 1.



Сравнение изображений

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



исходное
изображение



искаженное
изображение



Метрики на изображениях

- Среднеквадратичная ошибка (MSE)

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

- Пиковое отношение сигнал/шум (PSNR)

$$PSNR_{dB} = 10 \lg \frac{M^2}{MSE} \quad M - \text{максимальное значение пикселя}$$



Метрики на изображениях

- PSNR и MSE не учитывают особенности человеческого восприятия!



Оригинал



Метрики на изображениях

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



Повышена контрастность Добавлен белый гауссов шум

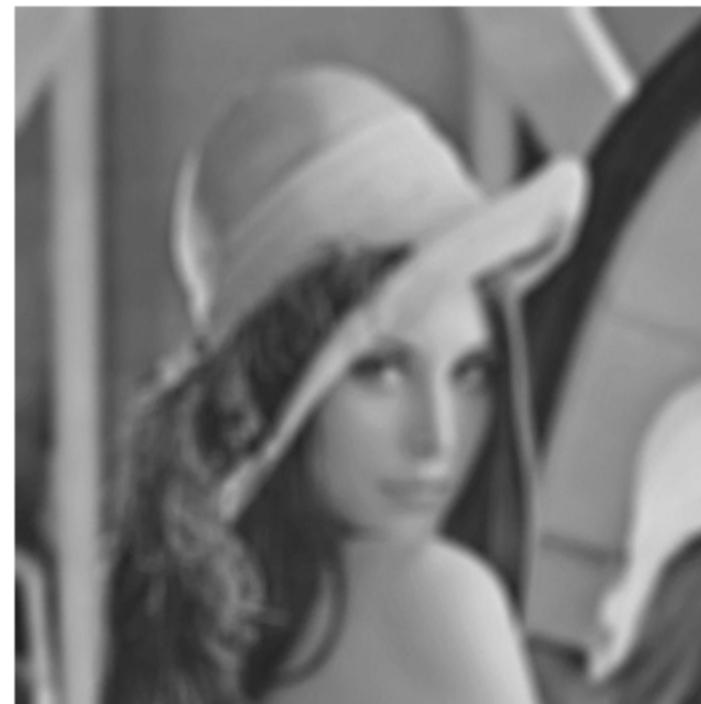


Метрики на изображениях

- И у этих – тоже примерно 25 dB!



Добавлен импульсный шум



Размытие



Метрики на изображениях

- И у этого – тоже!



Артефакт блочности после JPEG



Метрики на изображениях

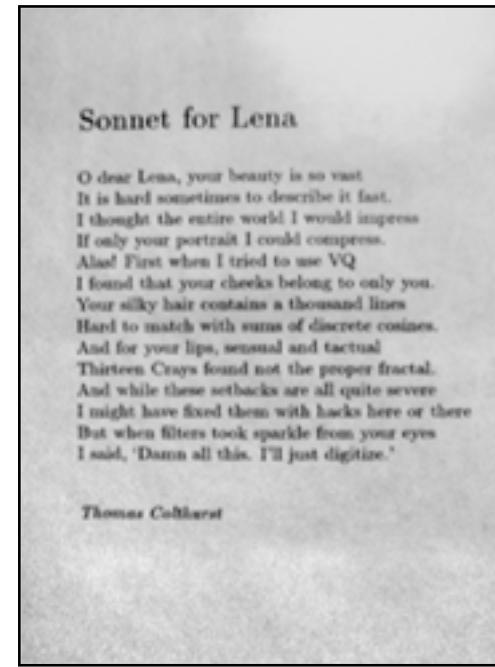
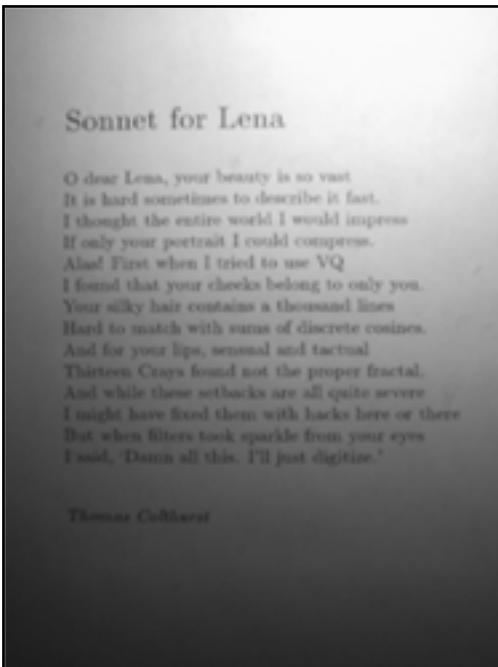
- Вывод: PSNR не всегда отражает реальный видимый уровень искажений.
- Как улучшить?

- 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)



Компенсация разности освещения

Пример





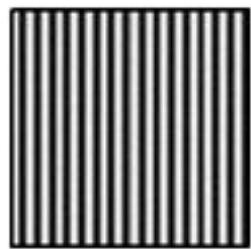
Компенсация разности освещения

Идея:

Формирование изображения:

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

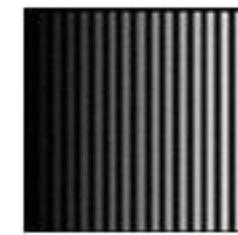
Плавные изменения яркости относятся к освещению,
резкие - к объектам.



объект $r(i, j)$



освещение $l(i, j)$



Изображение
освещенного
объекта $I(i, j)$



Выравнивание освещения

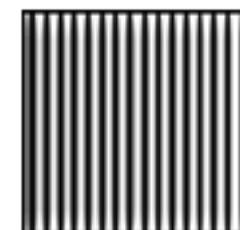
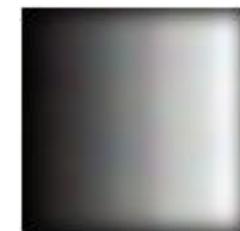
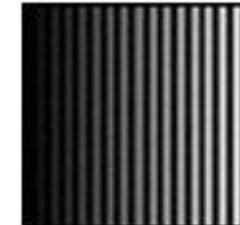
- Алгоритм Single scale retinex (SSR)
 - Получить приближенное изображение освещения путем низочастотной фильтрации

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

- Восстановить изображение по формуле

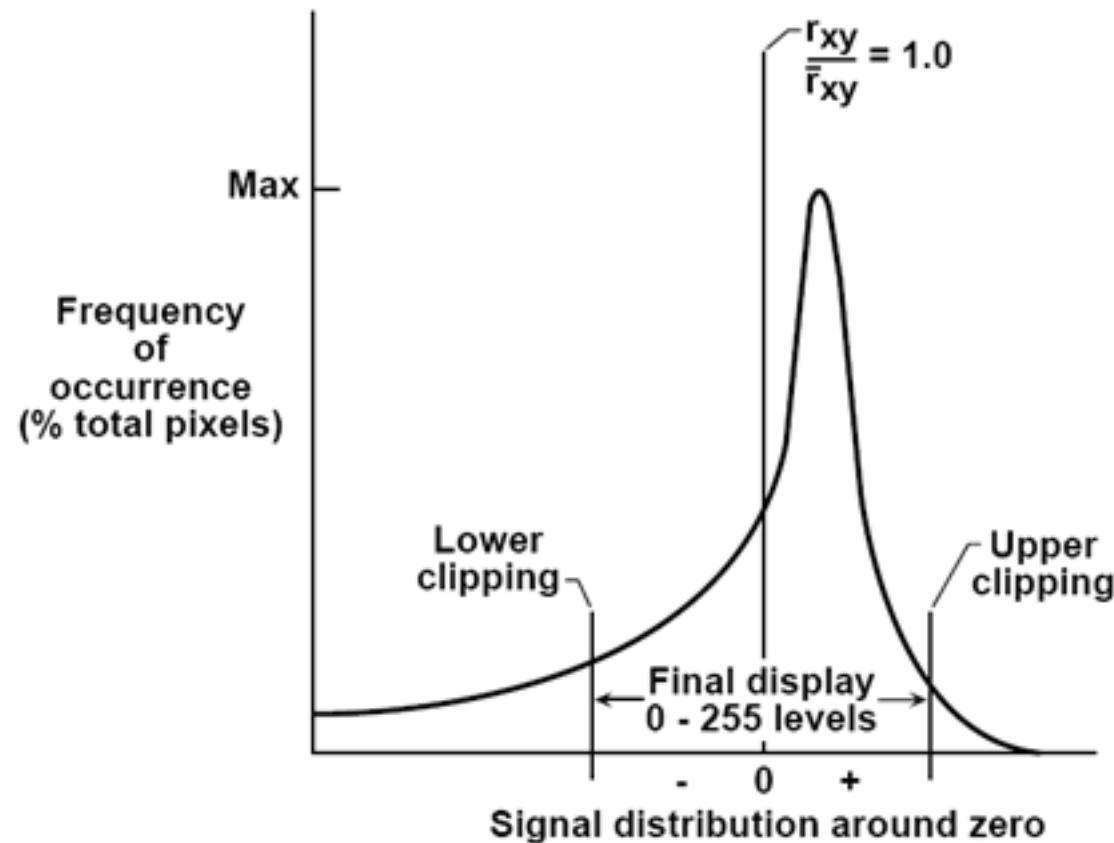
$$\hat{r}(i, j) = \log \frac{I(i, j)}{\hat{l}(i, j)}$$

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





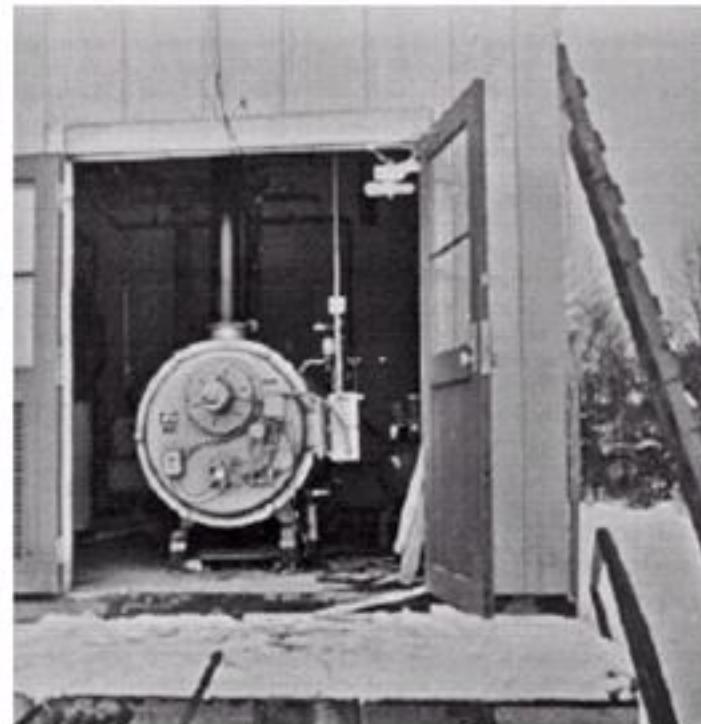
Обрезание по порогу





Выравнивание освещения

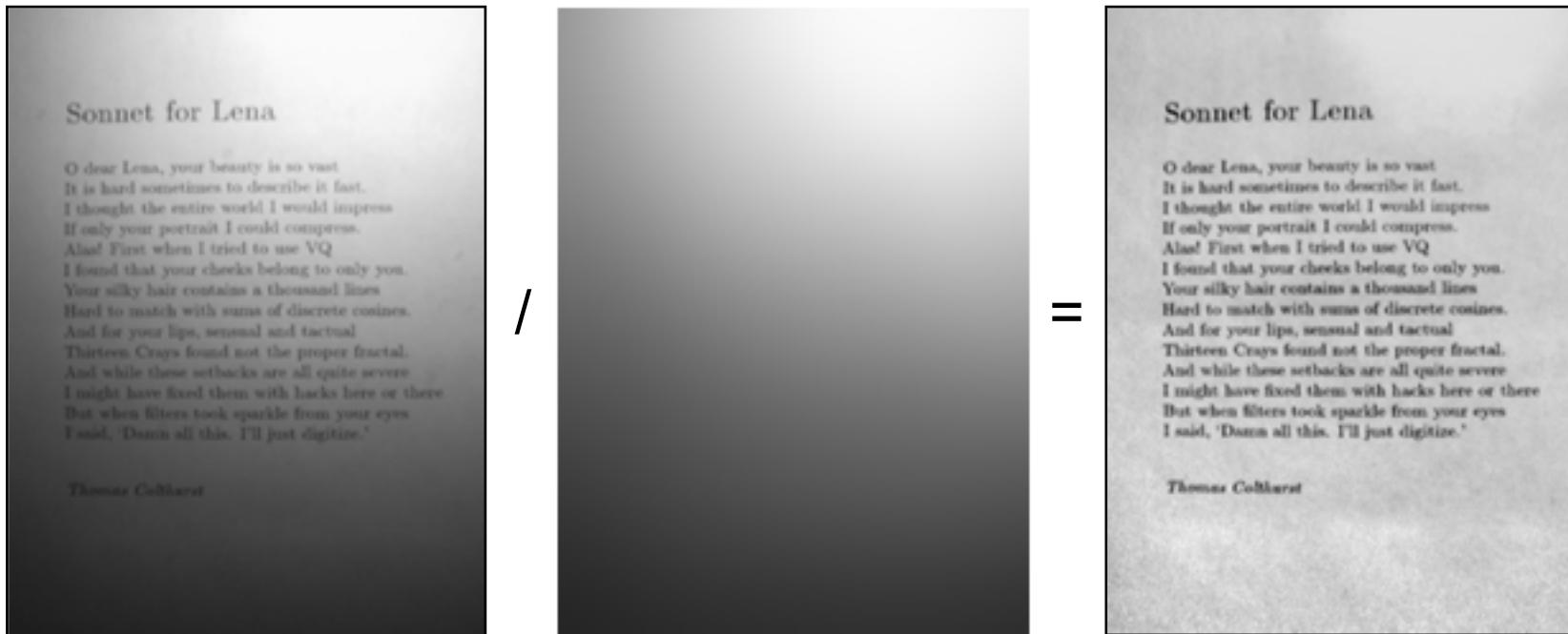
Пример





Компенсация разности освещения

Пример



Gauss 14.7 пикселей



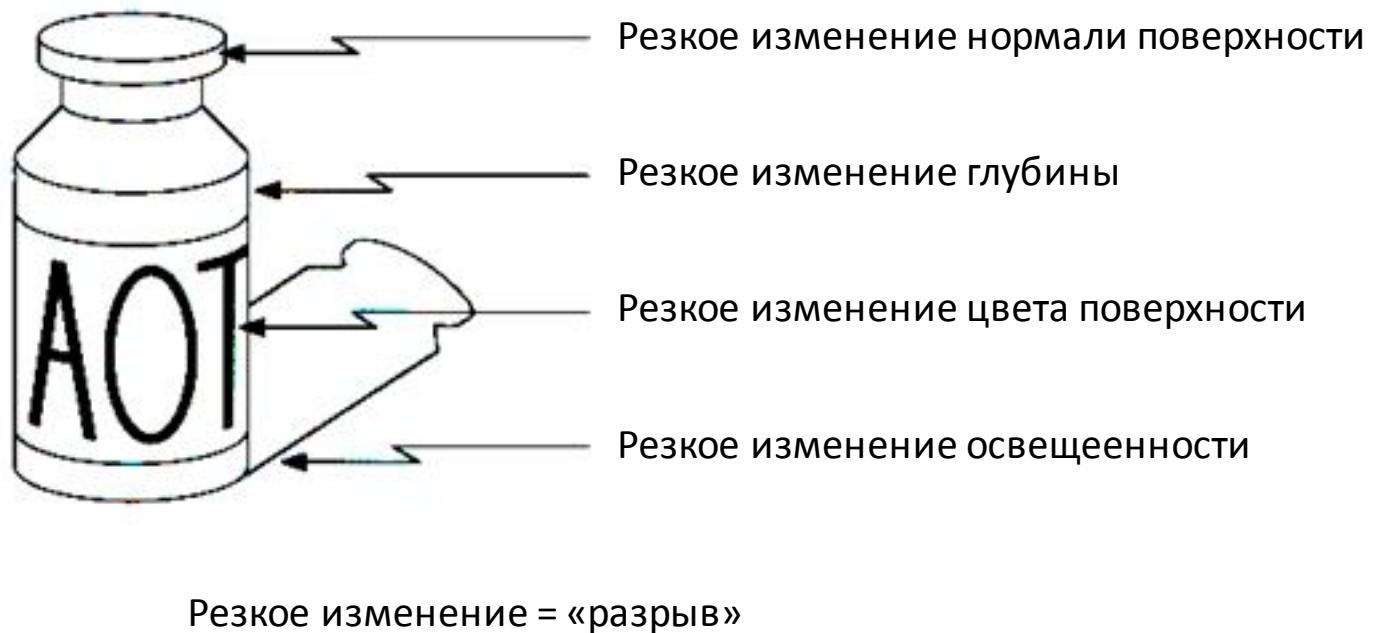
Выделение краев

- **Задача:** Выделить резкие изменения (разрывы) изображения
- Интуитивно понятно, что основная информация в картинке содержится как раз в краях (границах)
 - Компактное представление
 - Соответствует устройству мозга
- **Идеал:** рисунок художника (но артист уже пользуются своими знаниями об объектах)





Откуда берутся края

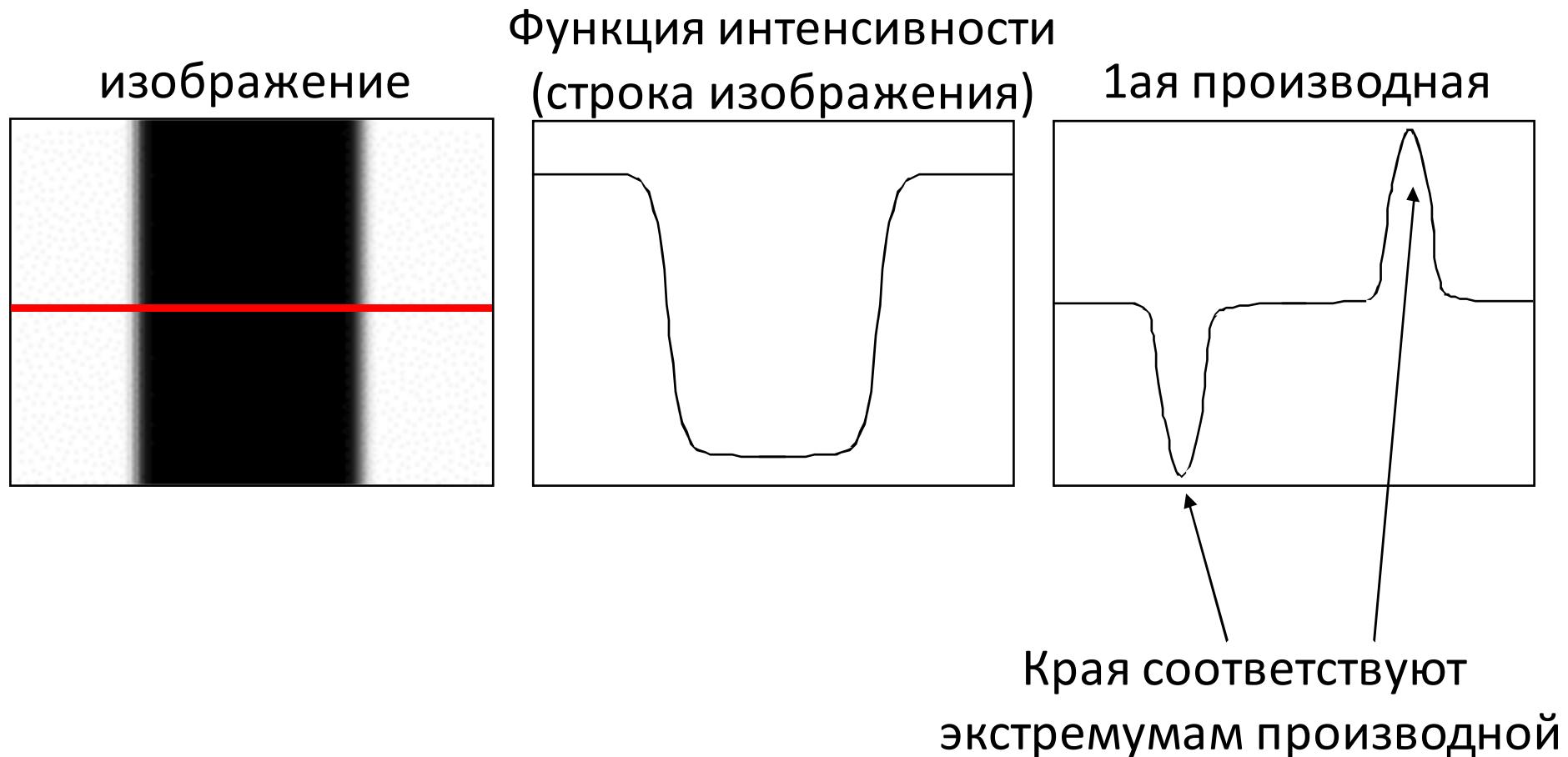


- Существует множество причин формирования краев на изображении



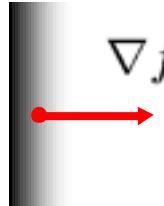
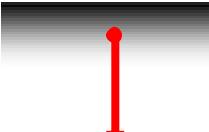
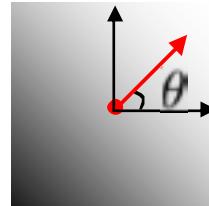
Описание «края»

- Край – это точка резкого изменения значений функции интенсивности изображения





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:

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



Дифференцирование и свёртка

- Для функции 2х переменных, $f(x,y)$:

$$\frac{\partial f}{\partial x} = \lim_{\varepsilon \rightarrow 0} \left(\frac{f(x + \varepsilon, y) - f(x, y)}{\varepsilon} \right)$$

- Линейная и инвариантная к переносу, поэтому м.б.
Результатом свертки

- Разностная производная:

$$\frac{\partial f}{\partial x} \approx \frac{f(x_{n+1}, y) - f(x_n, y)}{\Delta x}$$

- (Очевидно, свертка)

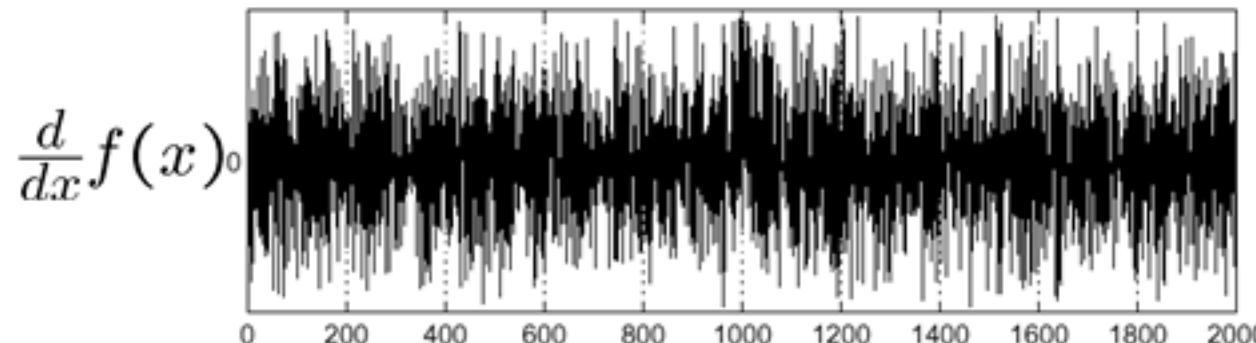
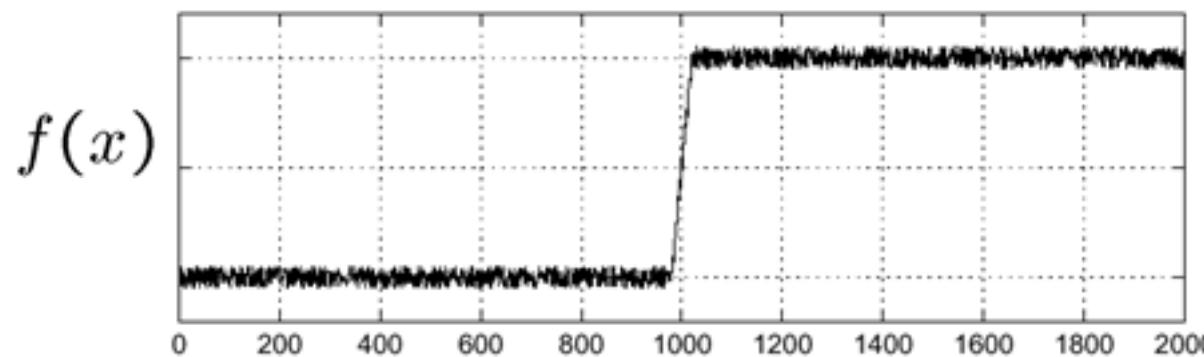
-1	1
----	---

Простейший фильтр



Влияние шума

- Рассмотрим строку или столбец изображения
 - Интенсивность от положения можно рассматривать как сигнал



Край исчез

Source: S. Seitz

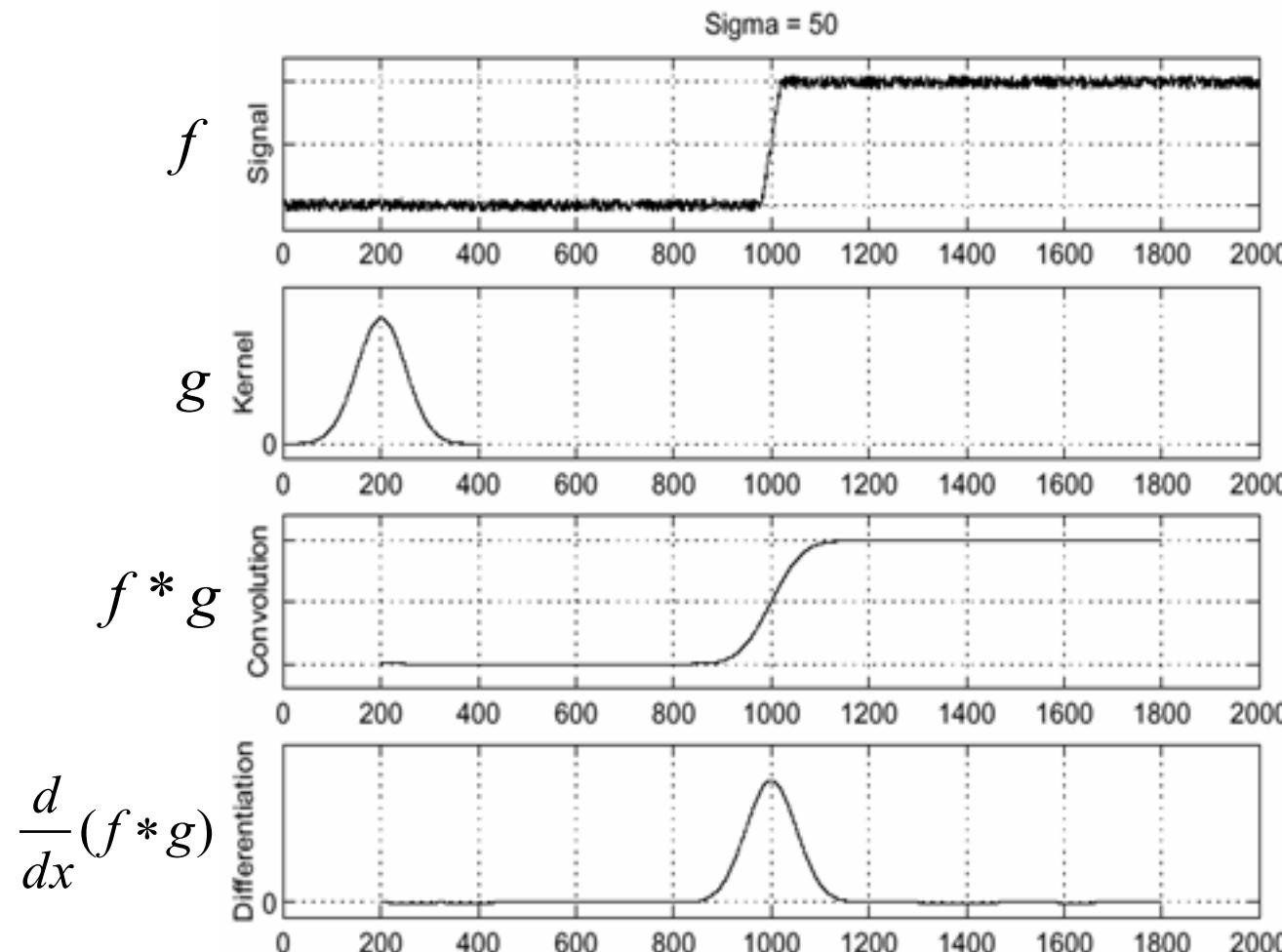


Влияние шума

- Разностные производные очень чувствительны к шуму
 - Зашумленные пиксели отличаются от соседей
 - Чем сильнее шум, тем выше отклик
- Сглаживание
 - Сглаживание делает все пиксели (зашумленные?) чуть более похожими на соседей



Предобработка (сглаживание)



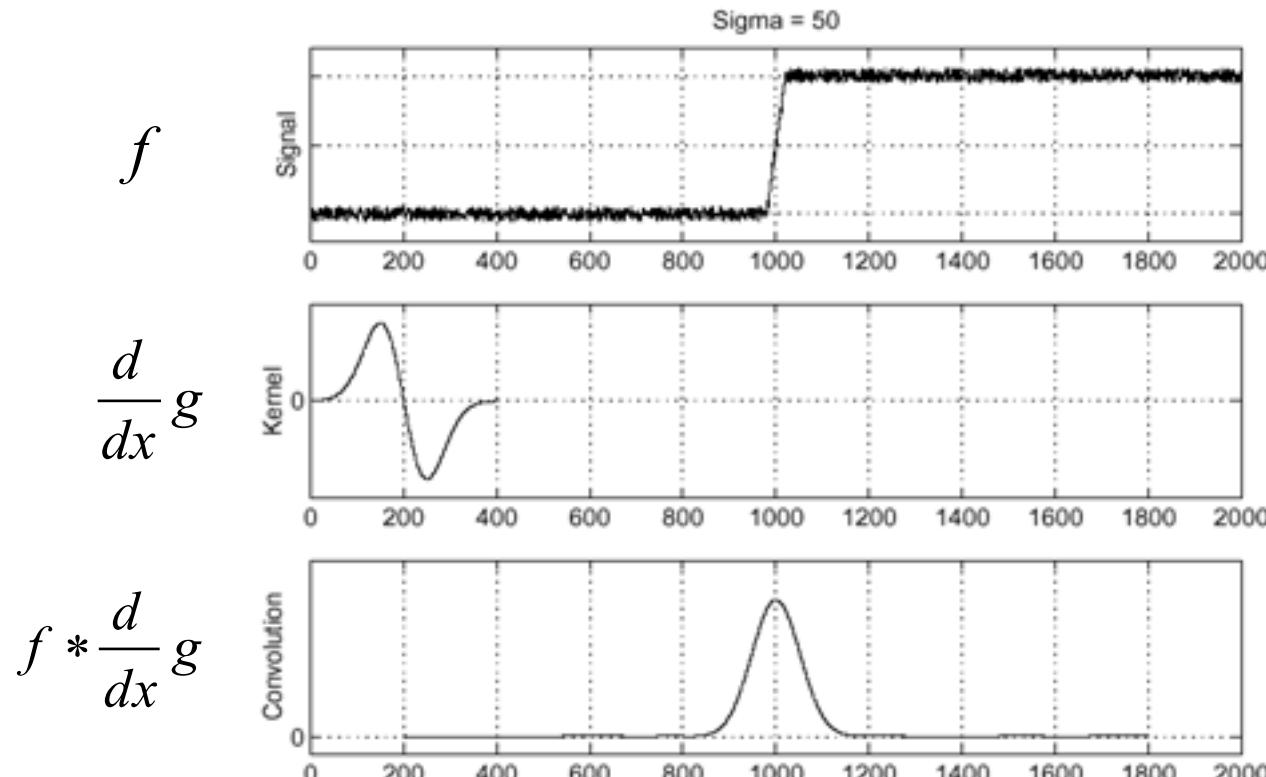
- Для поиска краев ищем пики в: $\frac{d}{dx}(f * g)$

Source: S. Seitz



Свойства свертки

- Операции свертки и дифференцирования ассоциативны:
 - Это экономит 1 операцию:
- $$\frac{d}{dx}(f * g) = f * \frac{d}{dx}g$$



Source: S. Seitz



Известные фильтры

Несколько фильтров, по разному оценивающие производные по направлению:

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Робертса

Превитт

Собеля

Превитт и Собель чуть-чуть сглаживают шум



Карта силы краев

Примеры:



Робертса



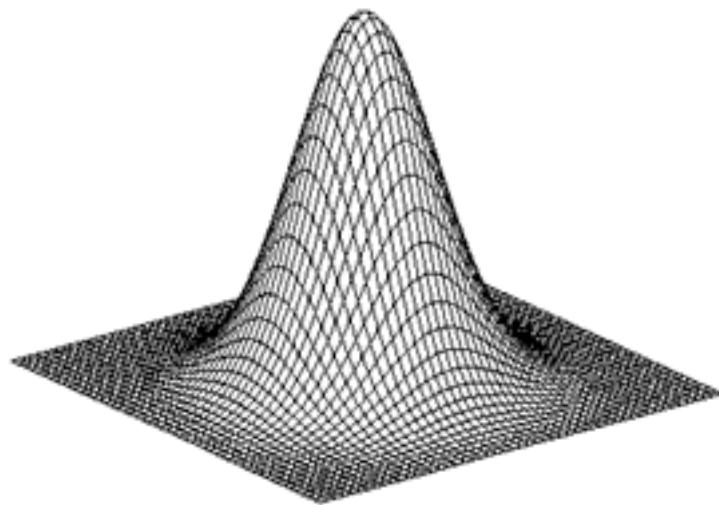
Превитт



Собеля



Производная фильтра Гаусса



$$* [1 \ -1] =$$

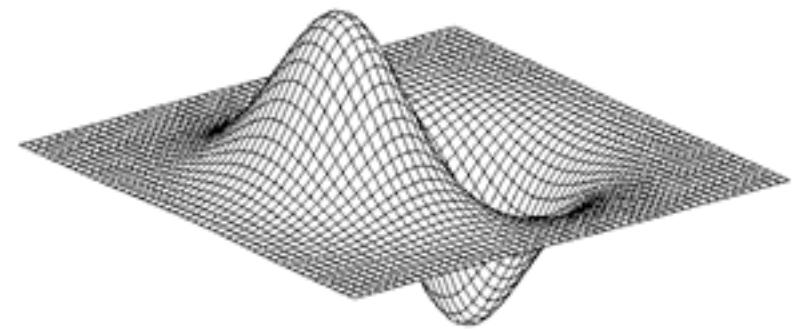
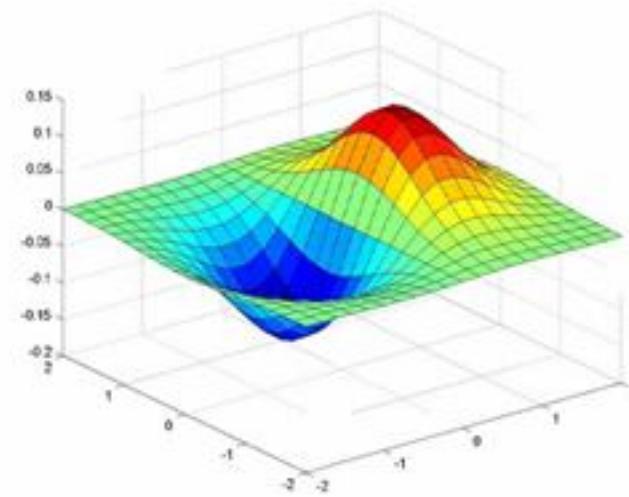


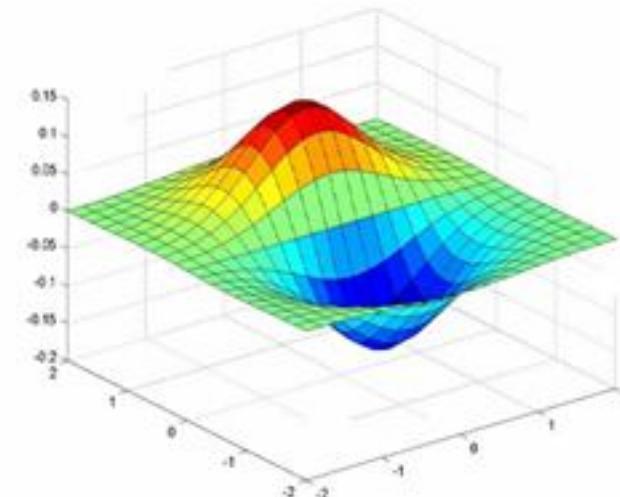
Иллюстрация фильтра



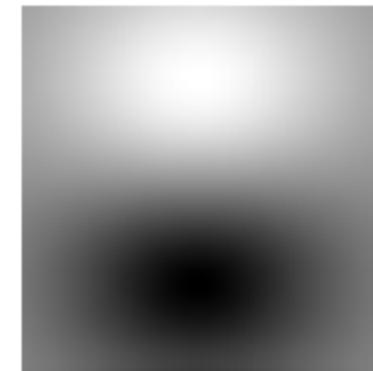
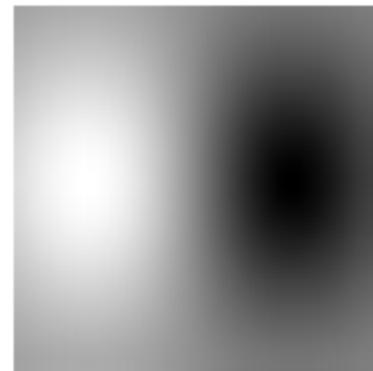
Производная фильтра Гаусса



x-direction



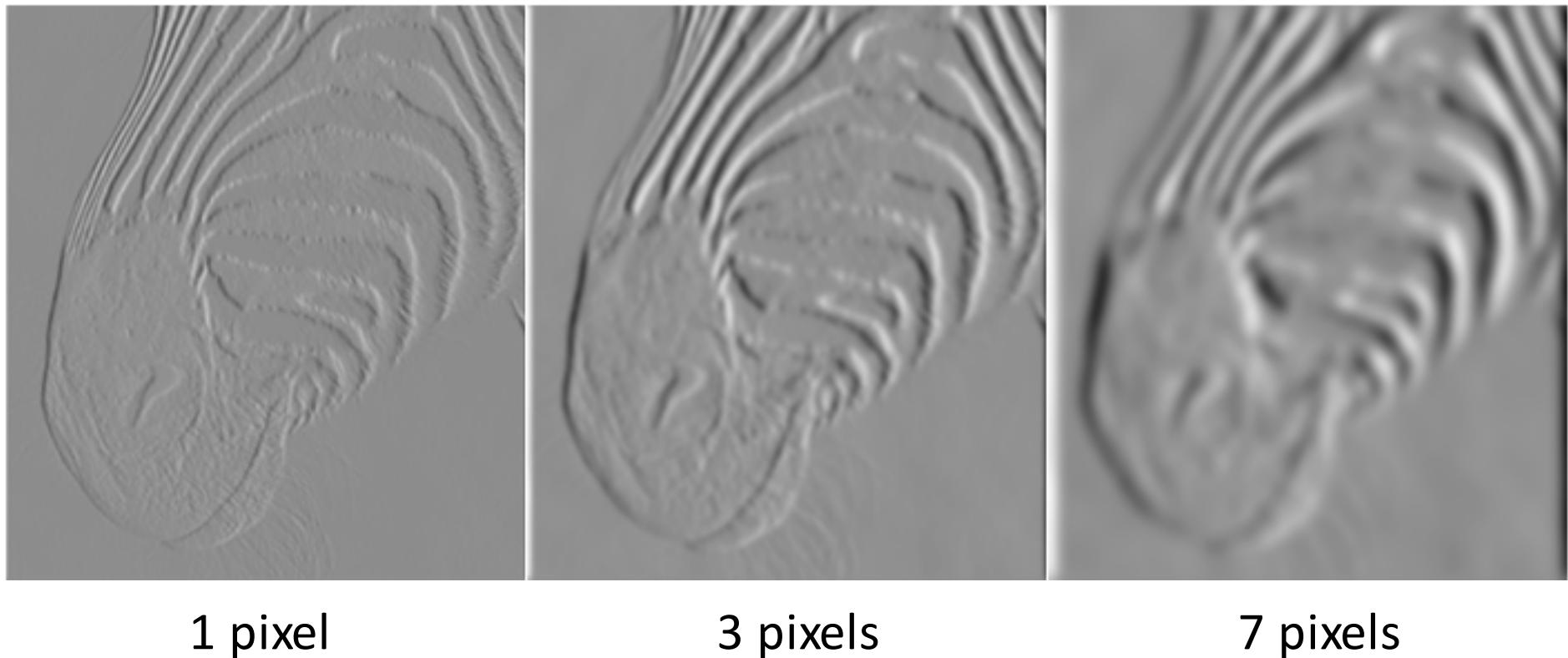
y-direction





Сглаживание и локализация

Применим сглаженные производные разного размера:



Сглаженные производные подавляют шум, но размыают края. Плюс края находится на разных «масштабах»

Source: D. Forsyth



Выделение краев

- Вычисление градиента – это еще не всё...



Исходное изображение



Карта силы краев

- Чего не хватает?
 - Точности – края «толстые» и размытые
 - Информации о связности



Детектор Canny

1. Свертка изображения с ядром – производной от фильтра гаусса
2. Поиск значения и направления градиента
3. Выделение локальных максимумов (Non-maximum suppression)
 - Уточнение полос в несколько пикселей до одного пикселя
4. Связывание краев и обрезание по порогу (гистерезис)
 - Определяем два порога: нижний и верхний
 - Верхний порог используем для инициализации кривых
 - Нижний порог используем для продолжения кривых



Рассмотрим на примере



- Исходное изображение (Lena)



Пример



Норма градиента



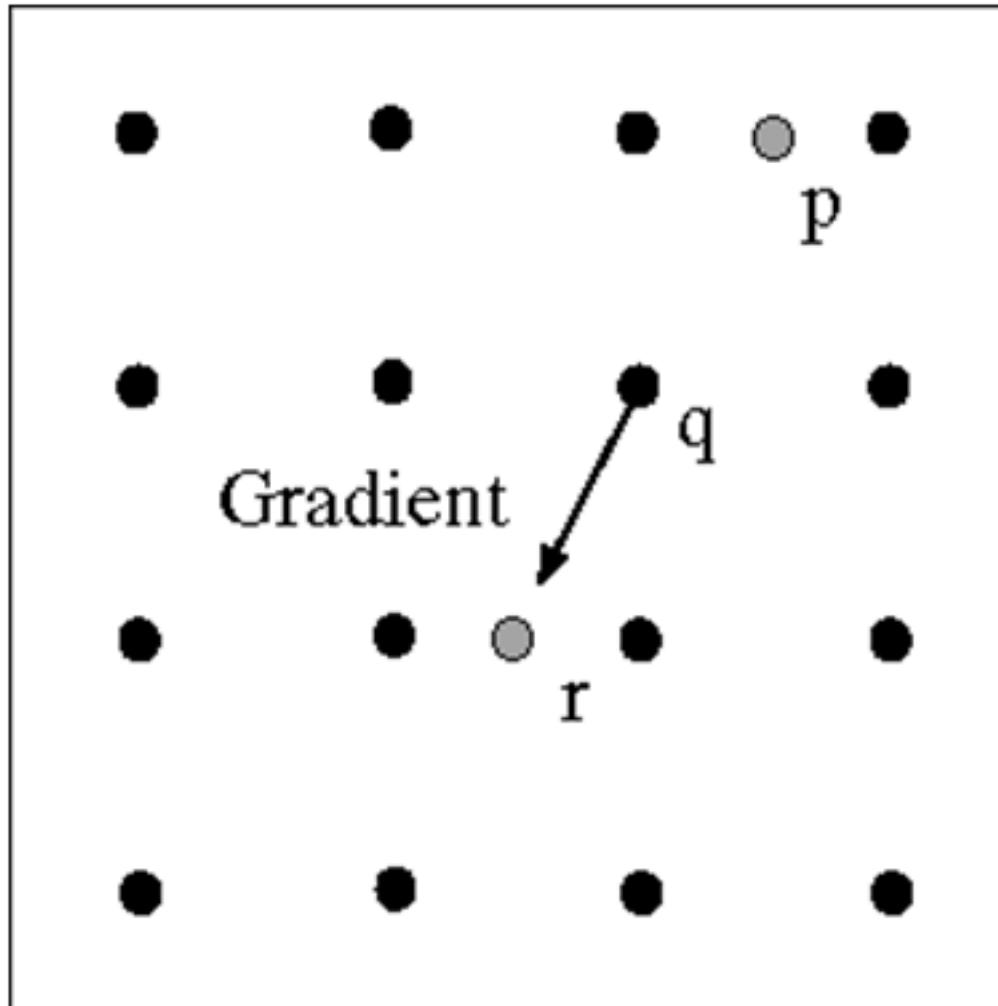
Пример



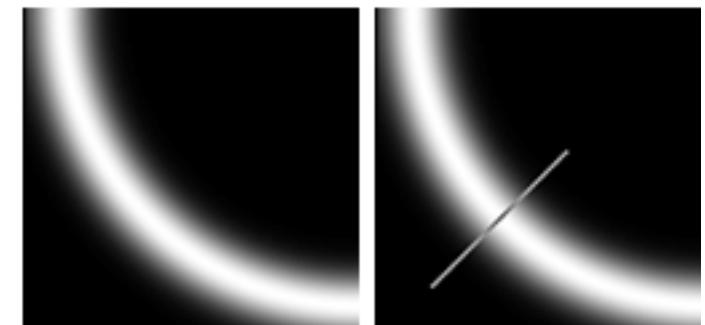
Отсечение по порогу



Поиск локальных максимумов

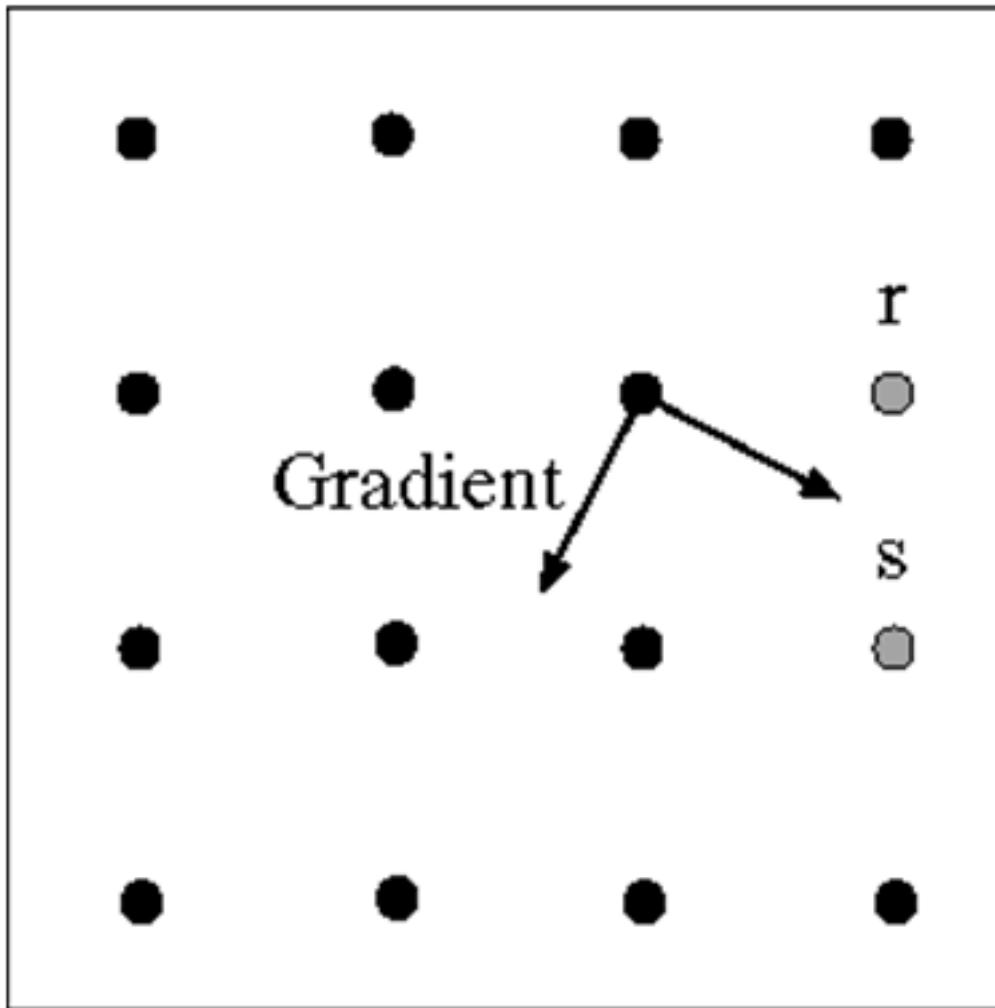


Максимум
достигается в q ,
если значение
больше p и r .
Значения в p и r
интерполируем.

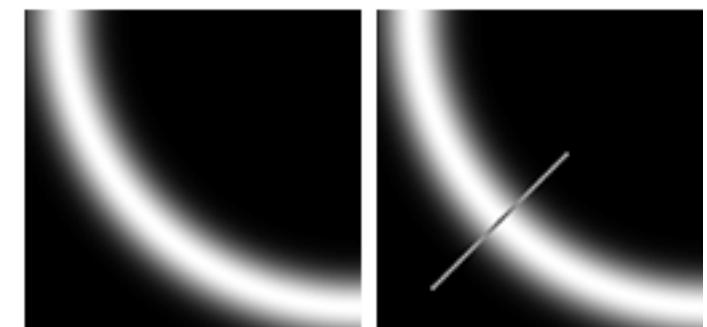




Связывание точек



Пусть отмеченная точка – край. Строим касательную к границе (нормаль к направлению градиента) и используем ее для предсказания новой точки (это либо s либо r).





Пример

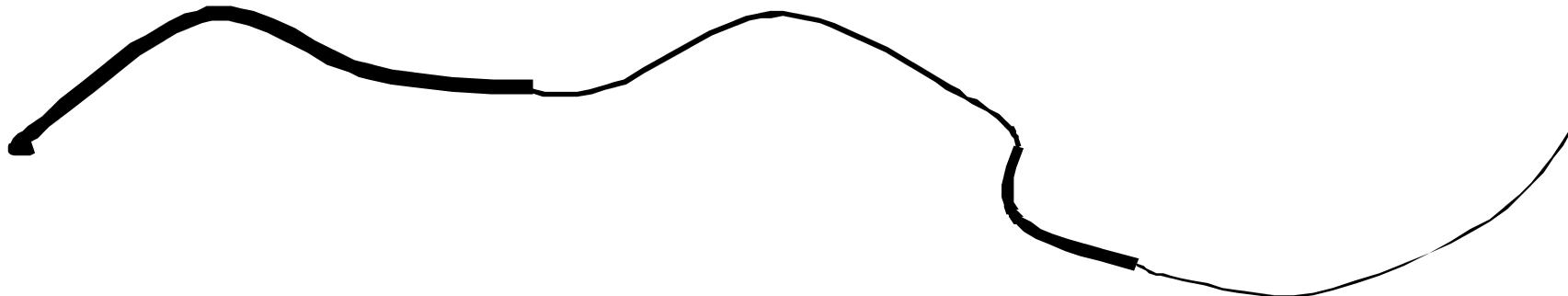


Утоньшение
(non-maximum suppression)



Отсечение по порогу

- Проверяем точку, чтобы значение градиента было выше порога
 - Используем **гистерезис**
 - Большой порог для начала построения кривой и низкий порог для продолжения края (связывания)





Эффект гистерезиса



Исходное изображение



Высокий порог
(сильные края)



Низкий порог
(слабые края)

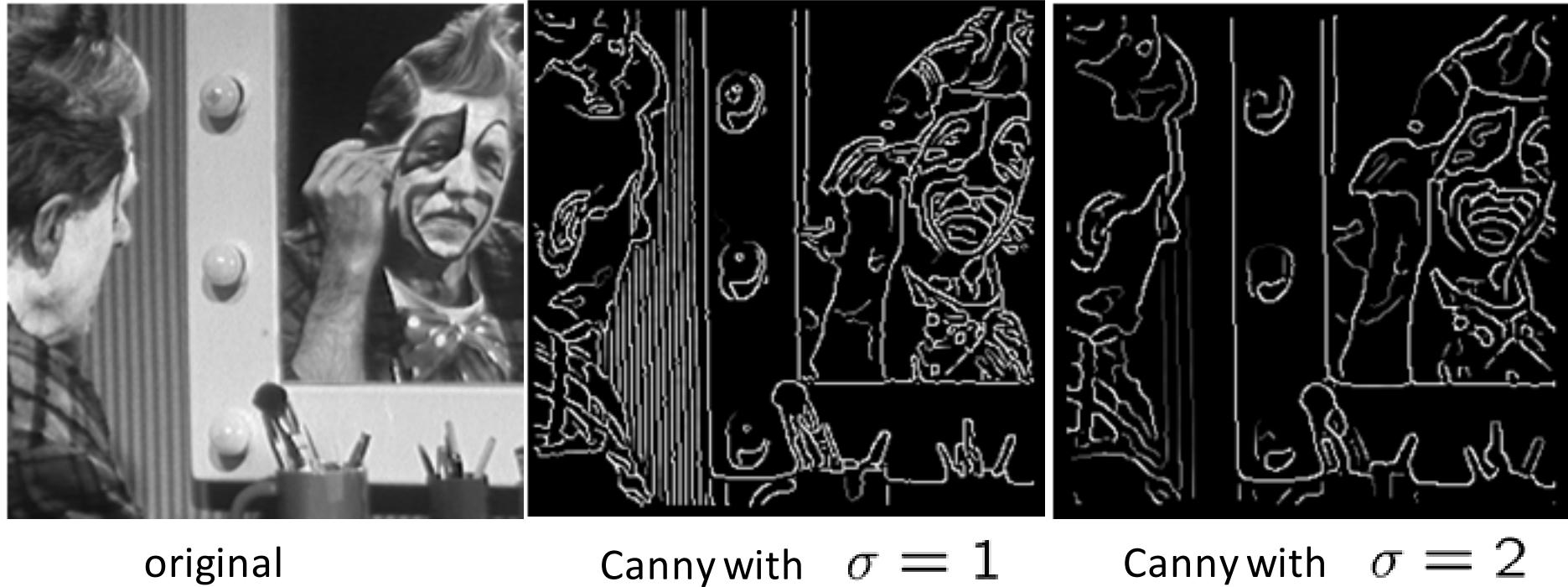


Порог по гистерезису

Source: L. Fei-Fei



Влияние σ (Размер ядра размытия)



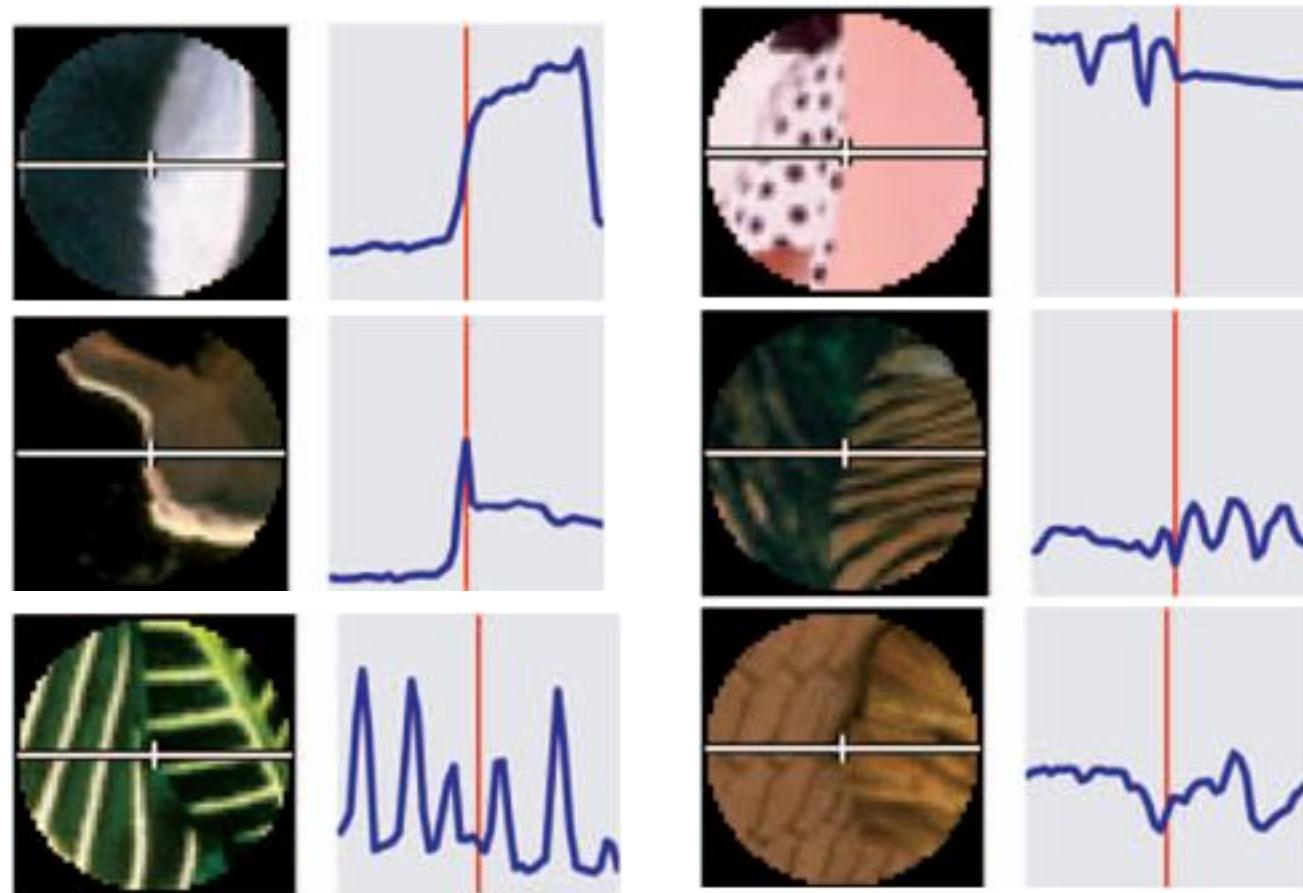
Выбор σ зависит от задачи

- большое σ - поиск крупных границ
- маленькое σ - выделение мелких деталей

Source: S. Seitz



Ограничения детектора Canny



Градиент яркости – это лишь часть информации, которая используется человеком для определения границ между объектами.

Source: Martin et al. 2003



Спецэффекты

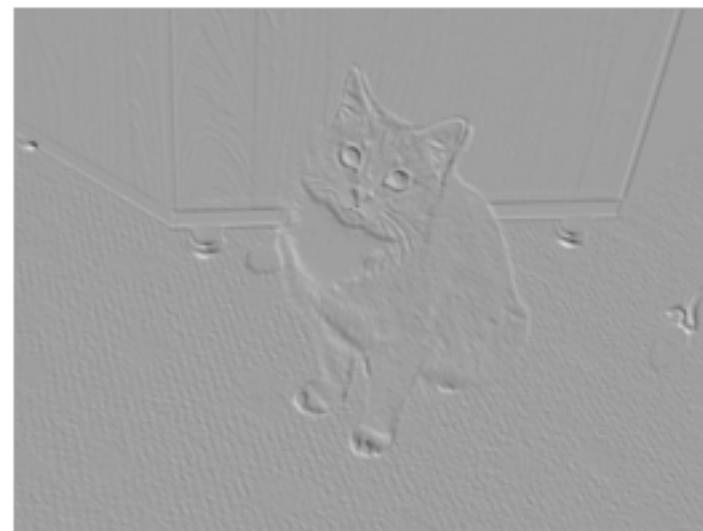
- Рассмотрим
 - Тиснение
 - Негатив
 - «Светящиеся» края
 - Геометрические эффекты
 - Перенос/поворот
 - Искажение
 - «Эффект стекла»



Тиснение

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

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»



«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»


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



Резюме лекции

- Цветокоррекция изображения требует оценки до 24 параметров нелинейной модели
 - Калибровочный цветовой шаблон
 - Угадывание (оценка) параметров
- Линейная фильтрация (свёртка) изображения позволяет решать целый ряд задач – шумоподавление, повышение резкости, оценка градиента
- Задачу выделение краёв изображения в простом случае можно решить как поиск локальных максимумов градиента яркости
- Всё это открытые задачи и сейчас активно продолжают исследоваться!