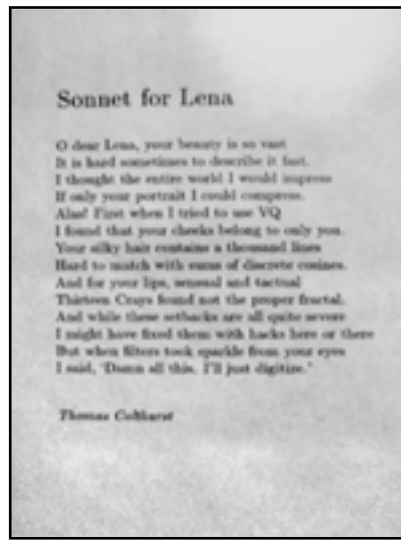
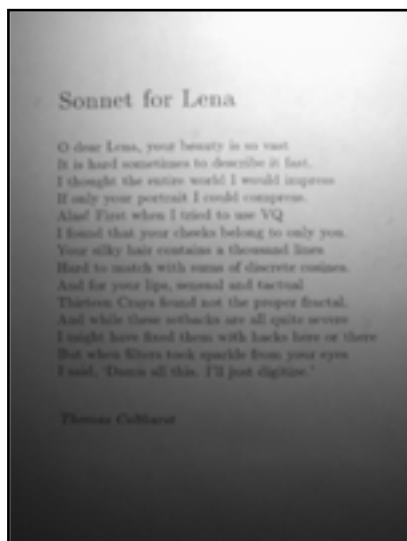




Основы обработки изображений



АНТОН КОНУШИН

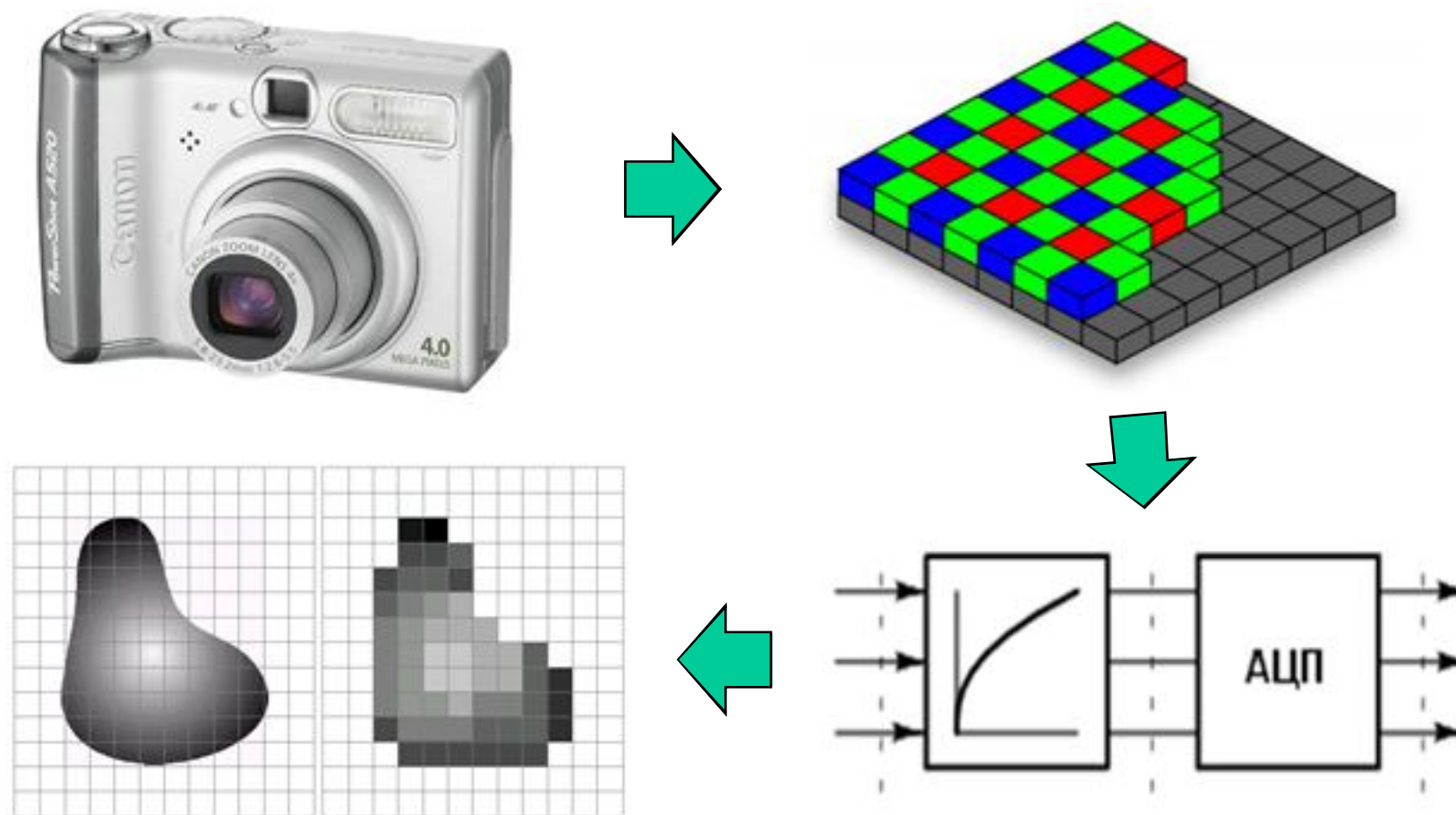


Обработка изображений

- 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



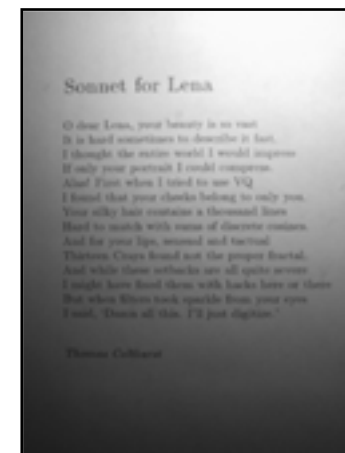
Неправильные цвета



Шумное



Нерезкое



Неравномерно
освещённое



Тональная коррекция



Постоянство цвета и освещенности



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

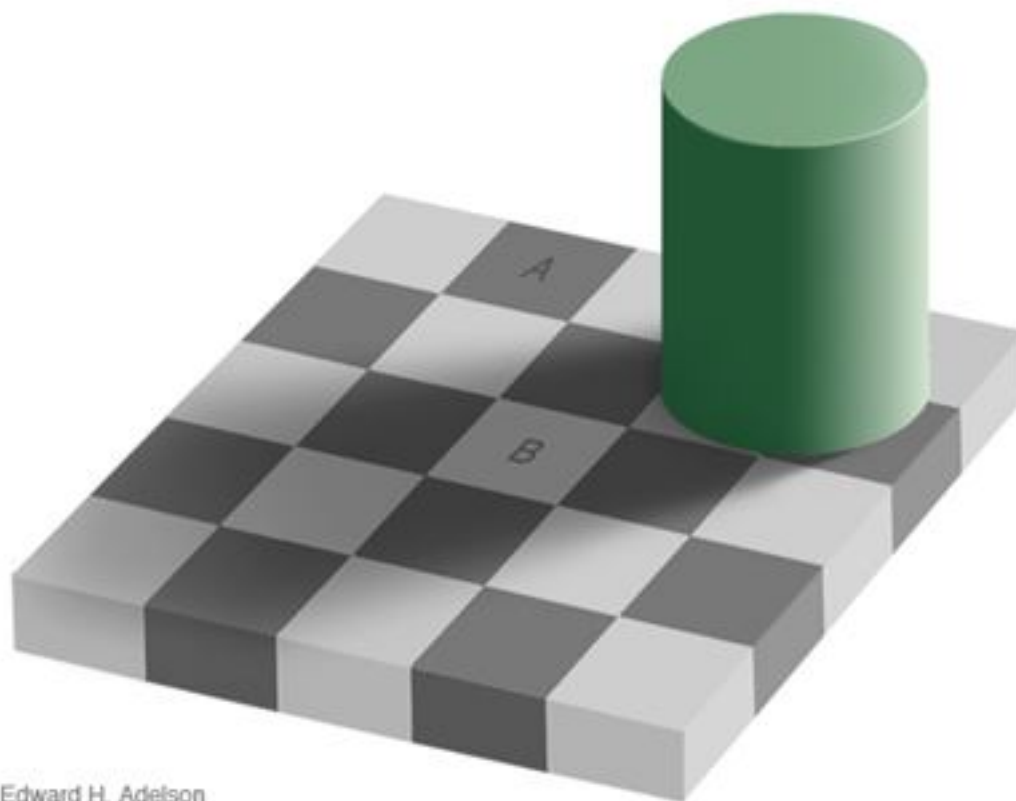
Освещённость:

- Полнолуние – 0.27 люкс
- Пасмурный день – 100-1000 люкс
- Яркий день (в тени) – 20000 люкс
- На солнце - 100000

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



Постоянство яркости

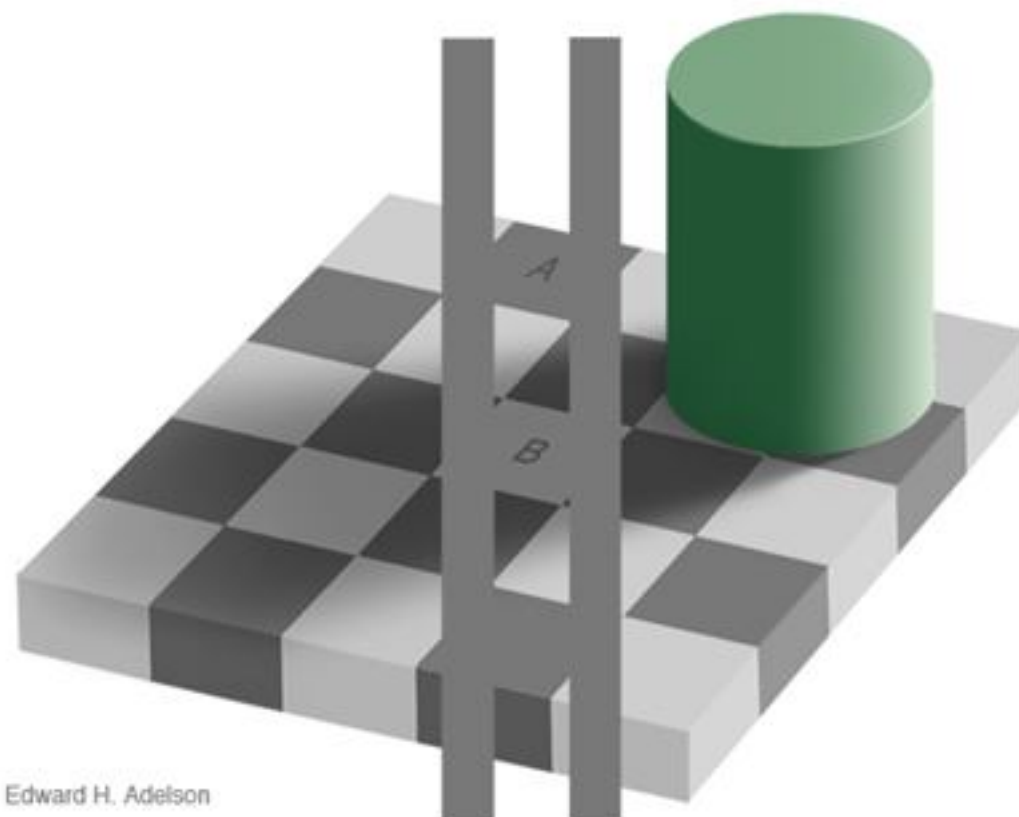


Edward H. Adelson

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



Постоянство яркости



Edward H. Adelson

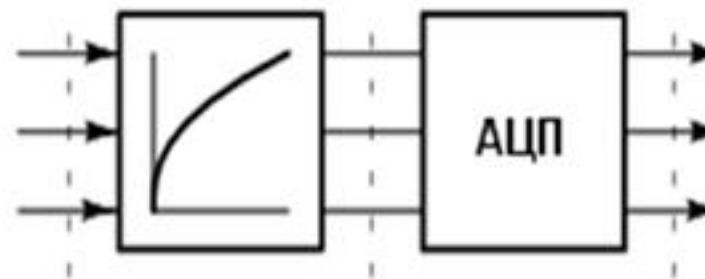
http://web.mit.edu/persci/people/adelson/checkersshadow_illusion.html

Slide by S. Lazebnik



Причины плохой передачи яркости

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



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



Гистограмма и её оценка

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



- Не полностью используется диапазон яркостей
- Концентрация яркостей вокруг определенных значений - неравномерное заполнение диапазона яркостей)



Точечные операторы

Оператор, который определяет значение выходного пиксела по значению только одного входного пиксела

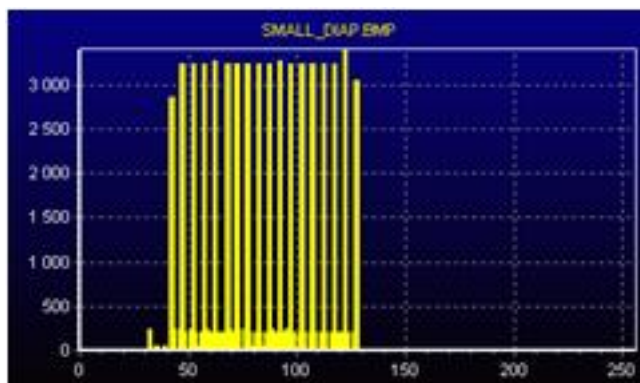
$$f^{-1}(y) = x$$

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



Линейная коррекция

Компенсация узкого диапазона яркостей – **линейное растяжение гистограммы** (Histogram equalization):



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

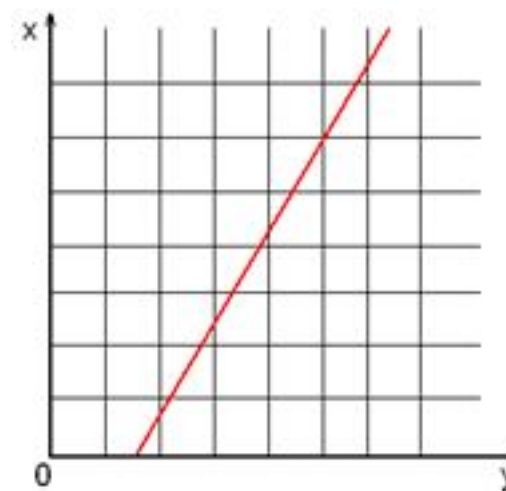
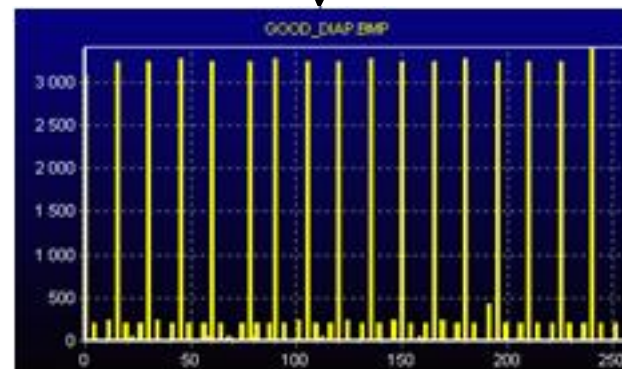
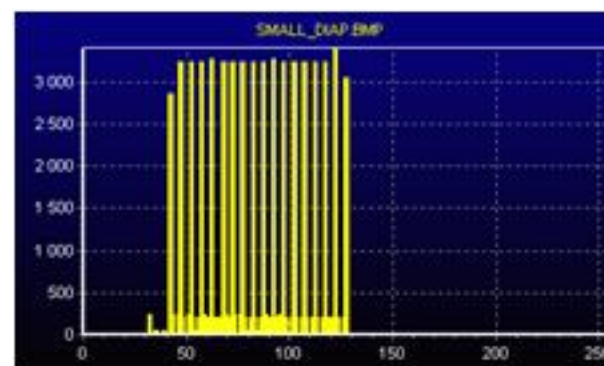


График функции $f^{-1}(y)$



Линейная коррекция

Компенсация узкого диапазона яркостей – линейное растяжение гистограммы:





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

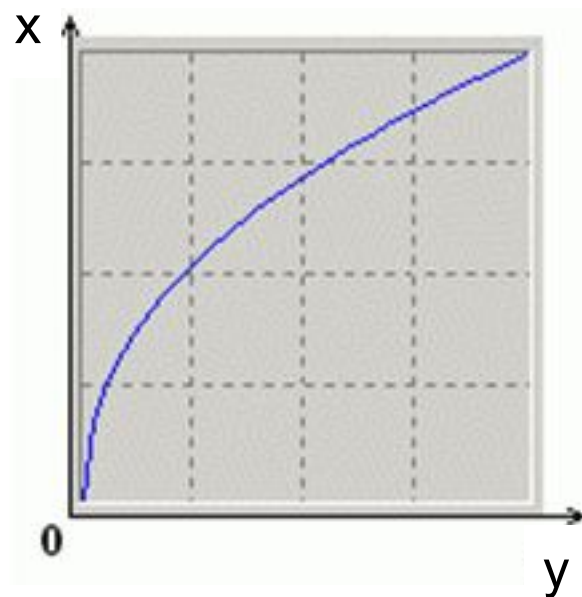


График функции $f^{-1}(y)$



Нелинейная коррекция

Часто применяемые функции:

- **Гамма-коррекция**

- Изначальная цель – коррекция для правильного отображения на мониторе.

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

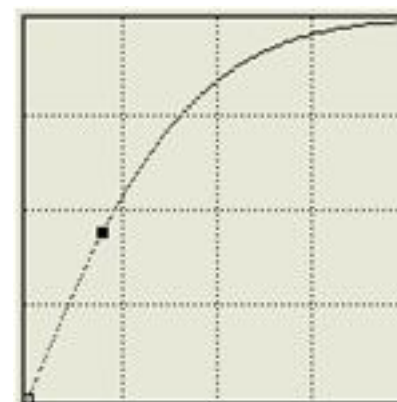
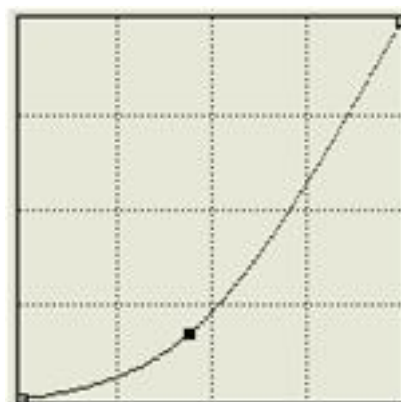
- **Логарифмическая**

- Цель – сжатие динамического диапазона при визуализации данных

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



Гамма-коррекция



Графики функции $f^{-1}(y)$



Нелинейная коррекция

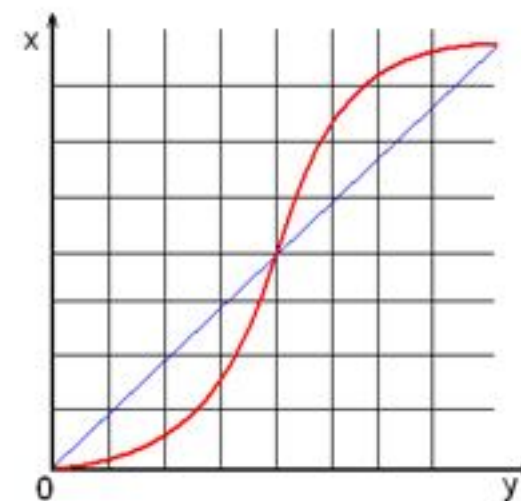
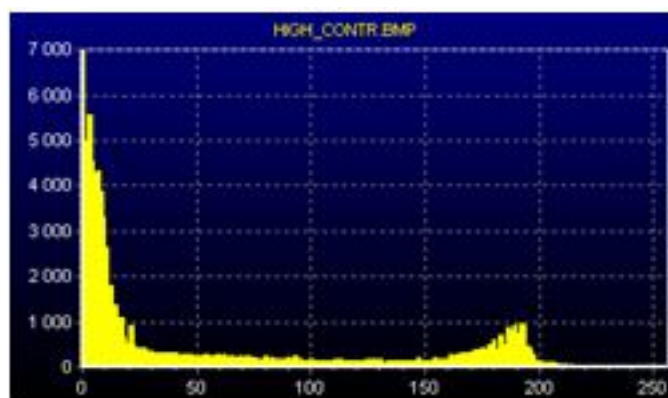
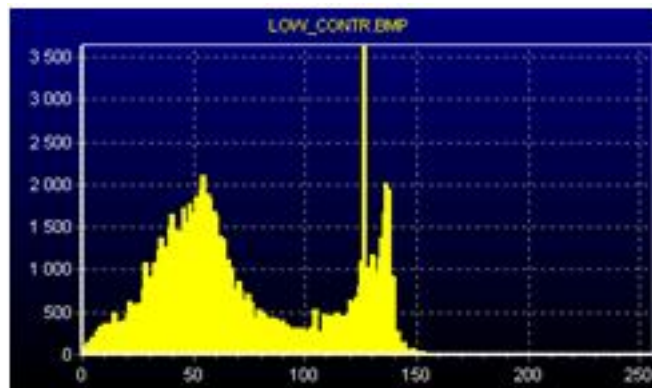


График функции $f^{-1}(y)$



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



Как
скорректировать
изображение?



Коррекция по шаблону

- Разумный подход:
 - Сфотографировать объект с известным цветом (шаблон)
 - Вычислить цветовое преобразование, чтобы цвет объекта на фотографии совпал с нужным
- Простейшая реализация:
 - Возьмём однотонные карточки (белые)
 - Будем домножать каждый канал отдельно, чтобы цвет карточек стал белым
 - Вычисление коэффициентов - Если цвет объект записывается как r_w, g_w, b_w , тогда веса $1/r_w, 1/g_w, 1/b_w$



Насколько такое преобразование корректно, какие могут быть недостатки?



Профессиональная цветокоррекция



Source: The dark knight



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

Используем цветной шаблон с многими цветами

Какое преобразование в камере?



Сложные модели



<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 цветowych 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





Поканальное растяжение

- 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:

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

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

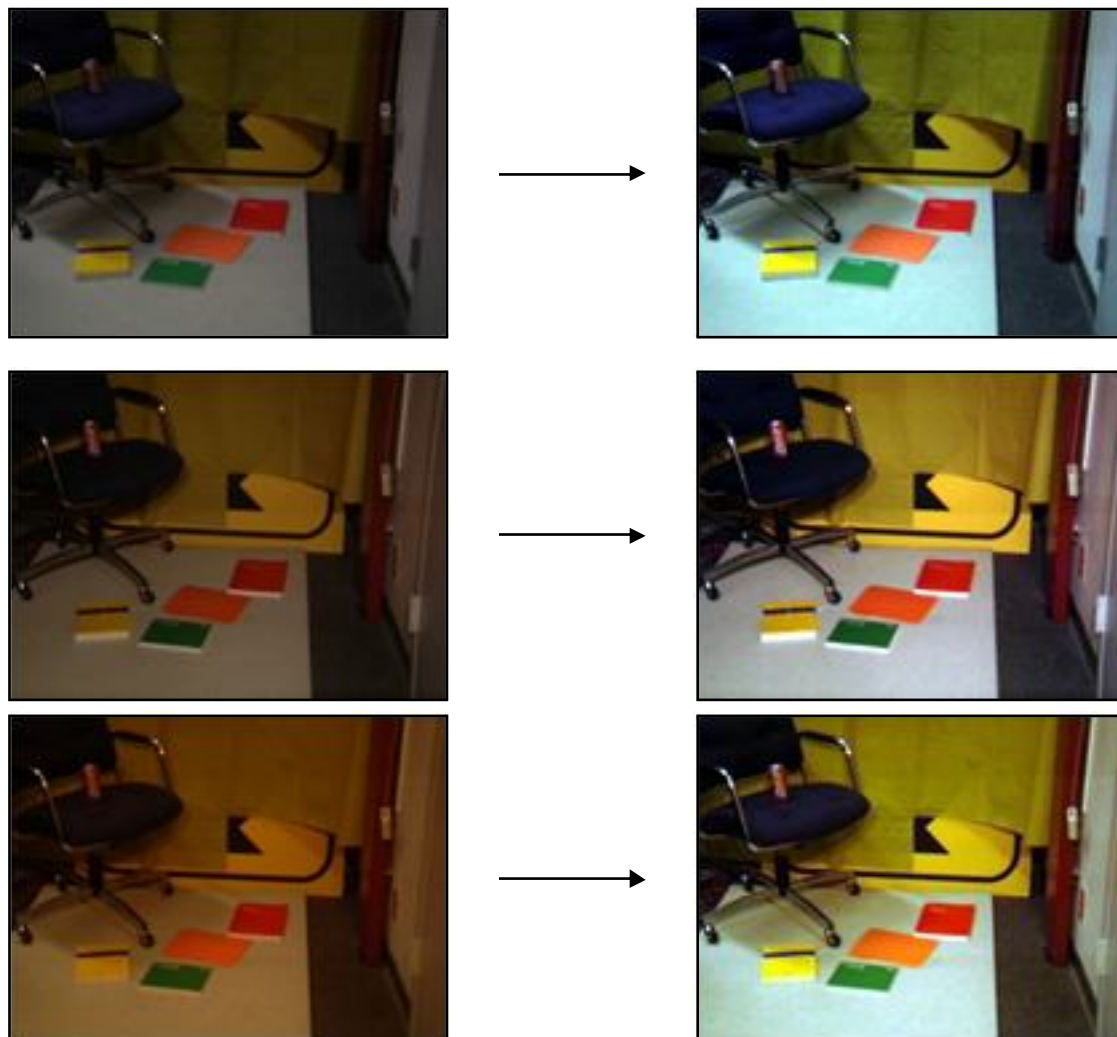


Растяжение контрастности



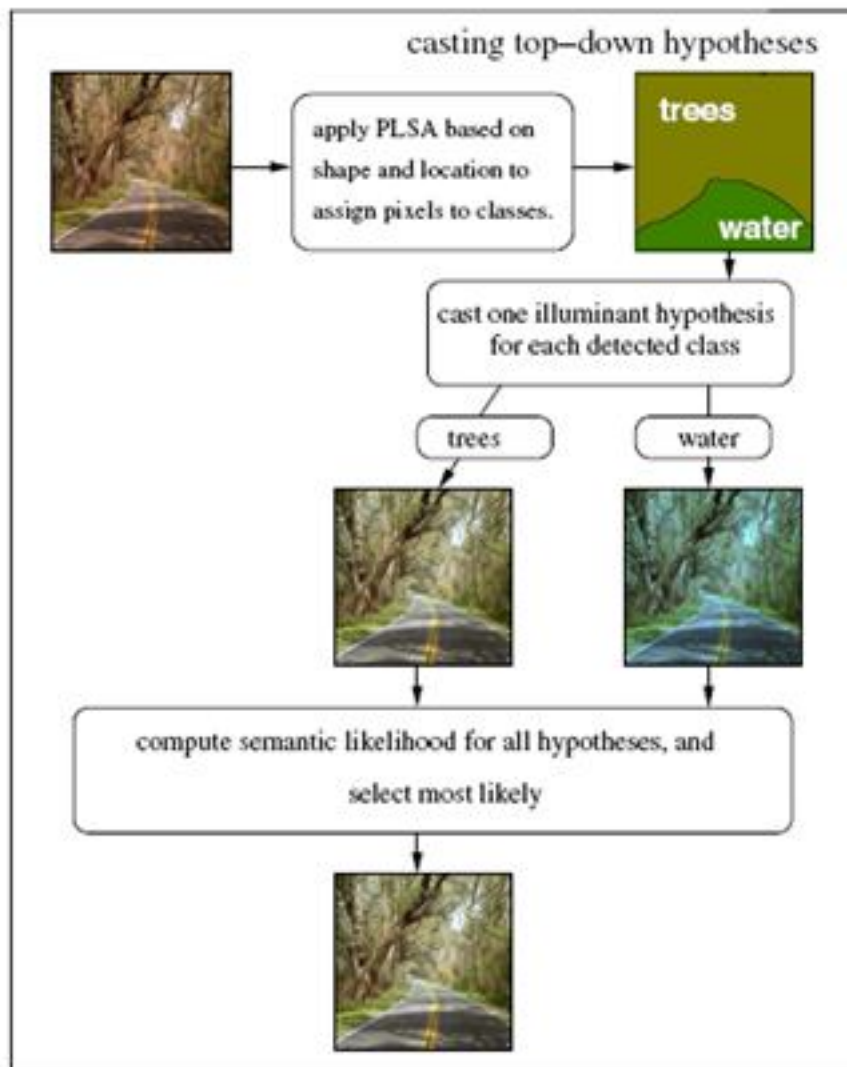


Растяжение контрастности





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 «типичных» изображениях



Шумоподавление



Шумоподавление

Причины возникновения шума:

- Несовершенство измерительных приборов
- Хранение и передача изображений с потерей данных



Шум фотоаппарата



Сильное сжатие JPEG



Виды шума



Original



Salt and pepper noise



Impulse noise



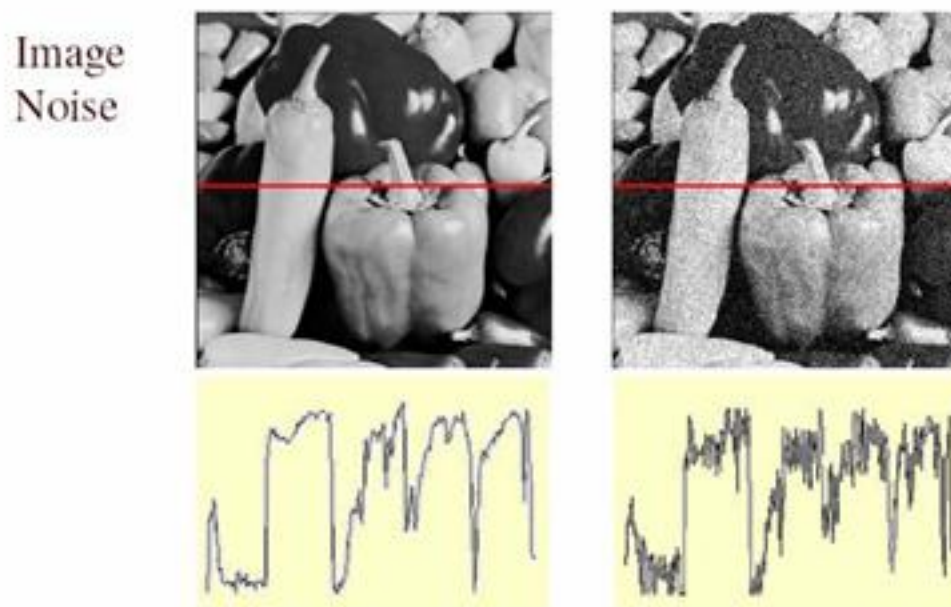
Gaussian noise

- **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



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

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



Временная фильтрация

Усреднение пикселов (измерений) в разные моменты времени



Серия зашумленных изображений

Результат усреднения по 10 изображениям

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

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

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



Пространственная фильтрация

- Усреднение по окрестности пиксела
- Заменяем каждый пиксель взвешенным средним по окрестности
- Веса обозначаются как *ядро фильтра*
- Для каждого пикселя веса (ядро) одинаковые!
- Пример простейшего фильтра для шумоподавления:

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

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

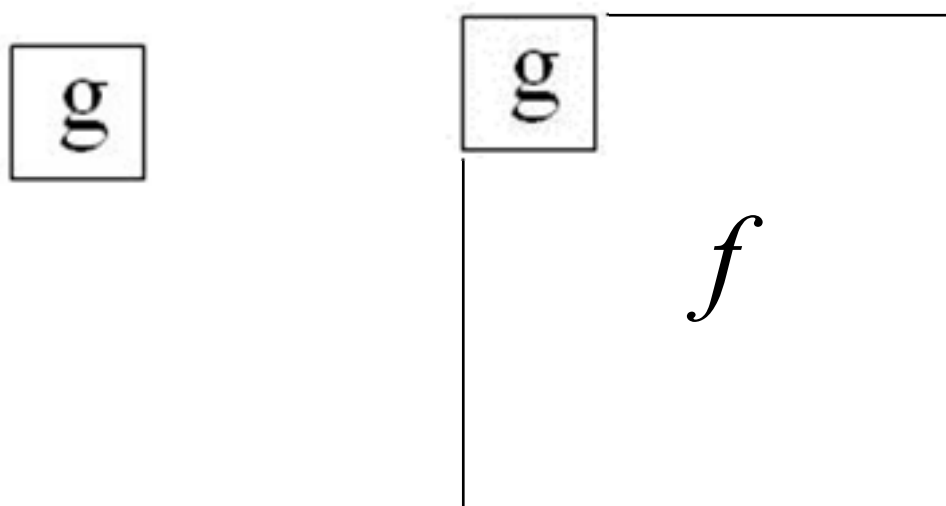
“box filter”



Свертка

- Такое взвешенное усреднение называется «свёрткой»
- Пусть f – изображение, g -ядро. Свертка изображения f с помощью g обозначается как $f * g$ и называется:

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





Линейная фильтрация

- Линейным фильтром называется такое преобразование изображения, которое удовлетворяет 2м свойствам:
 - $\text{filter}(f_1 + f_2) = \text{filter}(f_1) + \text{filter}(f_2)$
 - $\text{filter}(a * f_1) = a * \text{filter}(f_1)$
- Инвариантность к сдвигу: фильтрация не зависит от сдвига пиксела: $\text{filter}(\text{shift}(f)) = \text{shift}(\text{filter}(f))$
- Теорема: любой линейный оператор (фильтр), инвариантный к сдвигу, может быть записан в виде свертки
- Чтобы доказать нелинейность фильтра, можно воспользоваться основными свойствами, и показать их не выполнение на примере



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





Полезные свойства

- 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:
$$a * (b + c) = (a * b) + (a * c)$$



Примеры простых фильтров



Original

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

?



Примеры простых фильтров



Original

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |



Filtered
(no change)



Примеры простых фильтров



Original

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 0 | 0 |

?



Примеры простых фильтров



Original

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 0 | 0 |



Shifted left
By 1 pixel



Примеры простых фильтров



Original

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

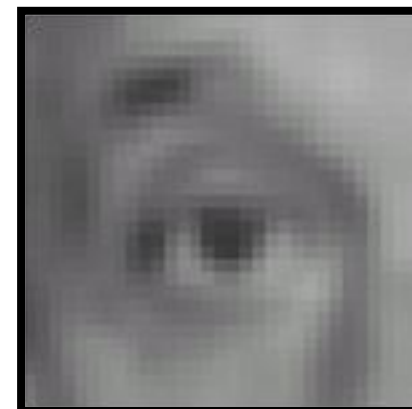
?



Примеры простых фильтров



Исходное

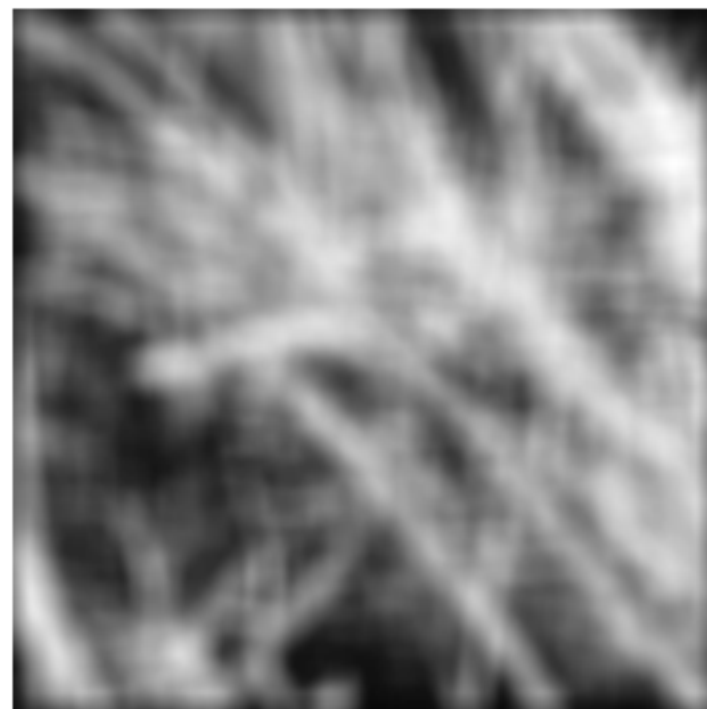
$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$


Результат



Сглаживание с box-фильтром

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

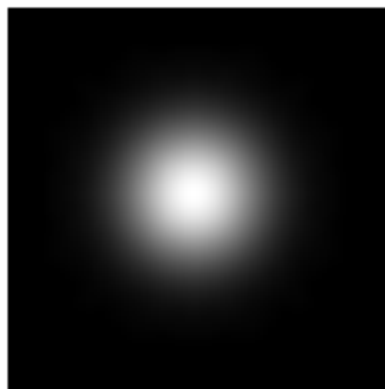
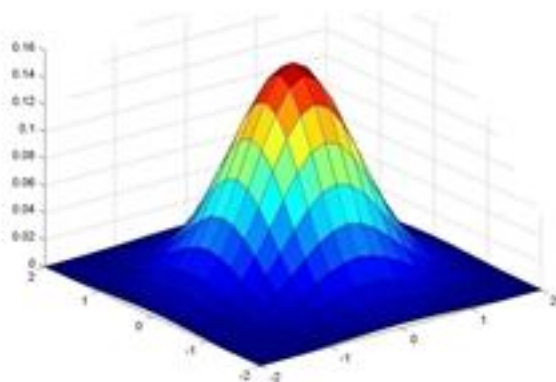




Фильтр Гаусса

Задаём вес пикселей при
сглаживании с учётом
близости к центру:

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



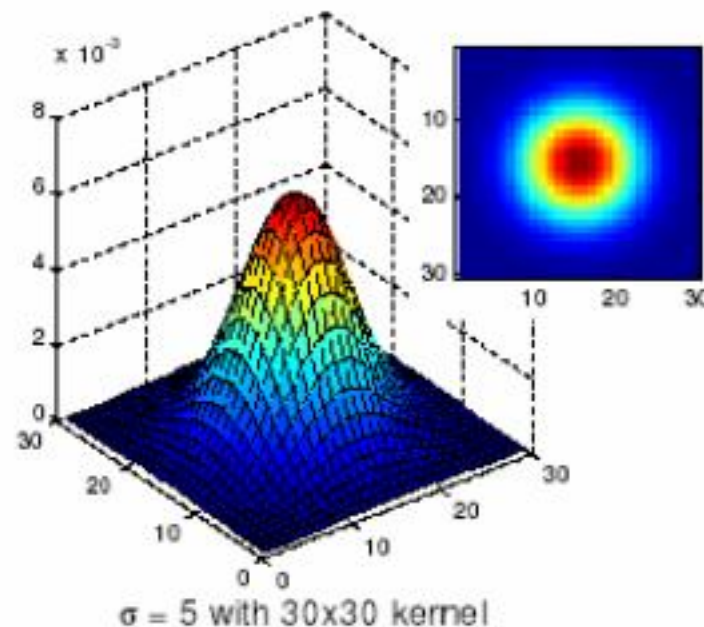
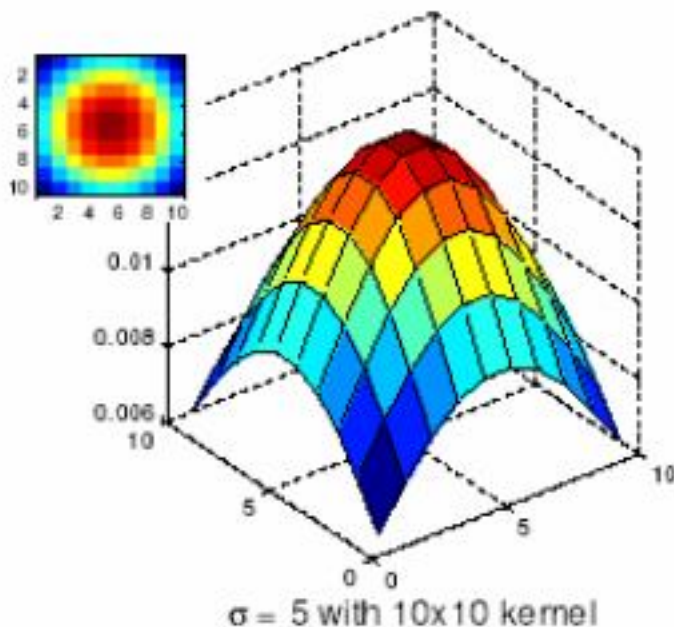
| | | | | |
|-------|-------|-------|-------|-------|
| 0.003 | 0.013 | 0.022 | 0.013 | 0.003 |
| 0.013 | 0.059 | 0.097 | 0.059 | 0.013 |
| 0.022 | 0.097 | 0.159 | 0.097 | 0.022 |
| 0.013 | 0.059 | 0.097 | 0.059 | 0.013 |
| 0.003 | 0.013 | 0.022 | 0.013 | 0.003 |

5 x 5, $\sigma = 1$



Выбор размера ядра

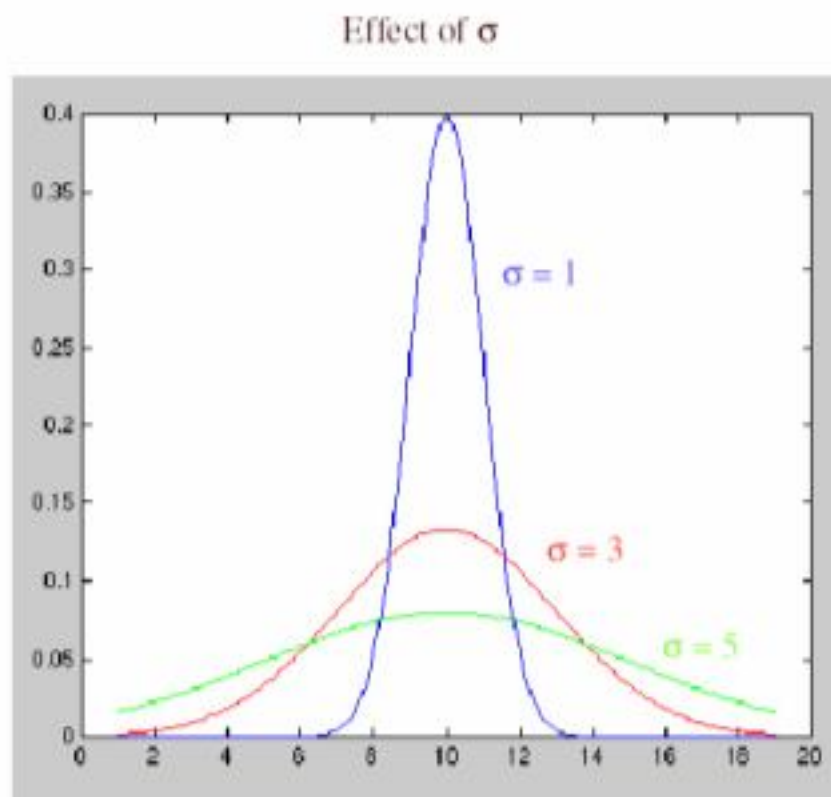
Размер ядра дискретного фильтра ограничен





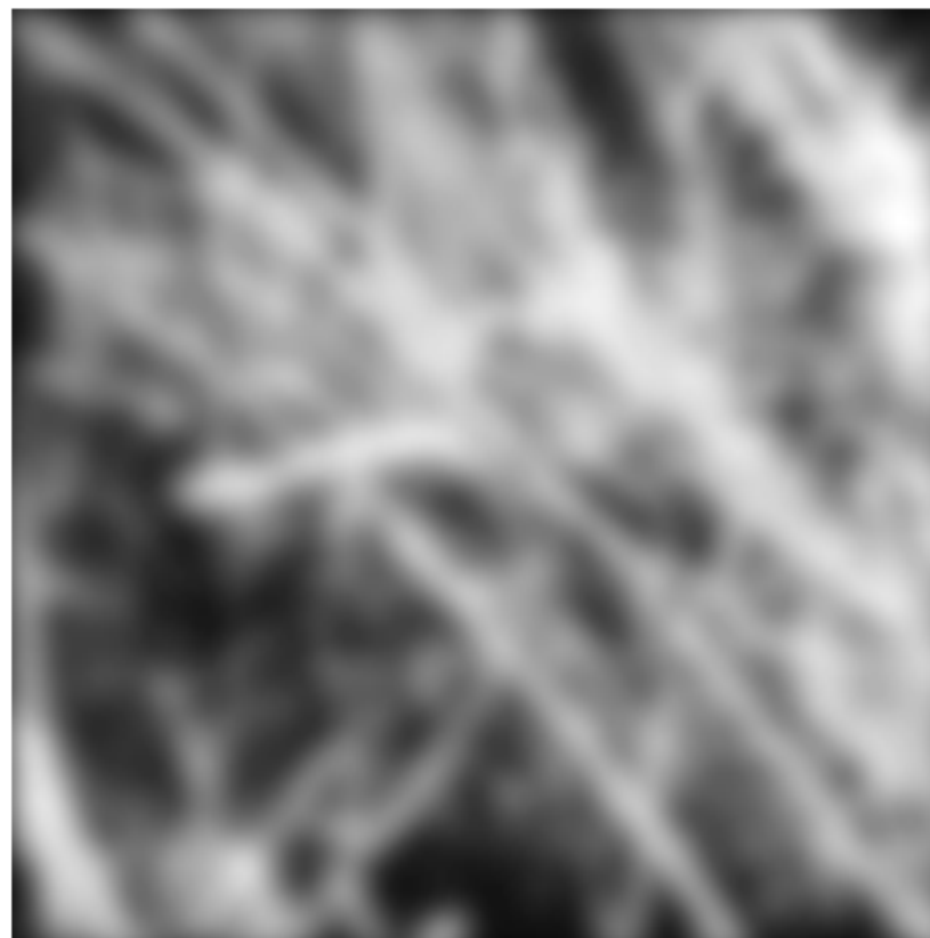
Выбор размера ядра

Эмпирика: полуразмер фильтра равен 3σ



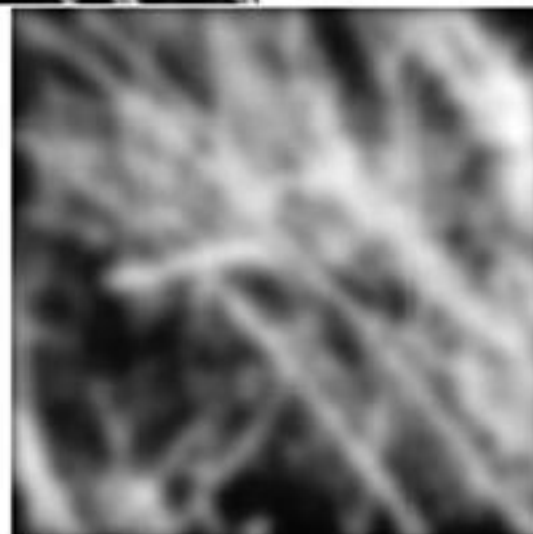
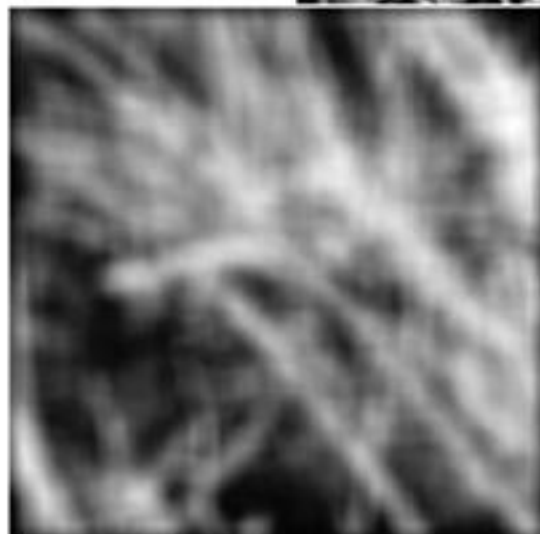


Сглаживание фильтром гаусса



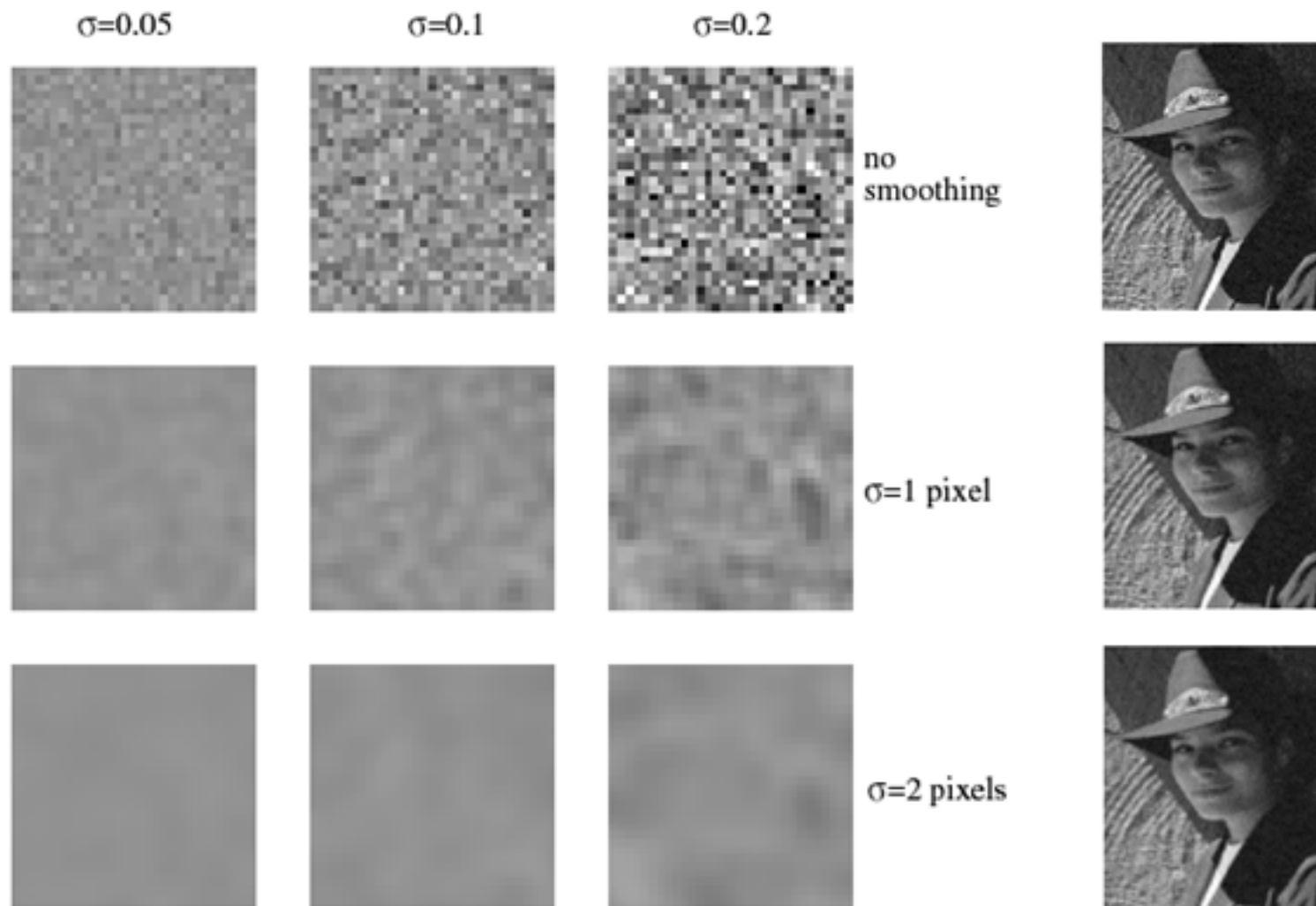


Сравнение





Подавление гауссова шума



Сглаживание фильтрами большого радиуса подавляет шум, но размывает изображение

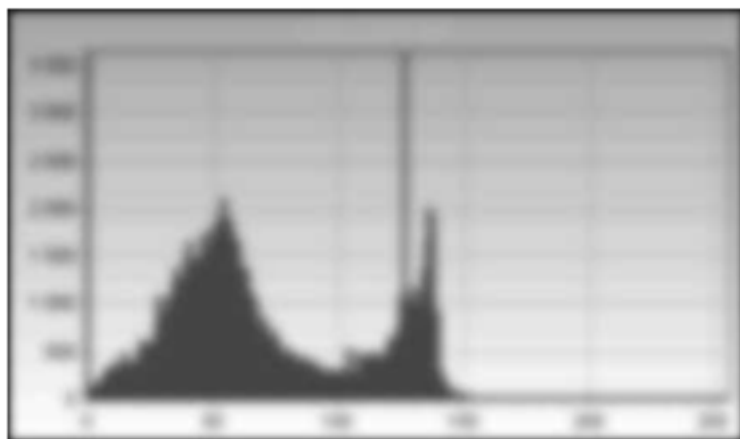


Свойства фильтра Гаусса

- 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 $\sigma\sqrt{2}$

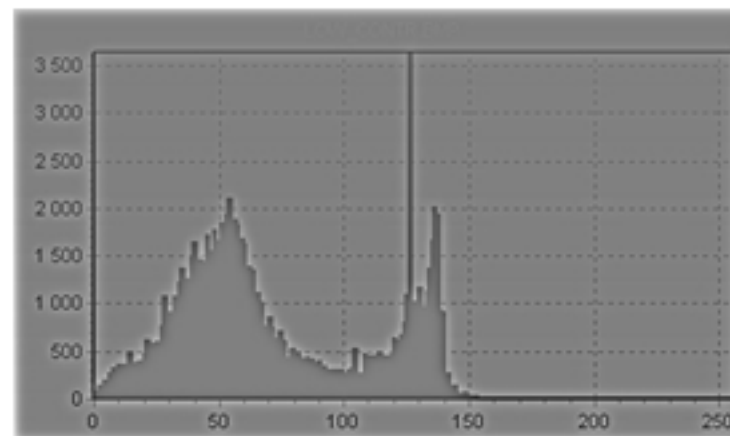


Маленькая экскурсия к Фурье

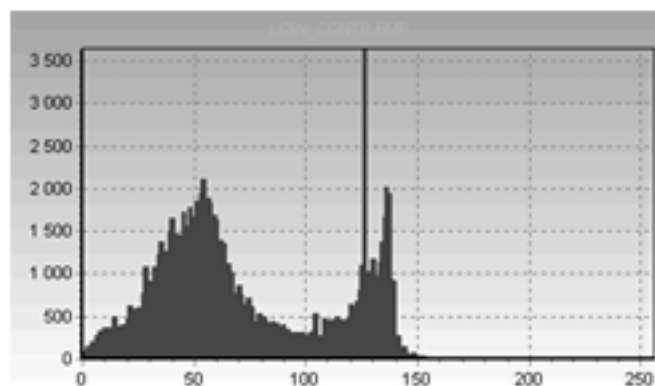


Низкие частоты

+



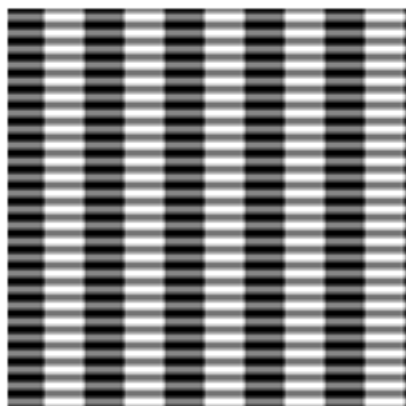
Высокие частоты



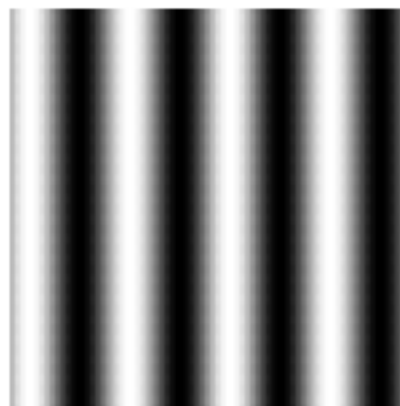


Фильтр Гаусса

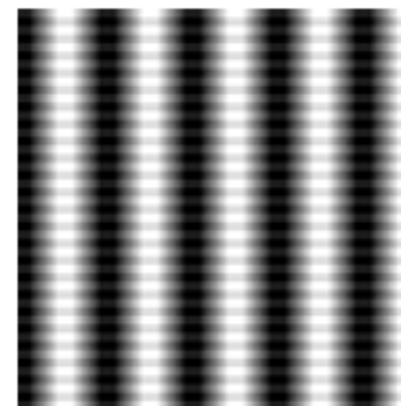
Результат свертки фильтром гаусса и усреднения



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

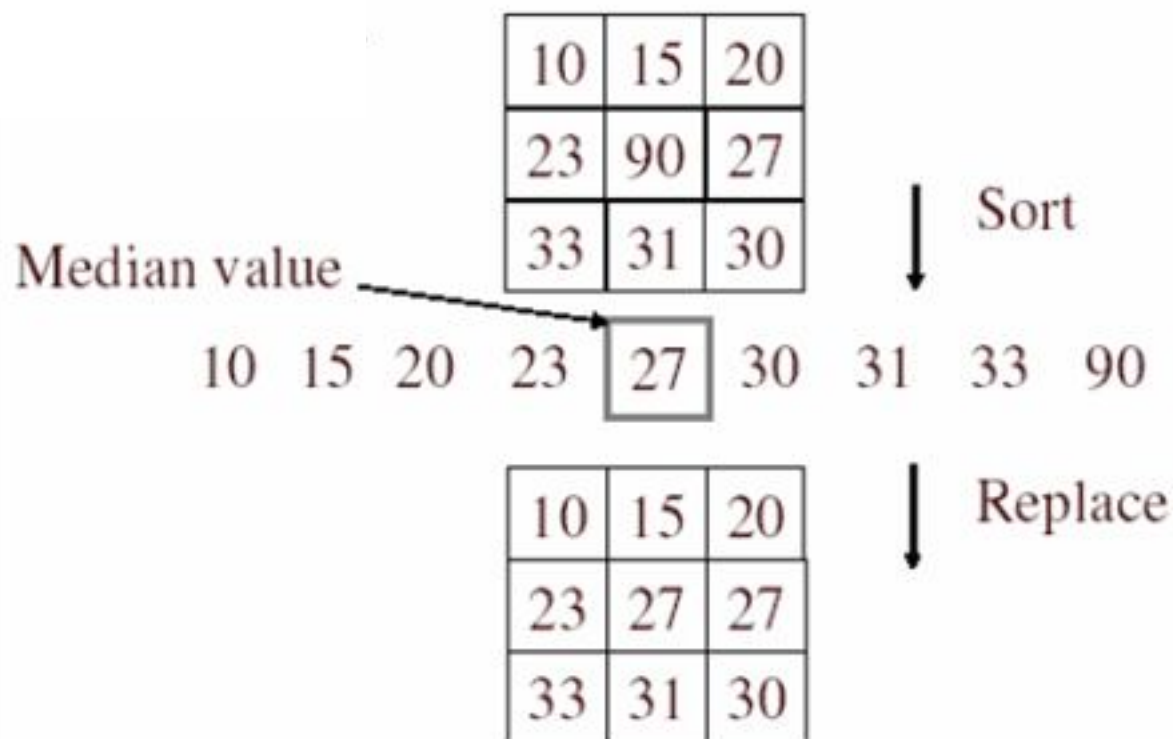


Применим фильтр Гаусса
Чем результат плох?



Медианный фильтр

Выбор медианы из выборки пикселей по окрестности данного

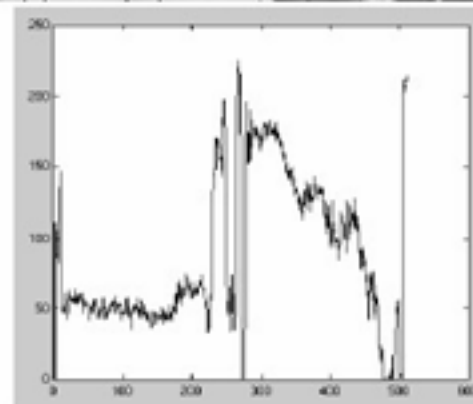
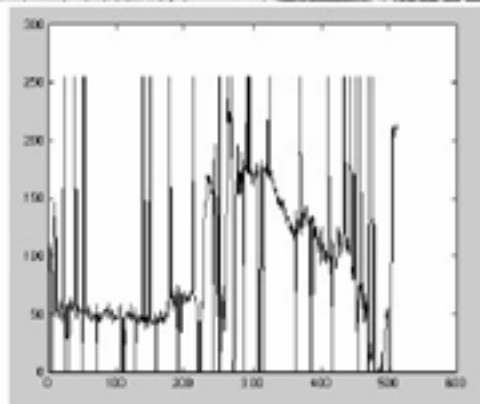
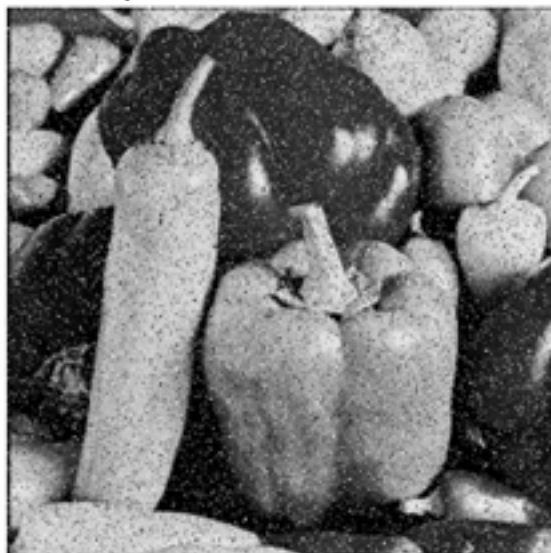


- Является ли фильтр линейным?



Медианный фильтр

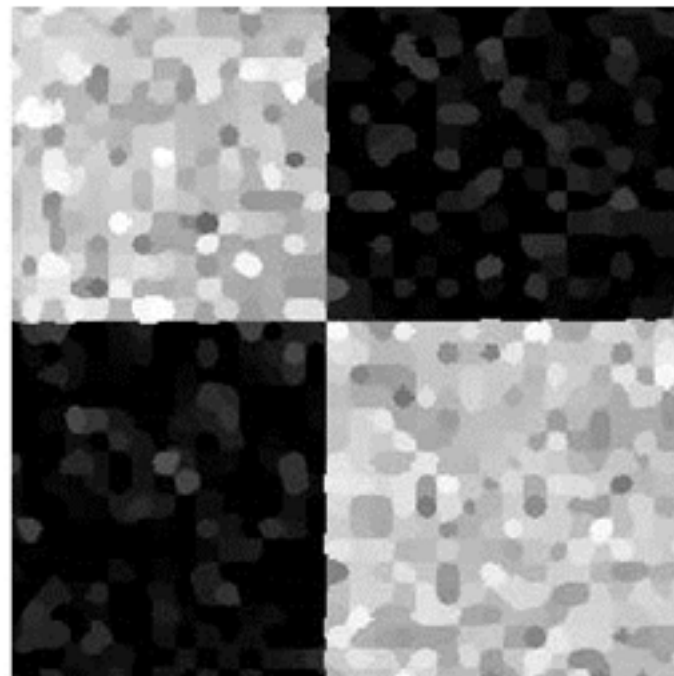
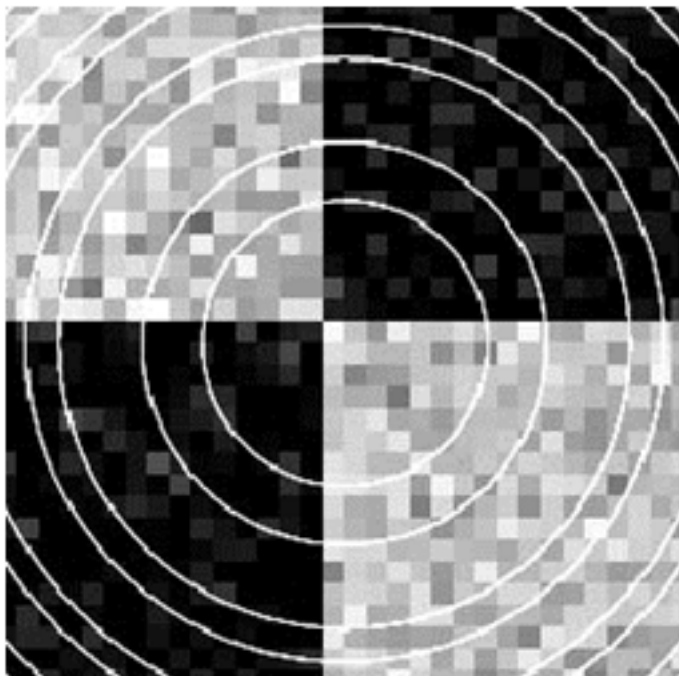
Шум «соль и перец» Медианная фильтрация





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





Современные методы

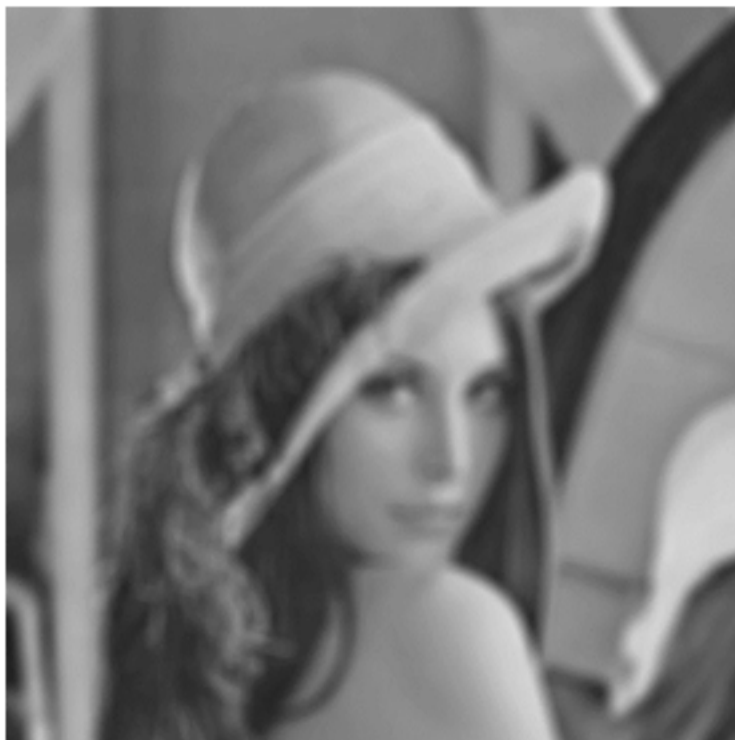
- Сейчас методы фильтрации изображений продолжают активно развиваться
- Можно выделить подходы:
 - Вейвлет и т.д. представления
 - Разреженные представления
 - Марковские случайные поля



Другие применения фильтрации



Повышение резкости

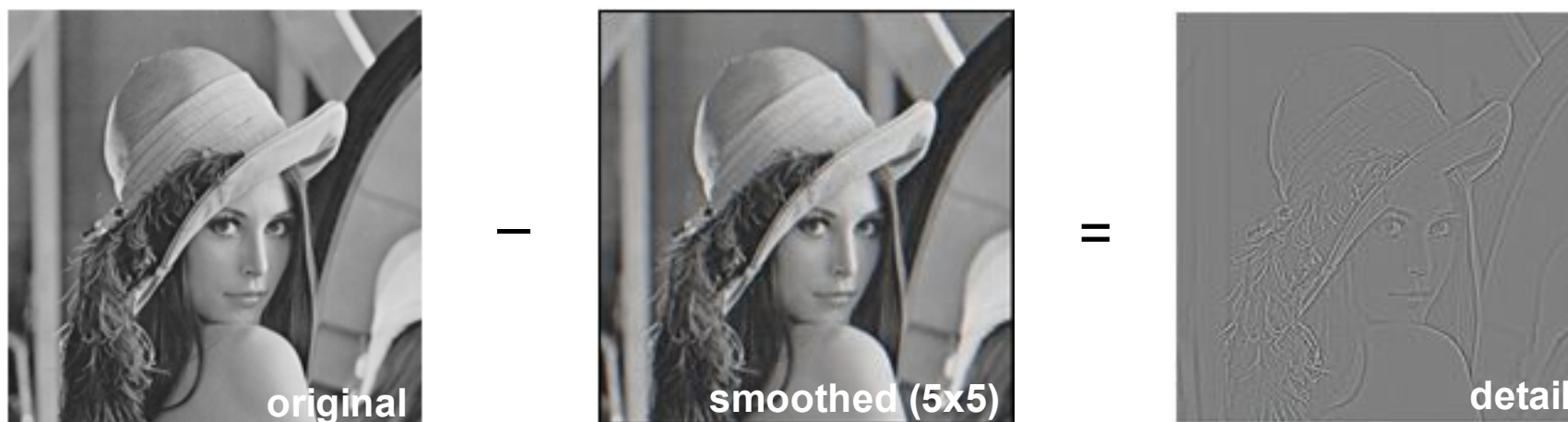


Иногда с помощью
фильтрации можно немного
повысить резкость
изображения (подчеркнуть
края)



Повышение резкости

Что теряется при сглаживании?



Добавим дополнительно высокие частоты:





Пример повышения резкости

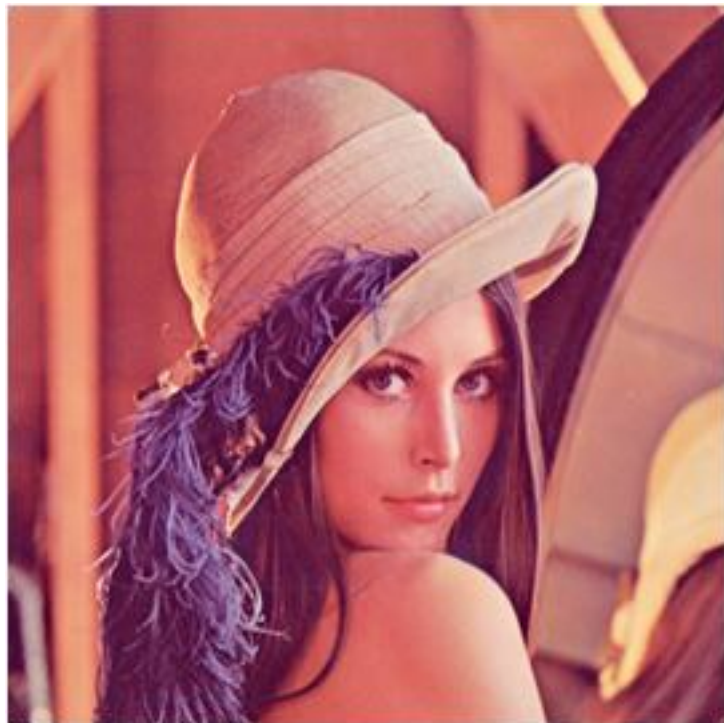
Ядро
свертки

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





Лирическое отступление



Lena (Lenna)

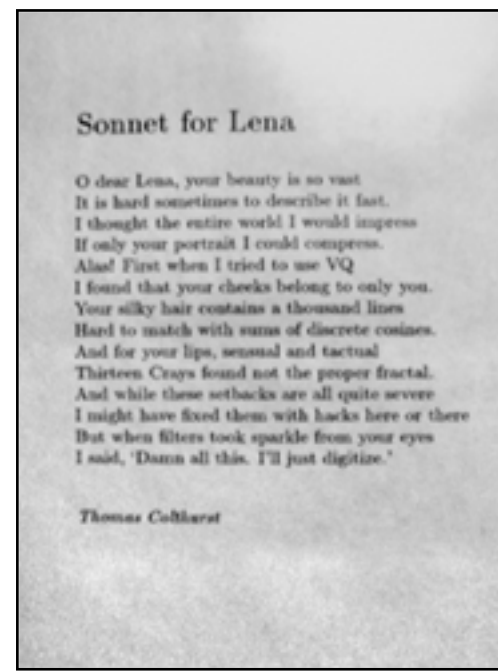
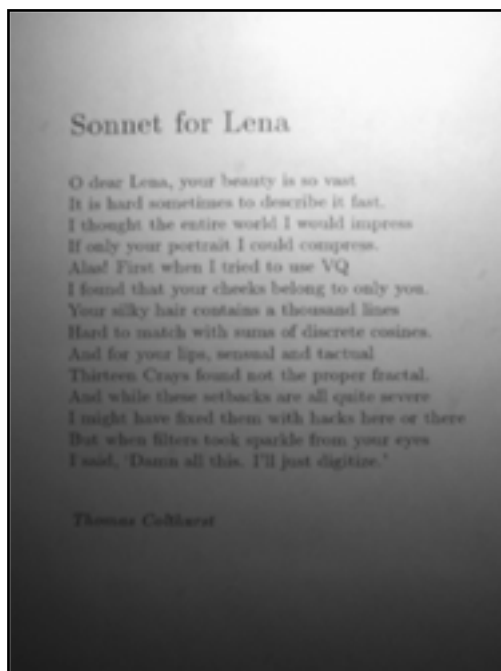
- Обрезанная (512x512) часть изображения с разворота Playboy, Nov 1972
- Самый популярный, но не первый случай использования Playboy в обработке изображений (первый в 1961)
- Пригласили на 50ую конференцию [Society for Imaging Science and Technology](#) (IS&T) in 1997

<http://en.wikipedia.org/wiki/Lenna>



Компенсация разности освещения

Пример





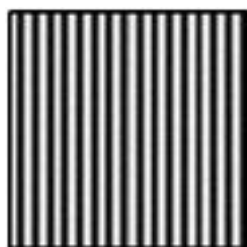
Компенсация разности освещения

Идея:

Формирование изображения:

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

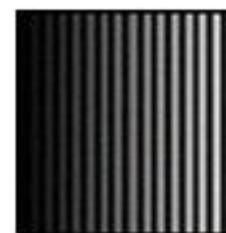
Плавные изменения яркости относятся к освещению,
резкие - к объектам.



объект $r(i, j)$



освещение $l(i, j)$



Изображение
освещенного
объекта $I(i, j)$



Алгоритм Single scale retinex (SSR)

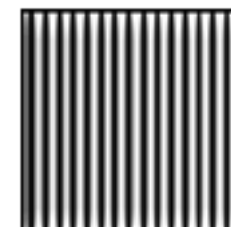
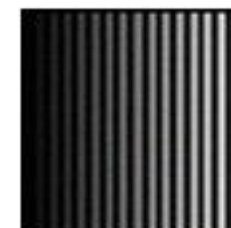
- Получить приближенное изображение освещения путем низочастотной фильтрации

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

- Восстановить изображение по формуле

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

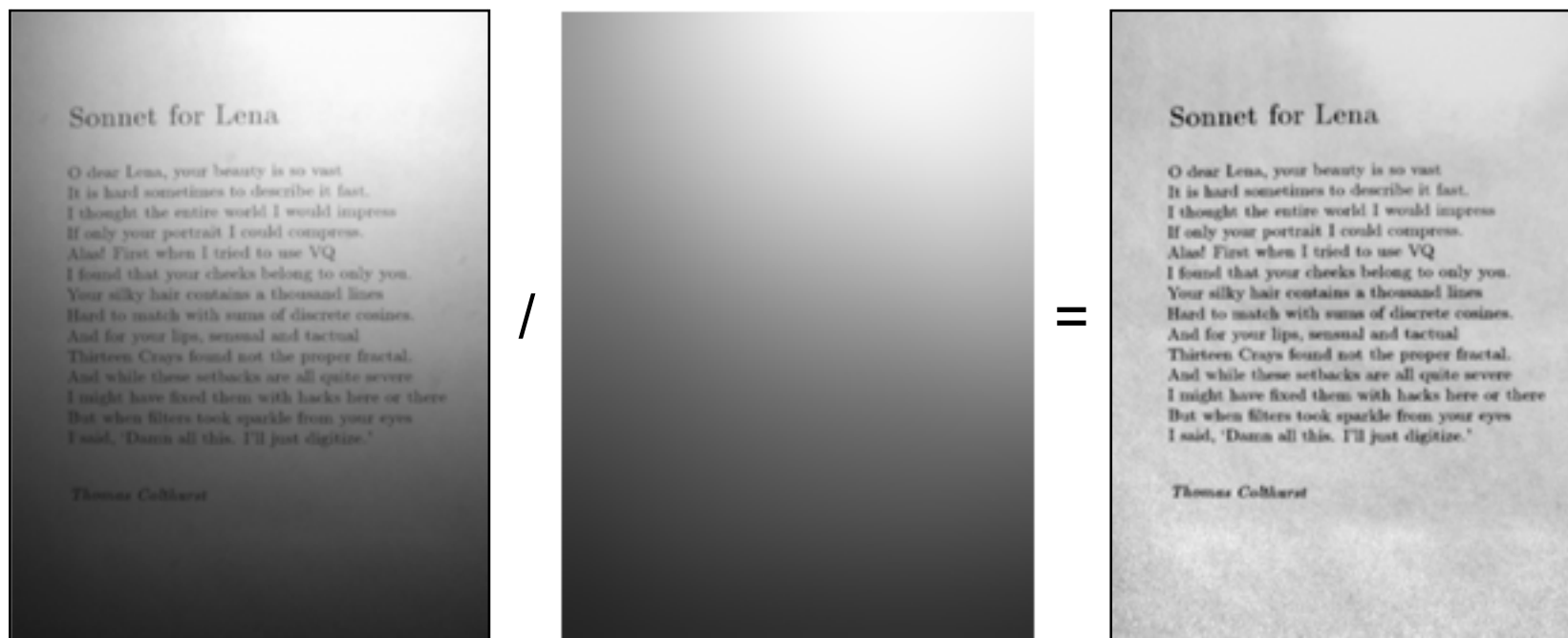
- После преобразования потребуется применить тональную коррекцию и определить значения, которые будут соответствовать черному и белому





Компенсация разности освещенности

Пример



Gauss 14.7 пикселей



Выделение краёв



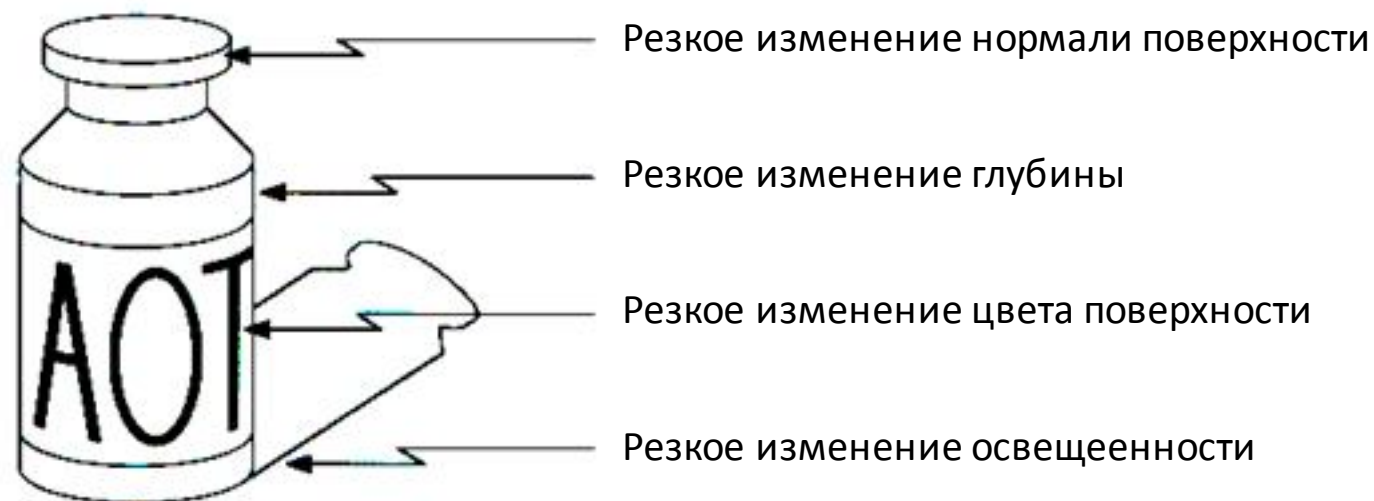
Выделение краев

- **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



Резкое изменение = «разрыв»

Существует множество причин формирования краев на изображении



Описание «края»

Край – это точка резкого изменения значений функции интенсивности изображения



Края соответствуют экстремумам производной



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

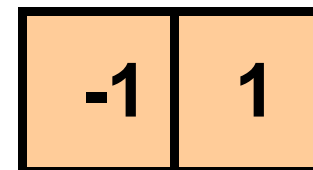
Для функции 2х
переменных, $f(x,y)$:

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

Разностная производная:

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

- Разностная производная - линейная и инвариантная к переносу
- Можно записать как свёртку



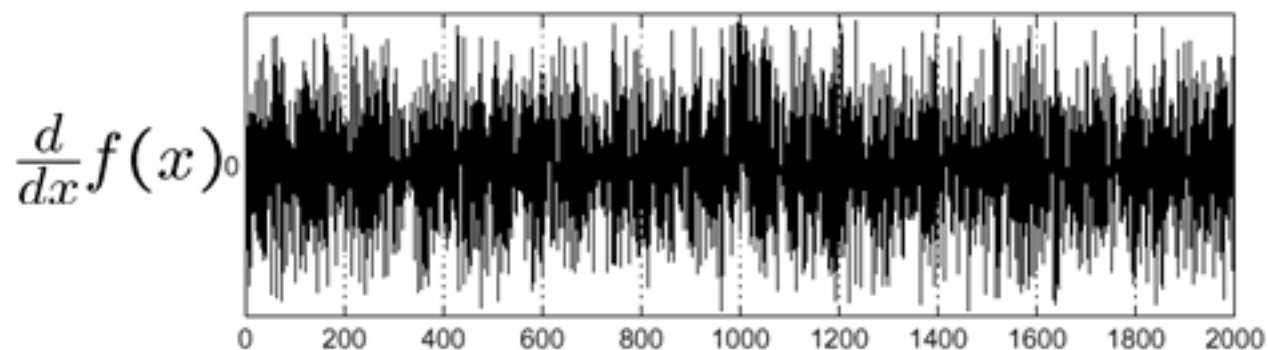
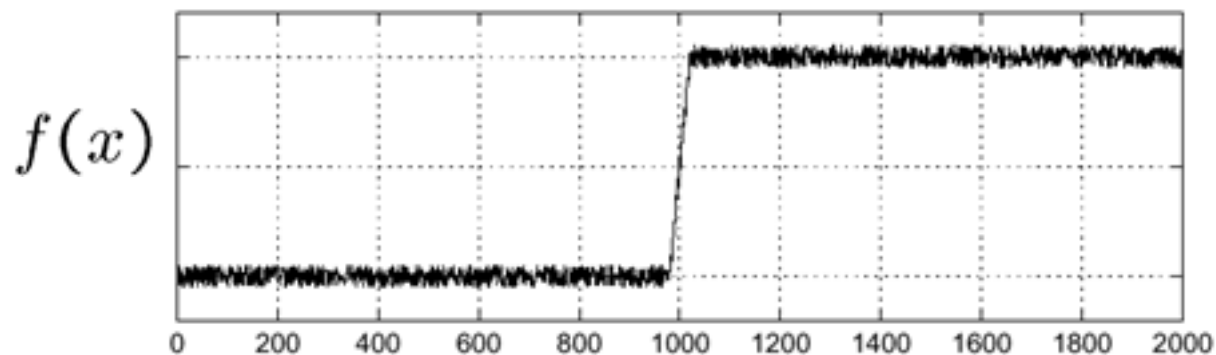
Простейший фильтр



Влияние шума

Рассмотрим строку или столбец изображения

- Интенсивность от положения можно рассматривать как сигнал



Край исчез

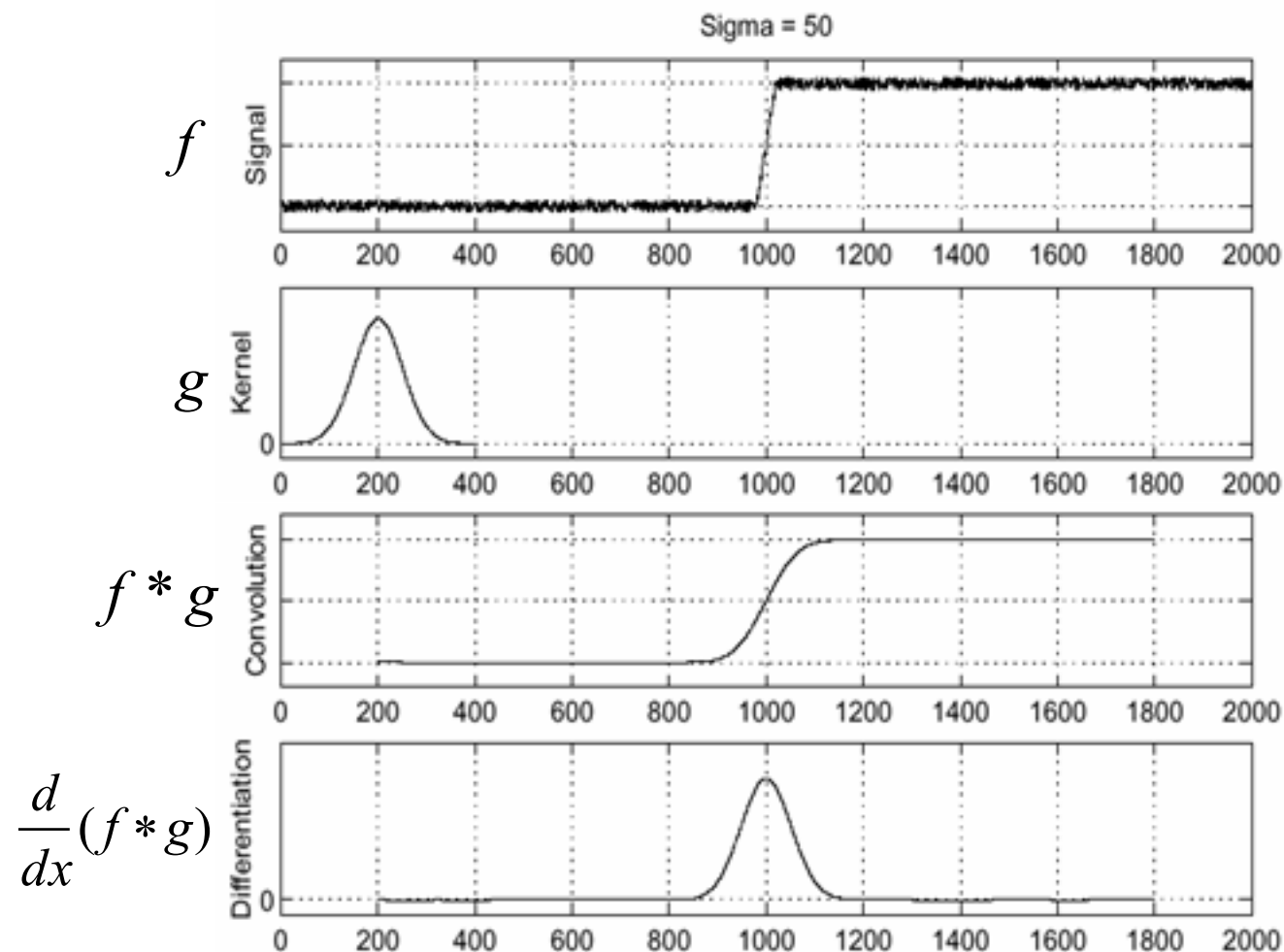


Влияние шума

- 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)

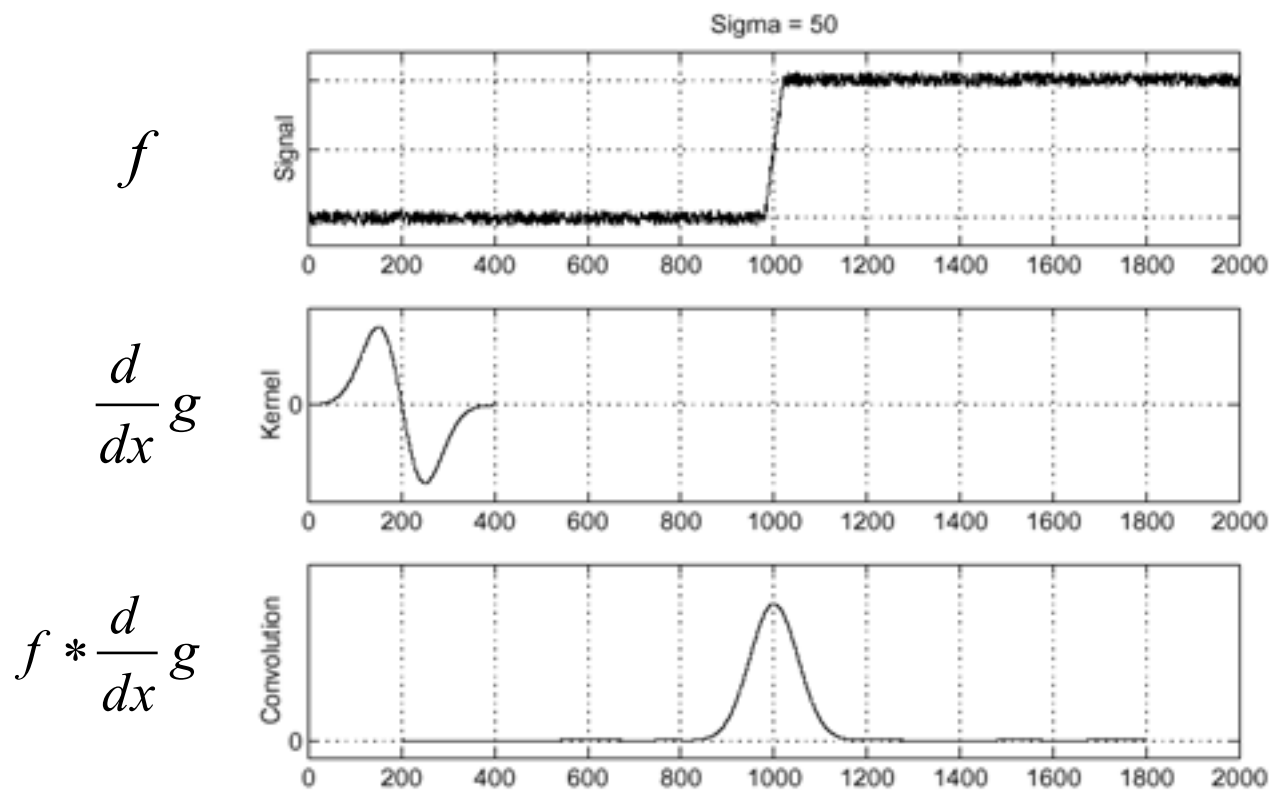


- Для поиска краев ищем пики в: $\frac{d}{dx}(f * g)$



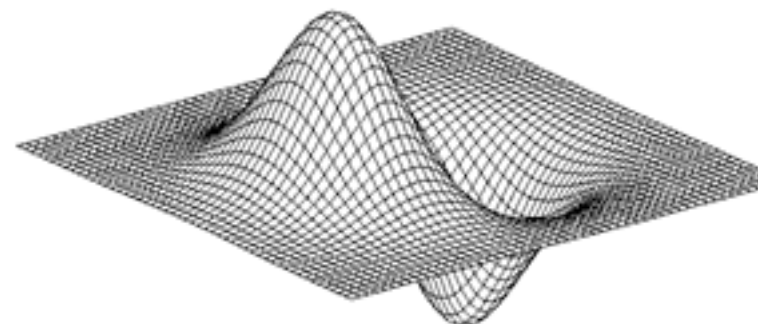
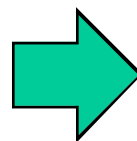
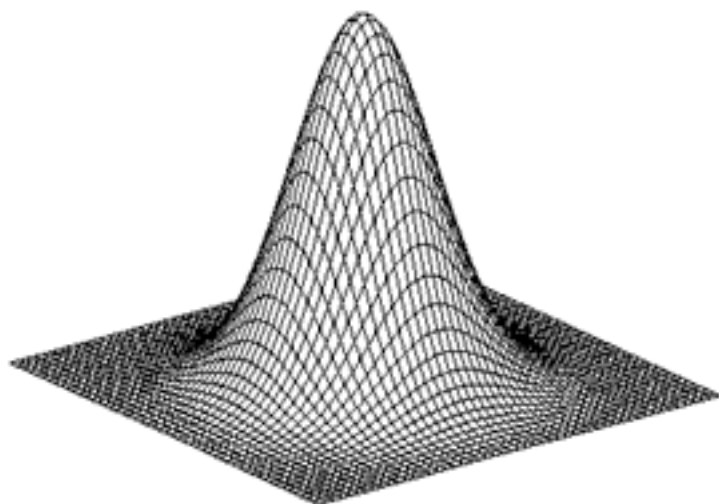
Свойства свертки

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

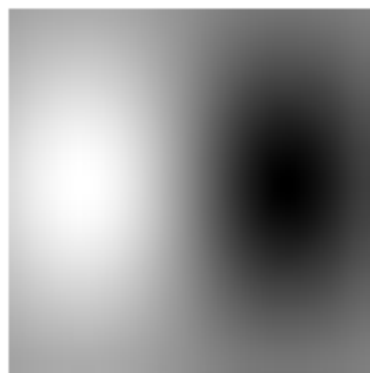
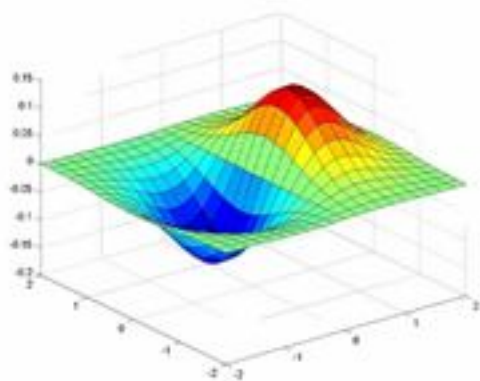




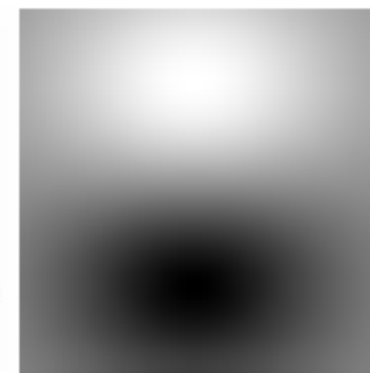
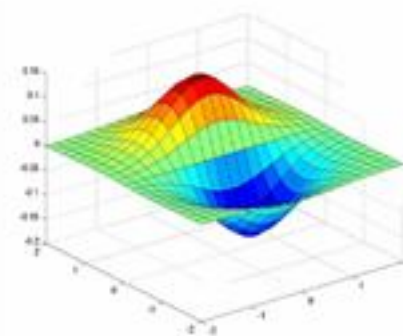
Производная фильтра Гаусса



По y



По y:





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

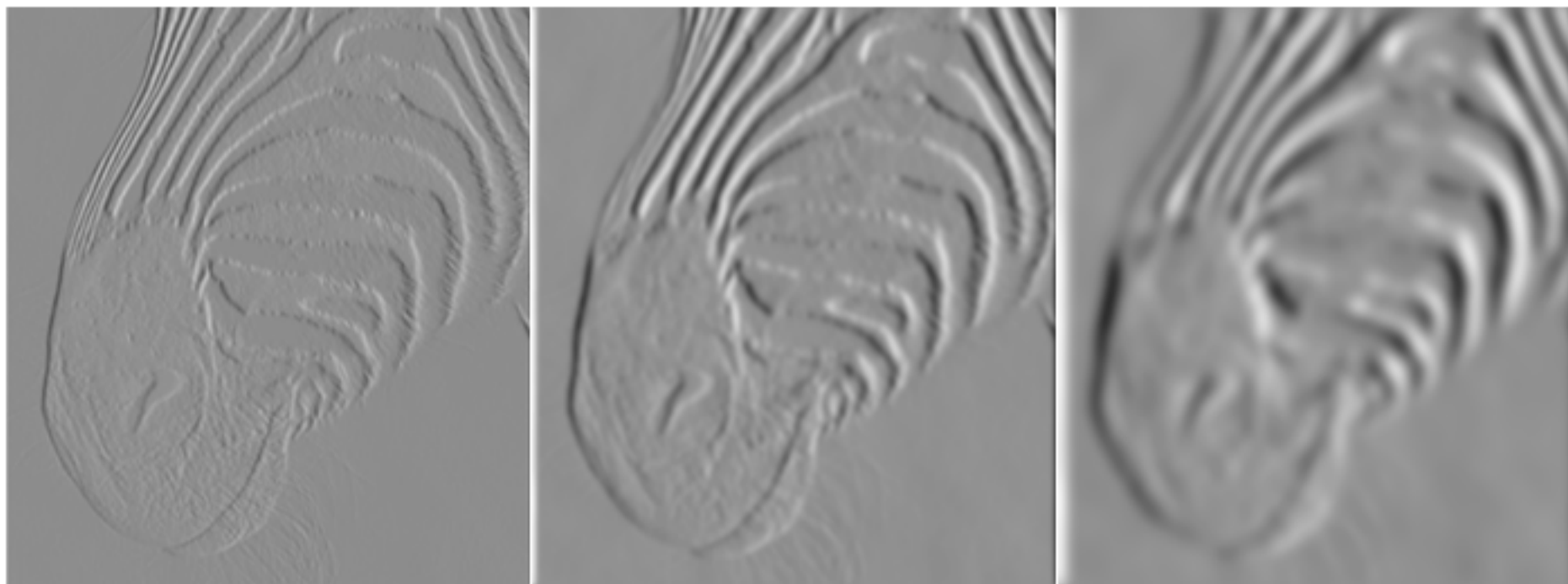


Собея



Сглаживание и локализация

Применим сглаженные производные разного размера:



1 pixel

3 pixels

7 pixels

Сглаженные производные подавляют шум, но размывают края. Плюс края находятся на разных «масштабах»



Выделение краев

- Вычисление градиента – не идеальный метод для поиска краёв.



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 силы и направления градиента
 3. Выделение локальных максимумов (Non-maximum suppression)
 - Утоньшение 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
- MATLAB: `edge(image, 'canny')`

J. Canny, [*A Computational Approach To Edge Detection*](#), IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

Source: D. Lowe, L. Fei-Fei



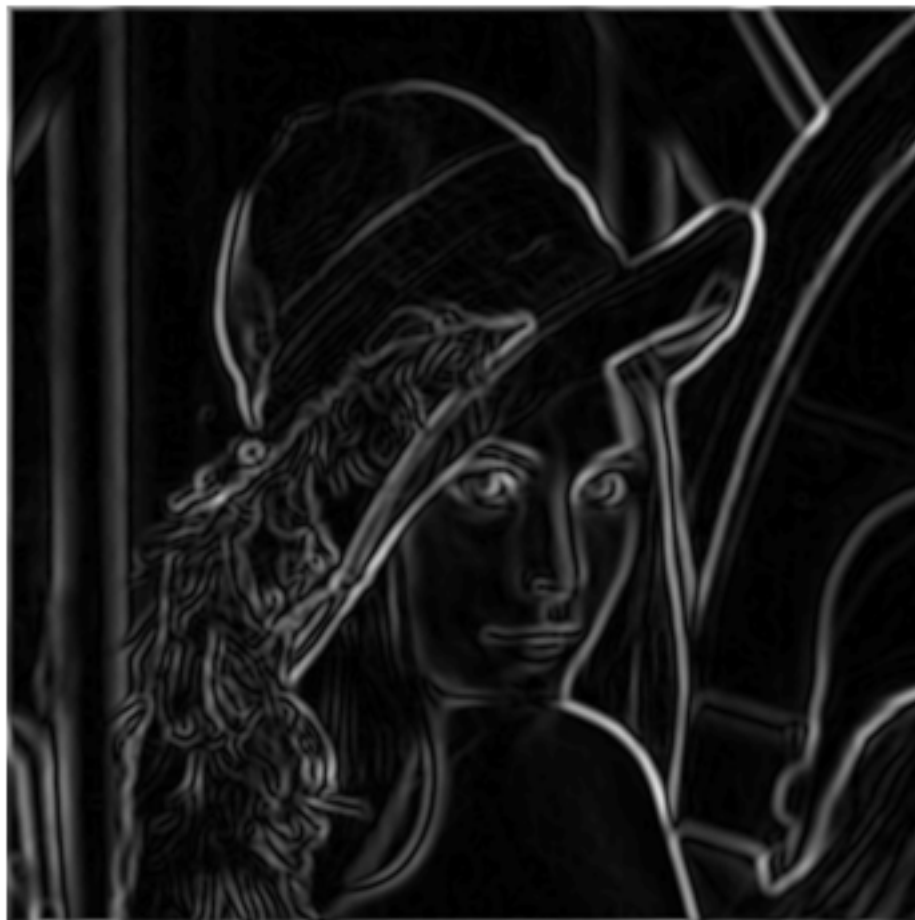
Пример



- Исходное изображение (Lena)



Пример



Норма градиента



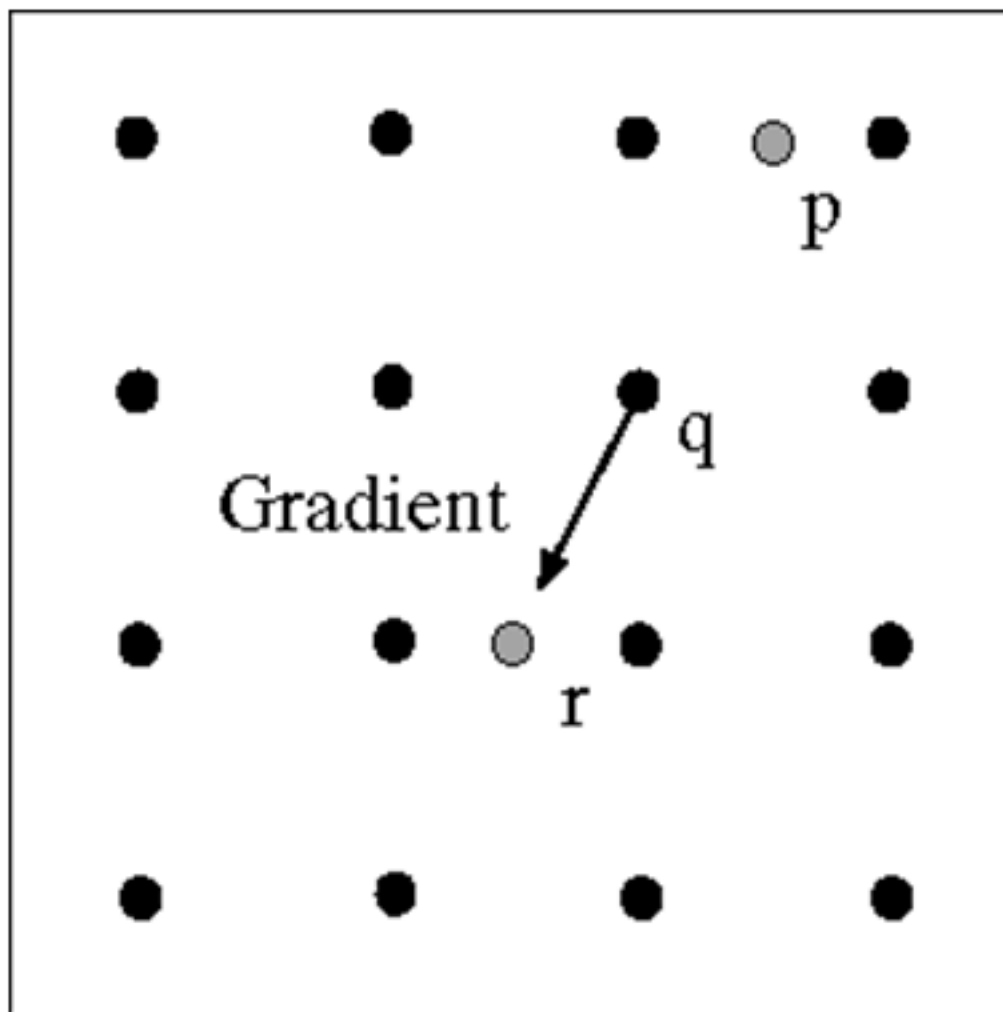
Пример



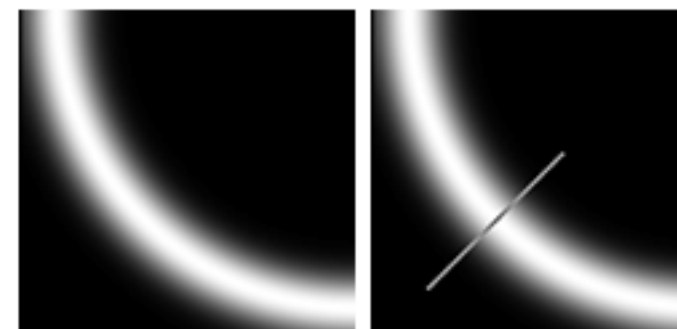
Отсечение по порогу



Поиск локальных максимумов

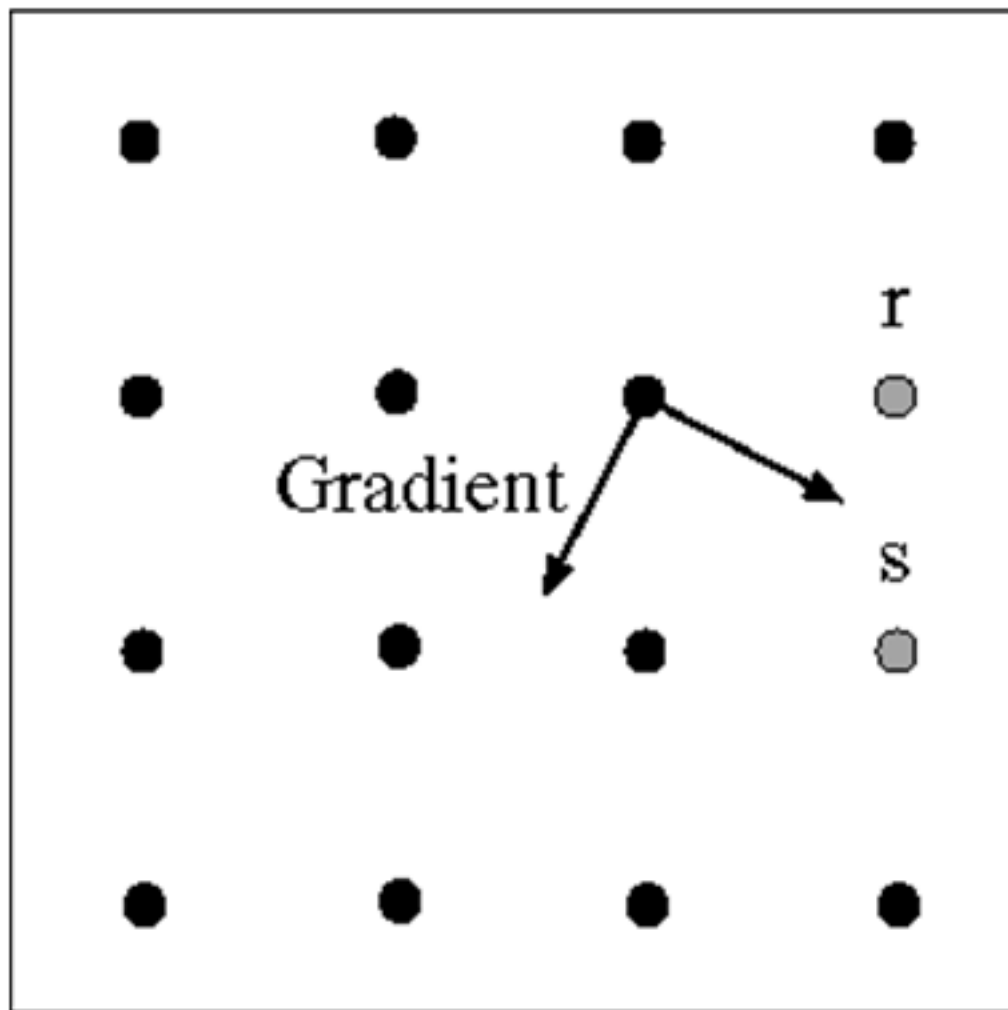


Максимум достигается в q , если значение больше p и r . Значения в p и r интерполируем.

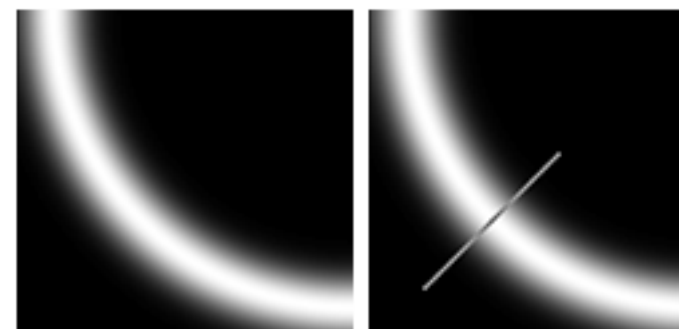




Связывание точек



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

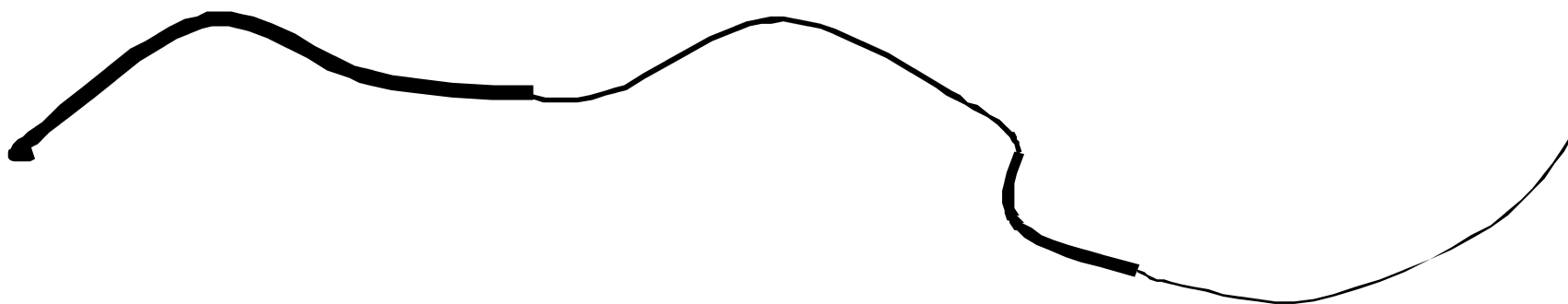


Утоньшение
(non-maximum suppression)



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



Влияние σ (Размер ядра размытия)



original

Canny with $\sigma = 1$

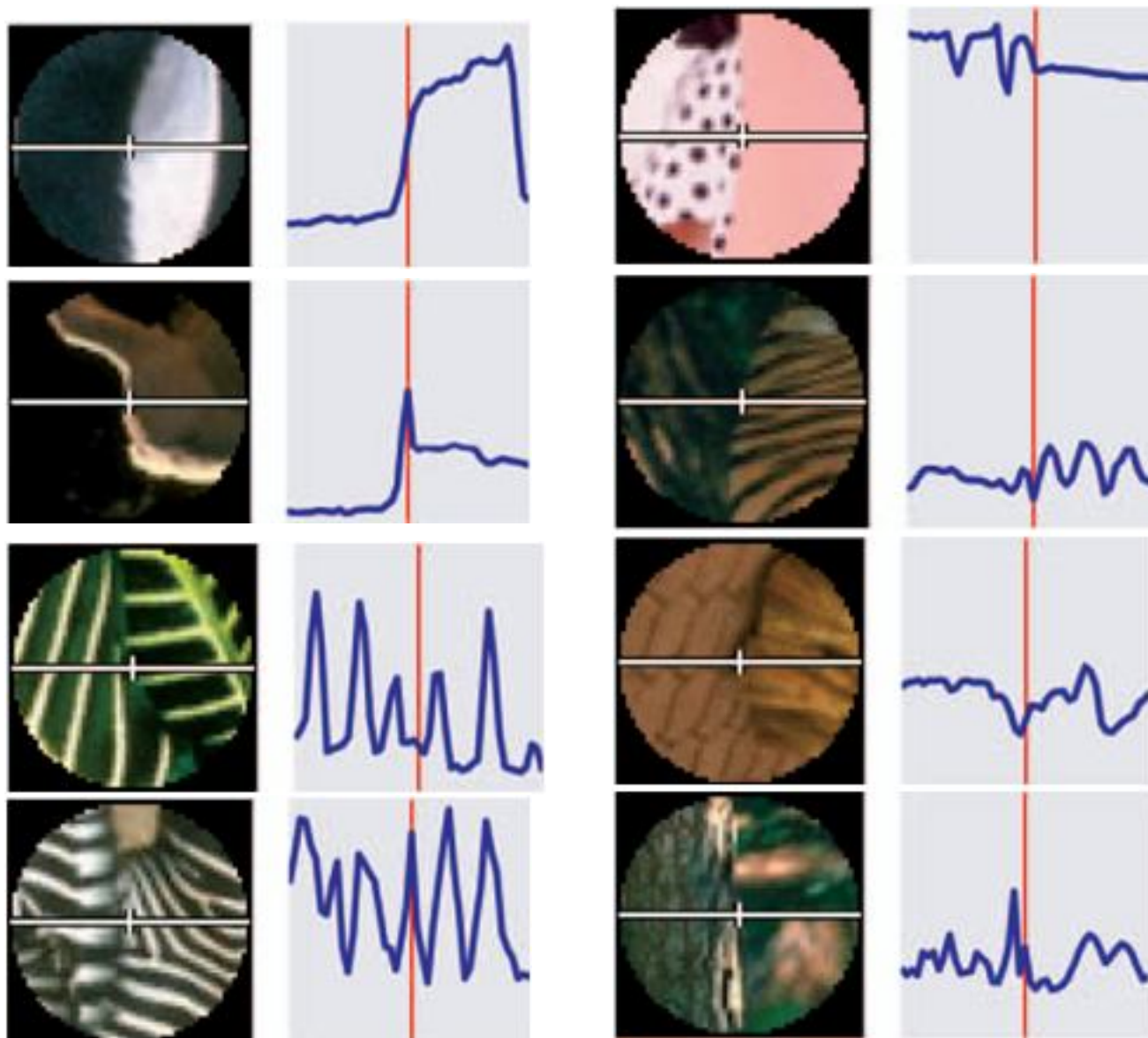
Canny with $\sigma = 2$

Выбор σ зависит от задачи

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



Ограничения детектора Canny



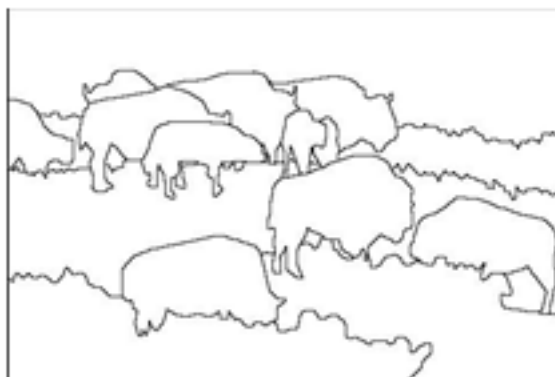


Поиск краев – это только начало...

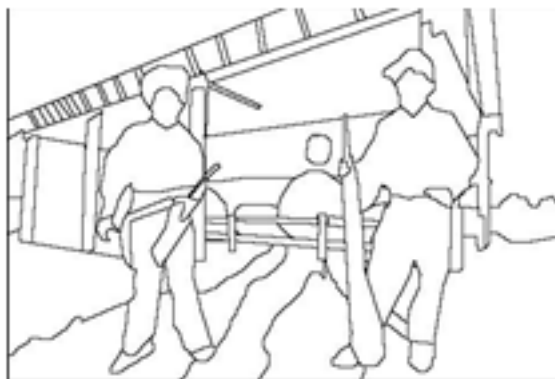
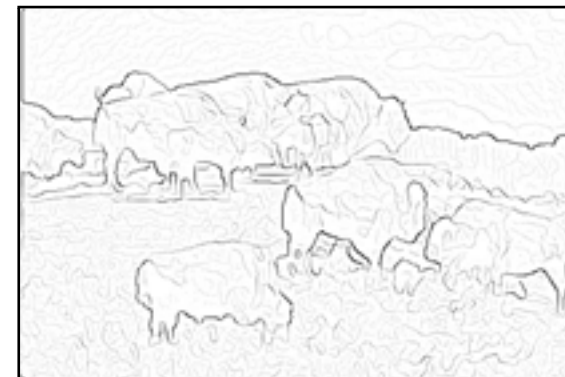
изображение



разметка вручную



норма градиента



- Berkeley segmentation database:
<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>



Спецэффекты

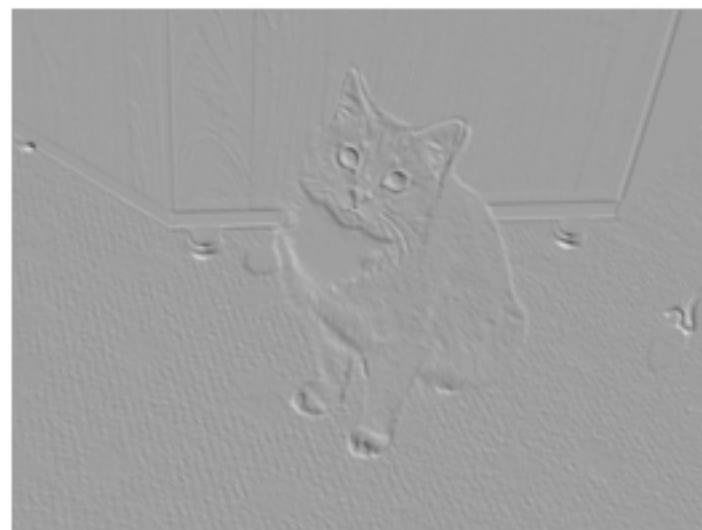
- Рассмотрим
 - Тиснение
 - Негатив
 - «Свелящиеся» края
 - Геометрические эффекты
 - Перенос/поворот
 - Искажение
 - «Эффект стекла»



Тиснение

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

Фильтр + сдвиг яркости, нормировка...





Цифровой негатив



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



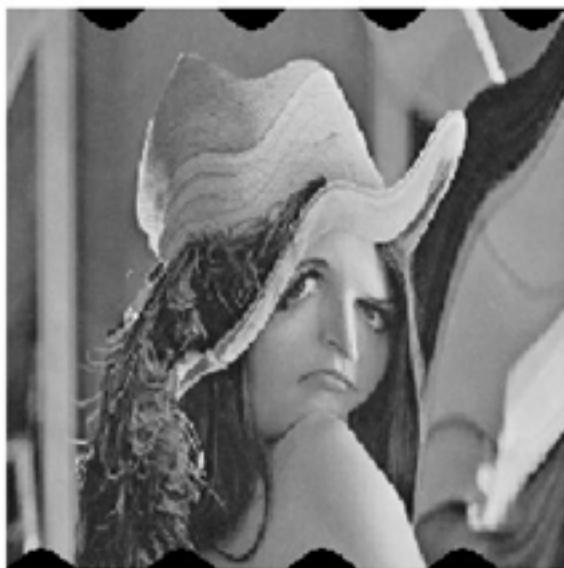
Светящиеся края



Медианный фильтр + выделение краев + фильтр «максимума»



«Волны»



Волны 1:

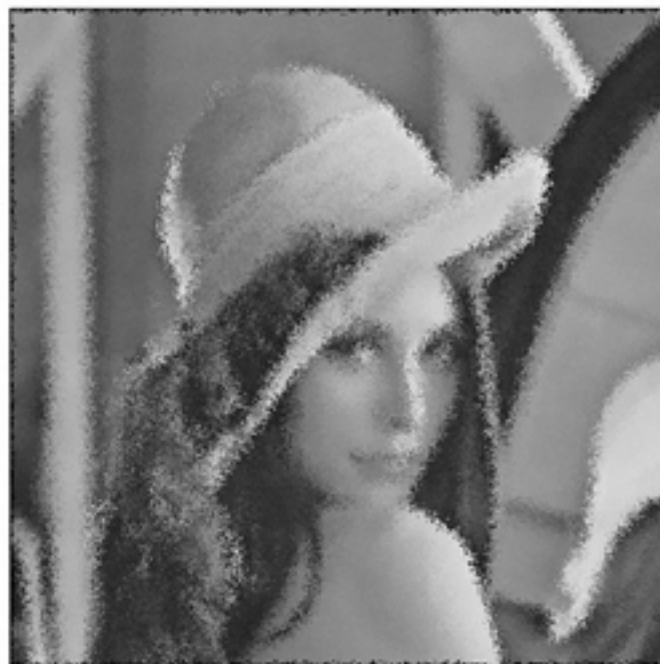
$$x(k; l) = k + 20\sin(2\pi l / 128); y(k; l) = l;$$

Волны 2:

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



«Эффект стекла»



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



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
- Выделение краёв изображения в простом случае можно достичь поиском локальных максимумов градиента яркости
- Всё это открытые задачи и сейчас активно продолжают исследоваться!