Лаборатория компьютерной
графики и мультимедиа
ВМК МГУ имени М.В. Ломоносова

Курс «Компьютерное зрение»

Лекция №2
«Основы обработки изображений»

Антон Конушин и Тимур Мамедов

2025 год

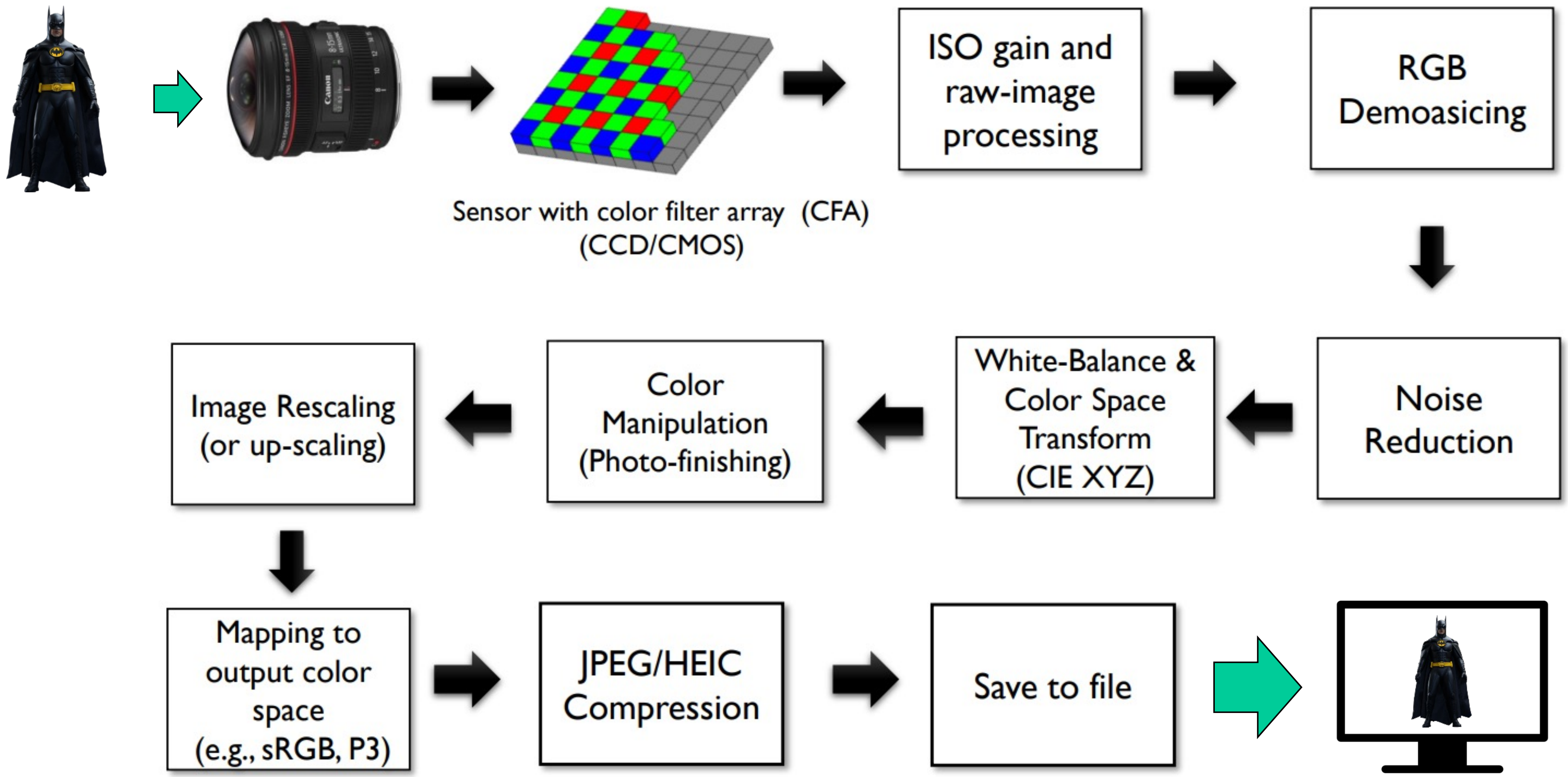
Обработка изображений

- Семейство методов и задач, где входной и выходной информацией являются изображения
- Формально $Y = f(X)$, где $X \in R^{n_x \times m_x \times k_x}, Y \in R^{n_y \times m_y \times k_y}$ - входное и выходное изображение

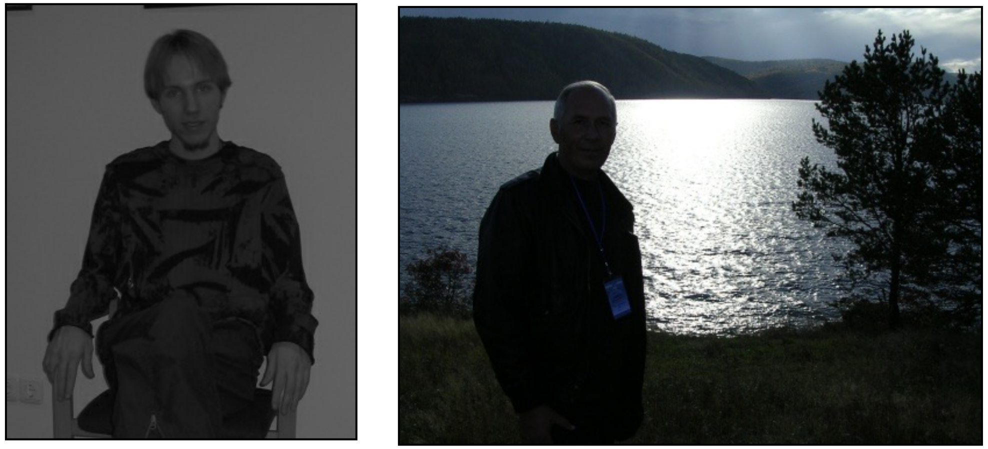
Цели:

1. Улучшение изображения для восприятия человеком
2. Улучшение изображения для восприятия компьютером
3. Извлечение признаков изображений для последующего анализа
4. Преобразование для технических нужд
5. Развлечение (спецэффекты)

Путь изображения в камере и дальше



Рассматриваемые темы



Повышение контраста

Цветокоррекция

Шумоподавление и фильтрация

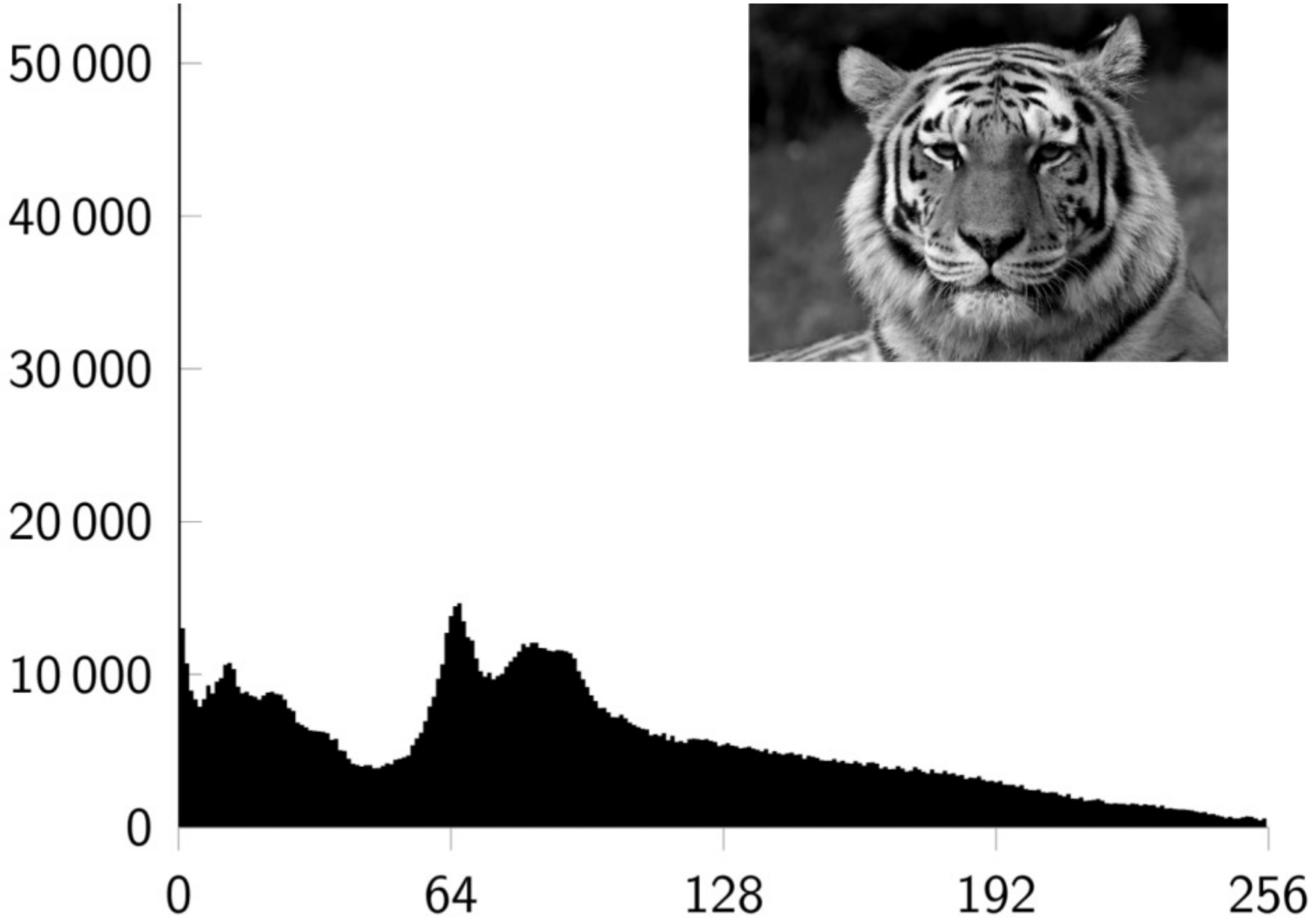
Выделение краёв

Тоновая (тональная) коррекция
или
Улучшение контраста

Задача тональной коррекции

- Повышение контрастности (распределения света и тени) в изображениях
  - Нужно оценить качество передачи тонов в изображении
  - Применить какую-то операцию преобразования яркостей пикселов

Гистограмма яркости

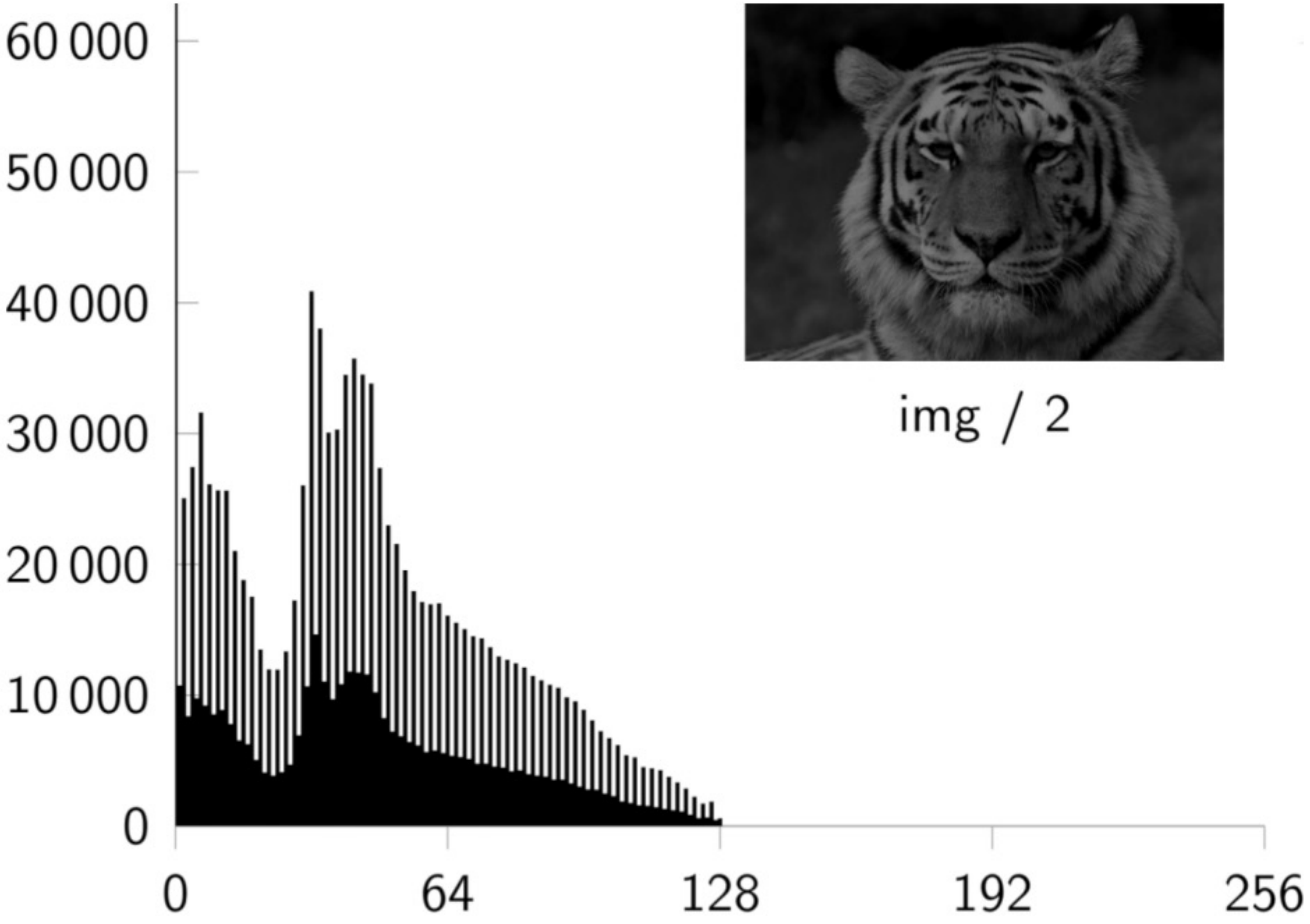


Гистограмма яркости – это график распределения яркостей на изображении.

По горизонтальной оси - шкала яркостей тонов от белого до черного.

По вертикальной оси - число пикселей заданной яркости.

Гистограмма яркости



img / 2

Теперь не полностью используется диапазон яркостей

Бывает и полностью, но с высокой концентрацией пикселей в узкой области диапазона

Точечные операторы

Оператор, который определяет значение выходного пиксела по значению только одного входного пиксела. Все пикселы обрабатываются независимо друг от друга.

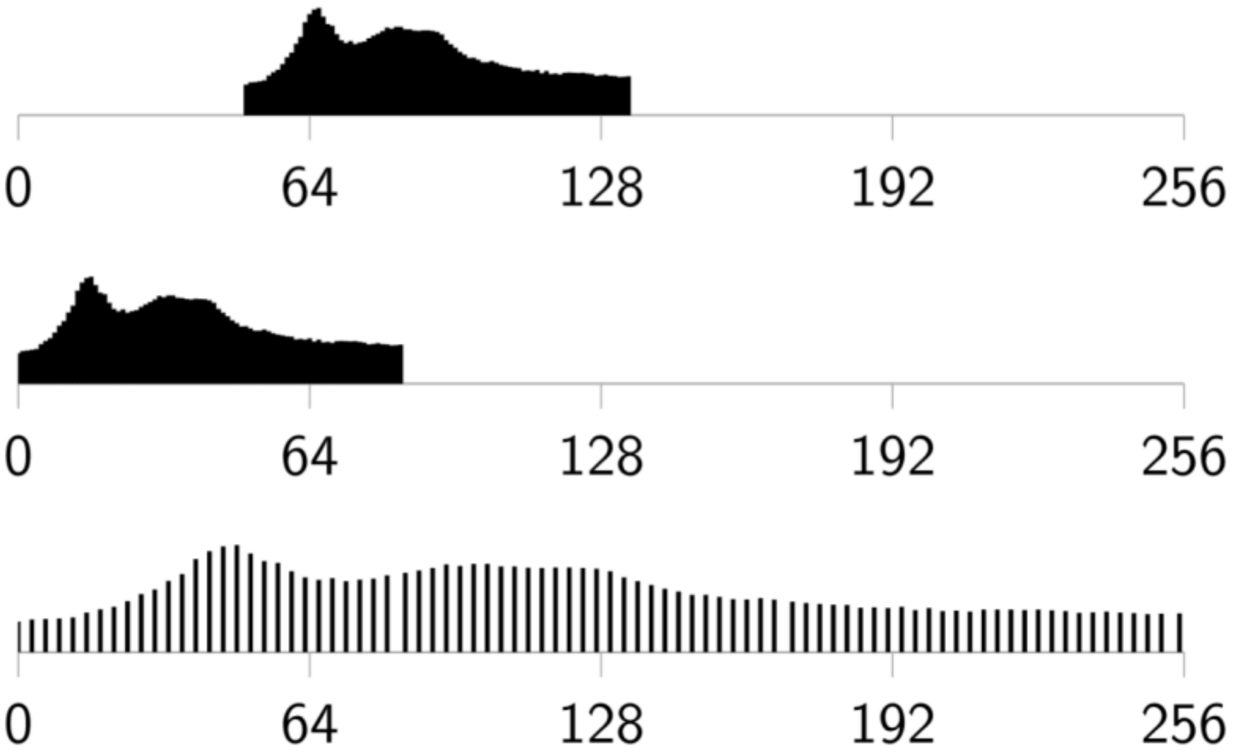
$$f^{-1}(y) = x$$

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

Пишем f^{-1}, потому что «восстанавливаем» истинное значение яркости по неправильному

Автоконтраст

Компенсация узкого диапазона яркостей – ***линейное растяжение гистограммы***:



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

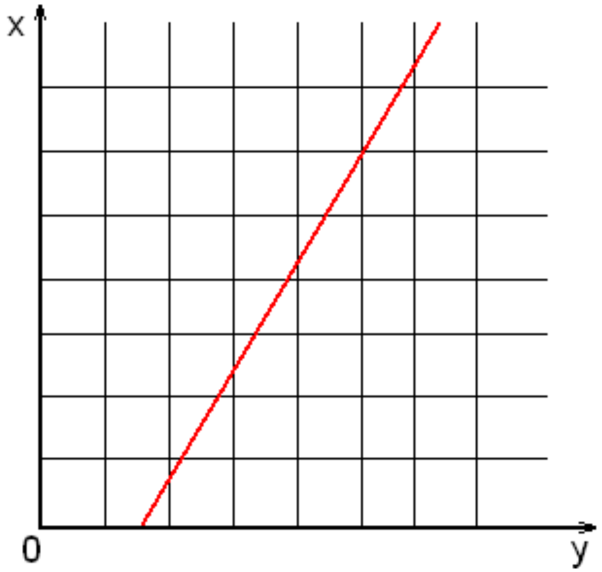
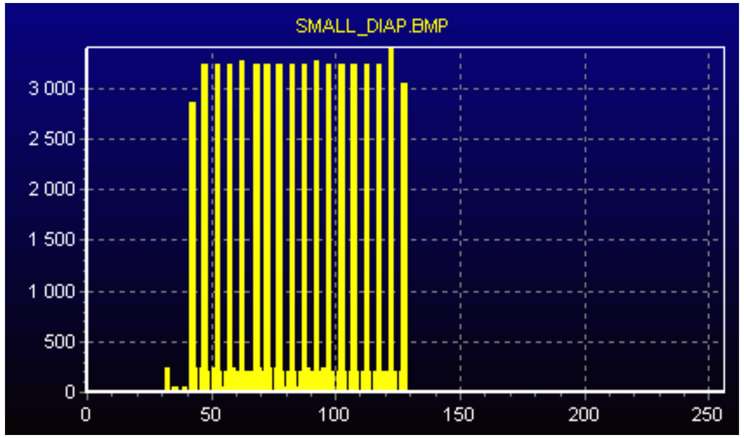
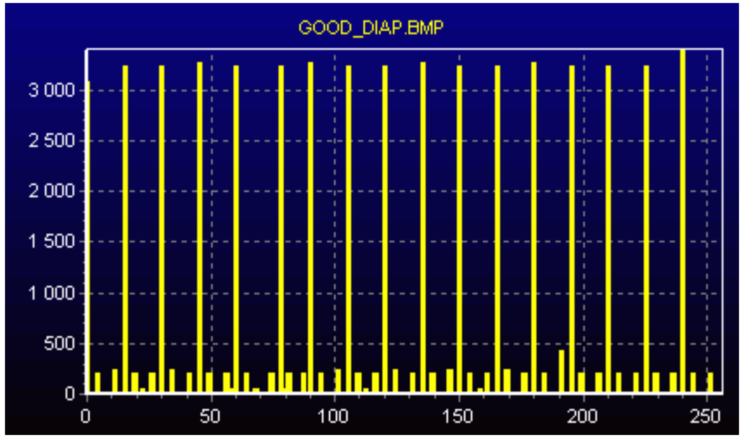


График функции $f^{-1}(y)$

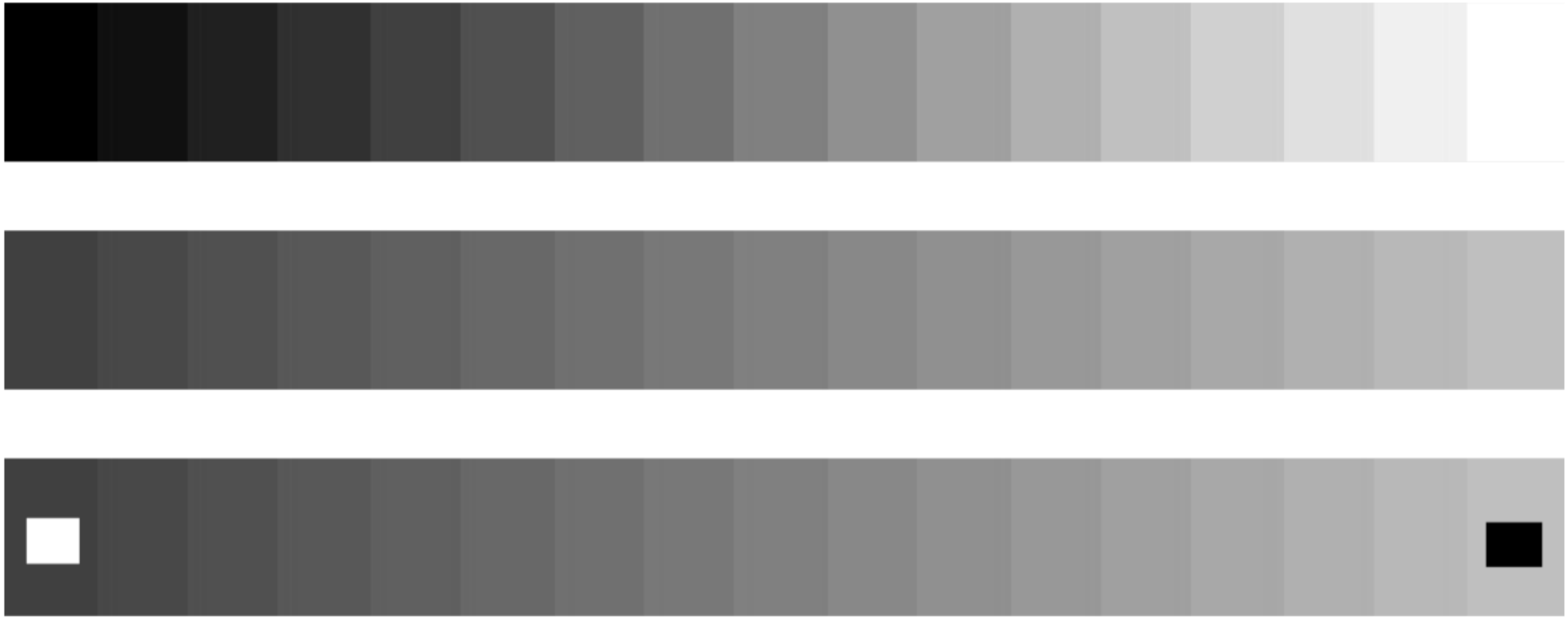
Линейная коррекция

Компенсация узкого диапазона яркостей – линейное растяжение гистограммы:





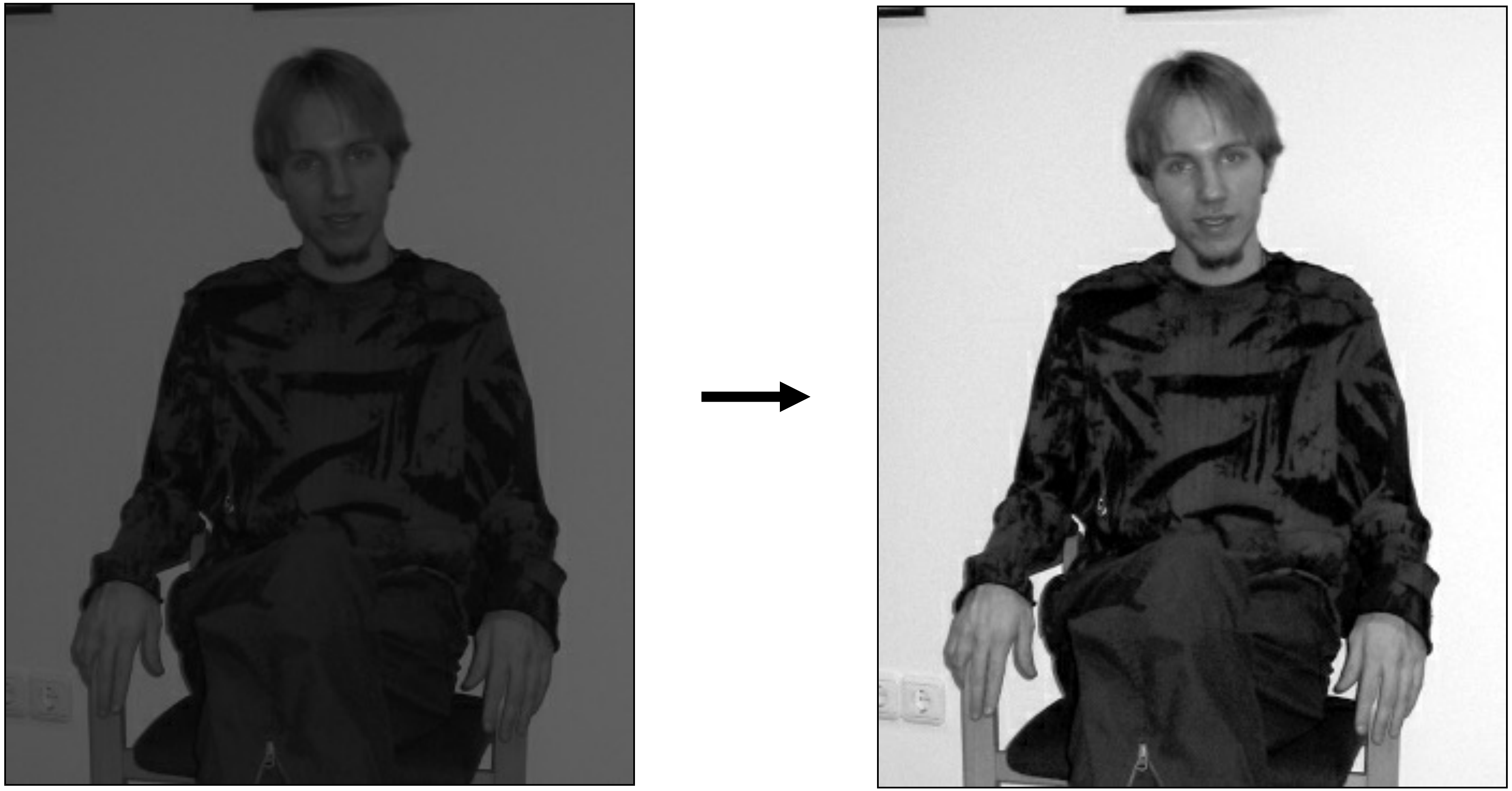
Робастная линейная коррекция



Робастная (устойчивая) версия метода:

- Вычислим такую линейную коррекцию, чтобы 1% самых темных пикселов стали черными и 1% самых светлых стали белыми

Линейная коррекция



Линейное растяжение – «как AutoContrast в Photoshop»

Линейная коррекция

Линейная коррекция помогает не всегда!

К слову, в чём может быть причина дефекта такого изображения?

Нелинейная коррекция

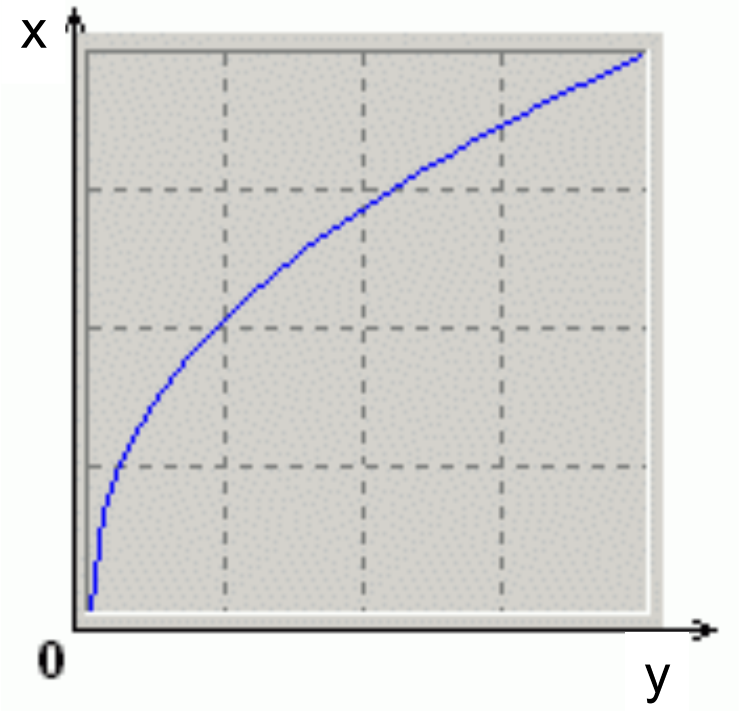
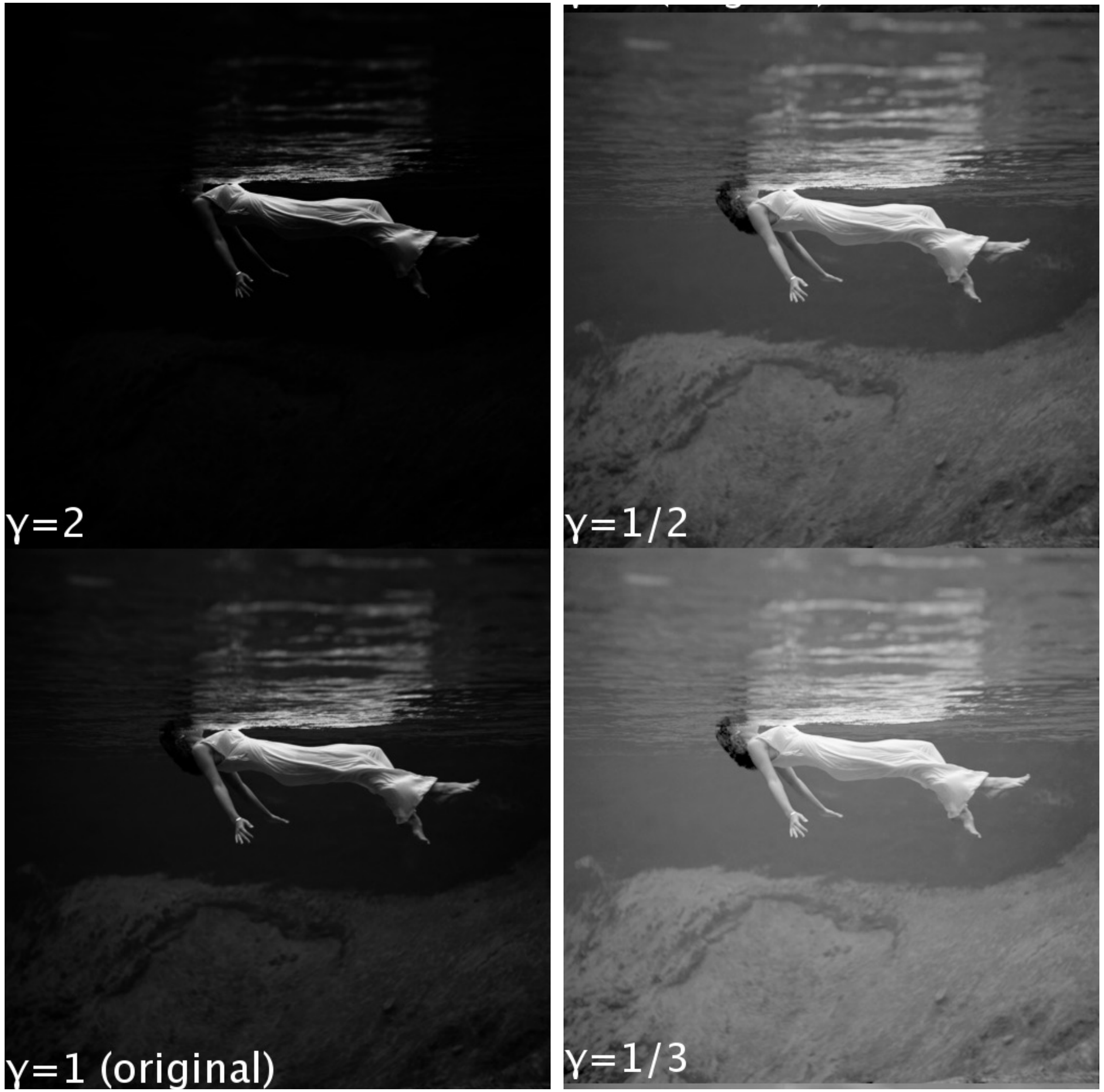
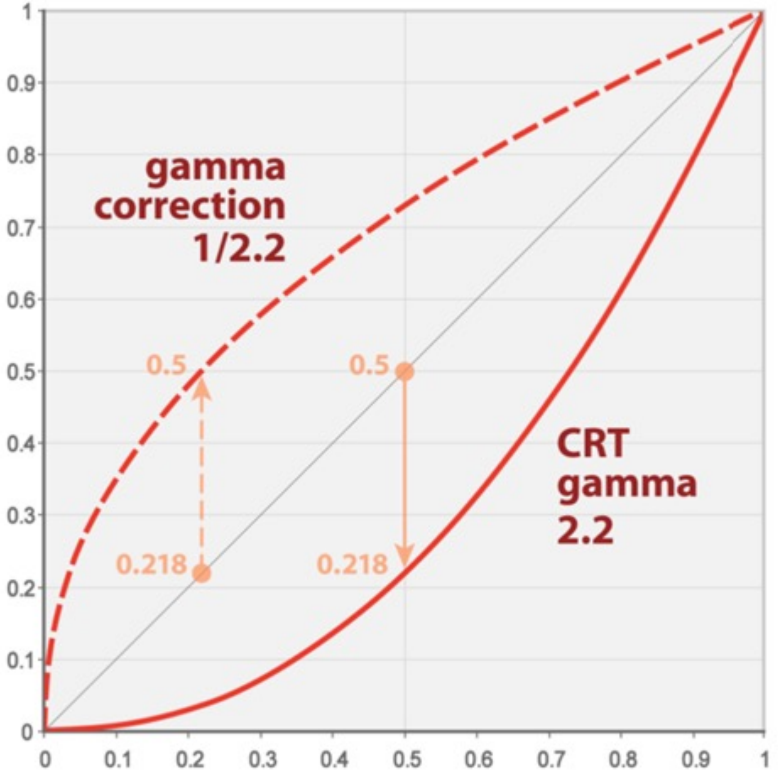


График функции $f^{-1}(y)$

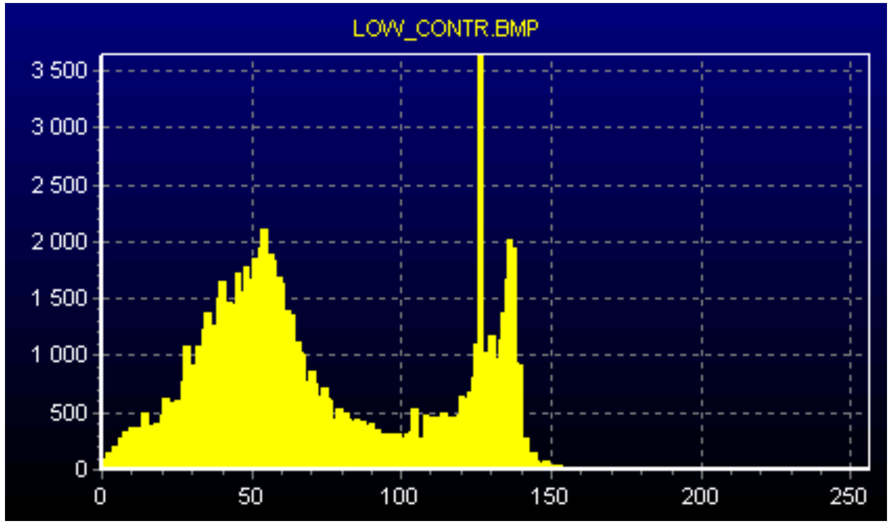
Гамма-коррекция

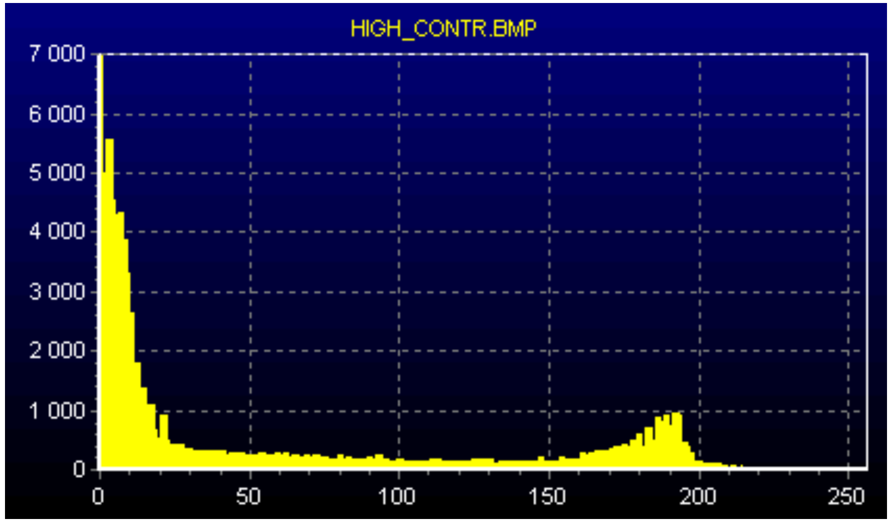




$$y = c \cdot x^{\gamma}$$

Произвольная нелинейная коррекция





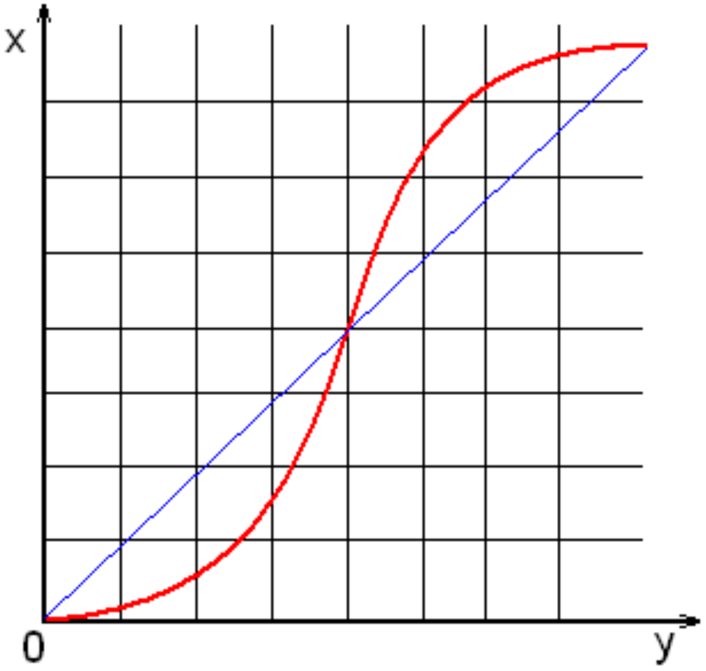
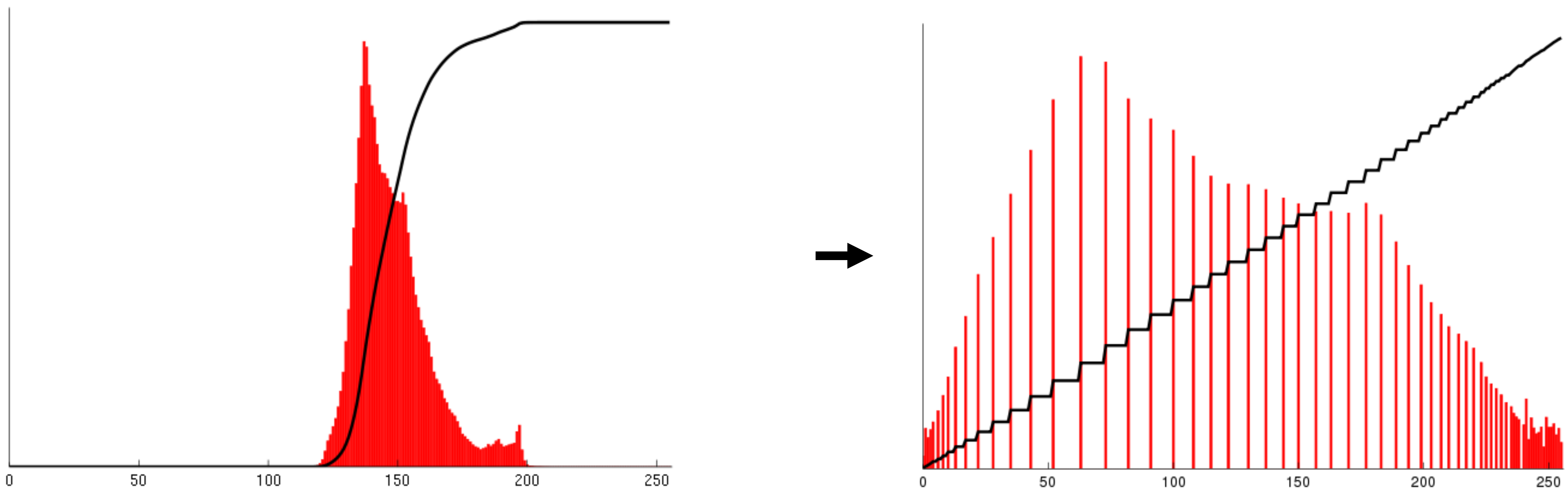


График функции $f^{-1}(y)$

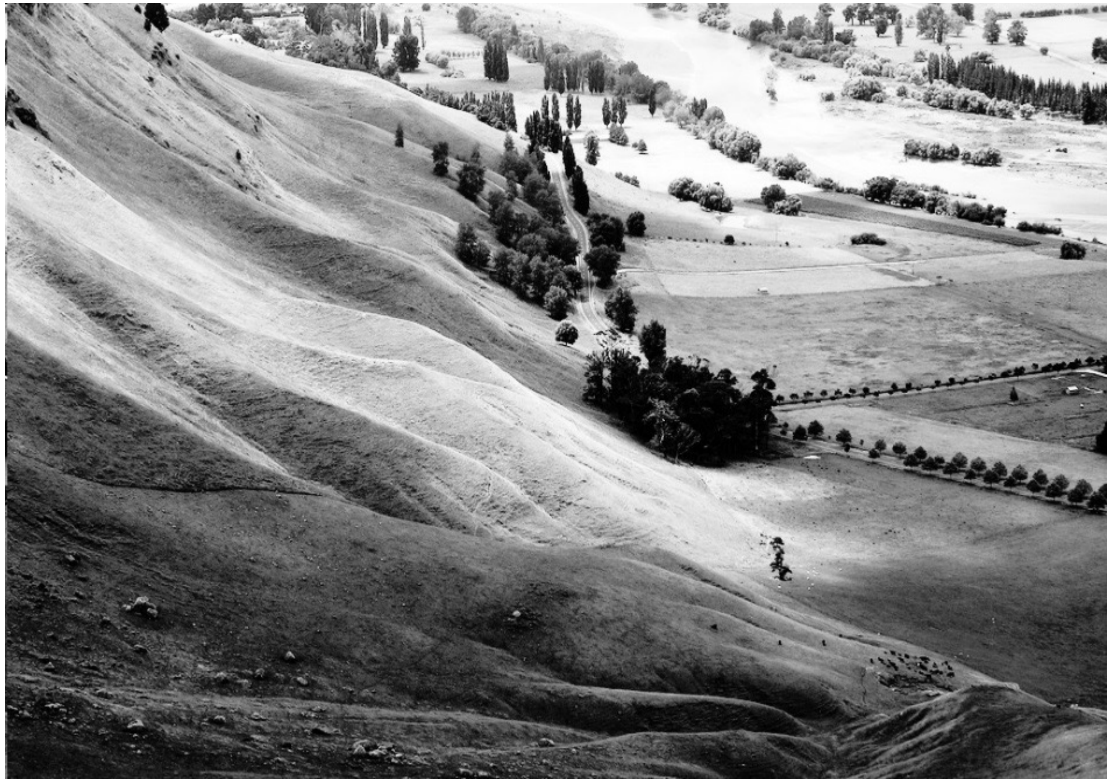
Выравнивание гистограммы



$$f(x) = \mathrm{round}\left(\frac{cdf(x) - cdf_{\mathrm{min}}}{\#pix - 1} \cdot 255\right)$$

https://en.wikipedia.org/wiki/Histogram_equalization

Выравнивание гистограммы



https://en.wikipedia.org/wiki/Histogram_equalization

Цветокоррекция

Цветокоррекция

Неправильный баланс	Правильный баланс
	

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

- Как определить, что цветопередача неправильная?
- 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

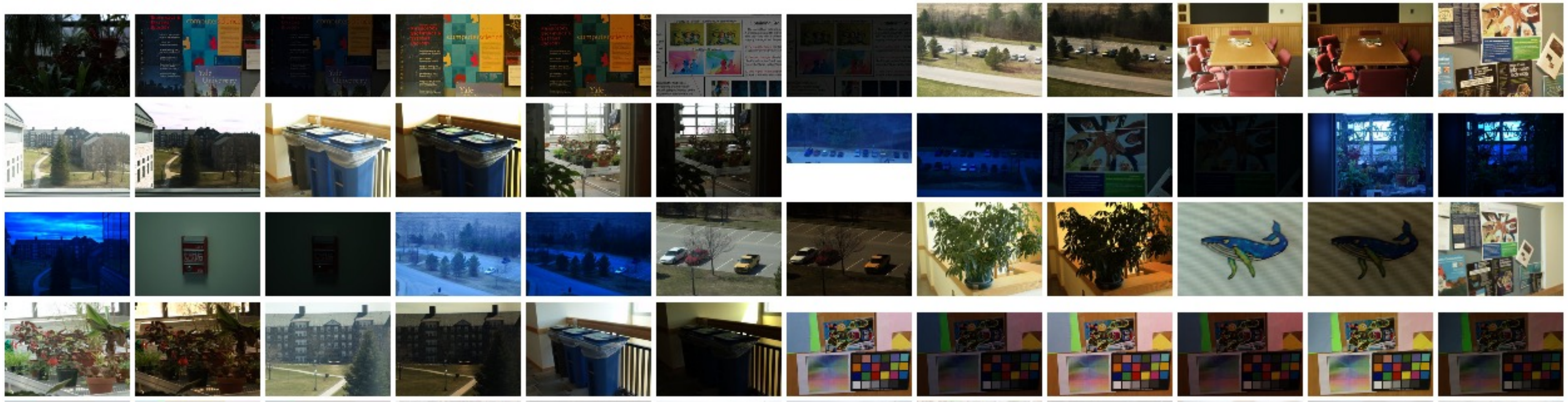
Source: The dark knight

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

Модель «Серого мира» (Grayworld)

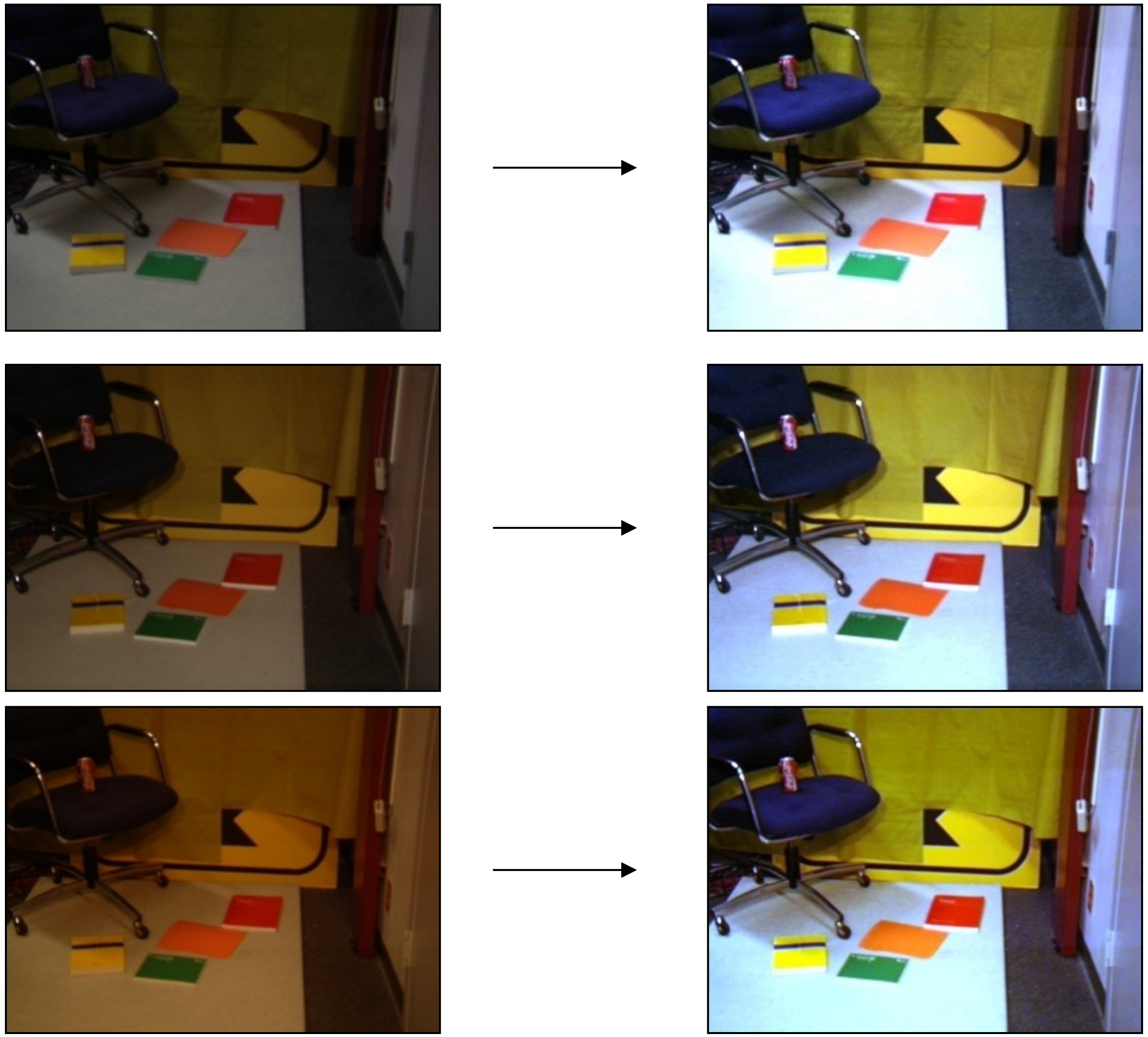
- Средний уровень («серый») по каждому каналу должен быть одинаков для всех каналов
- Если цветовой баланс нарушен, тогда «серый» в этом канале больше «серого» других каналов
- Вычислим коэффициенты усиления так, чтобы среднее в каждом канале стало одинаковым:

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

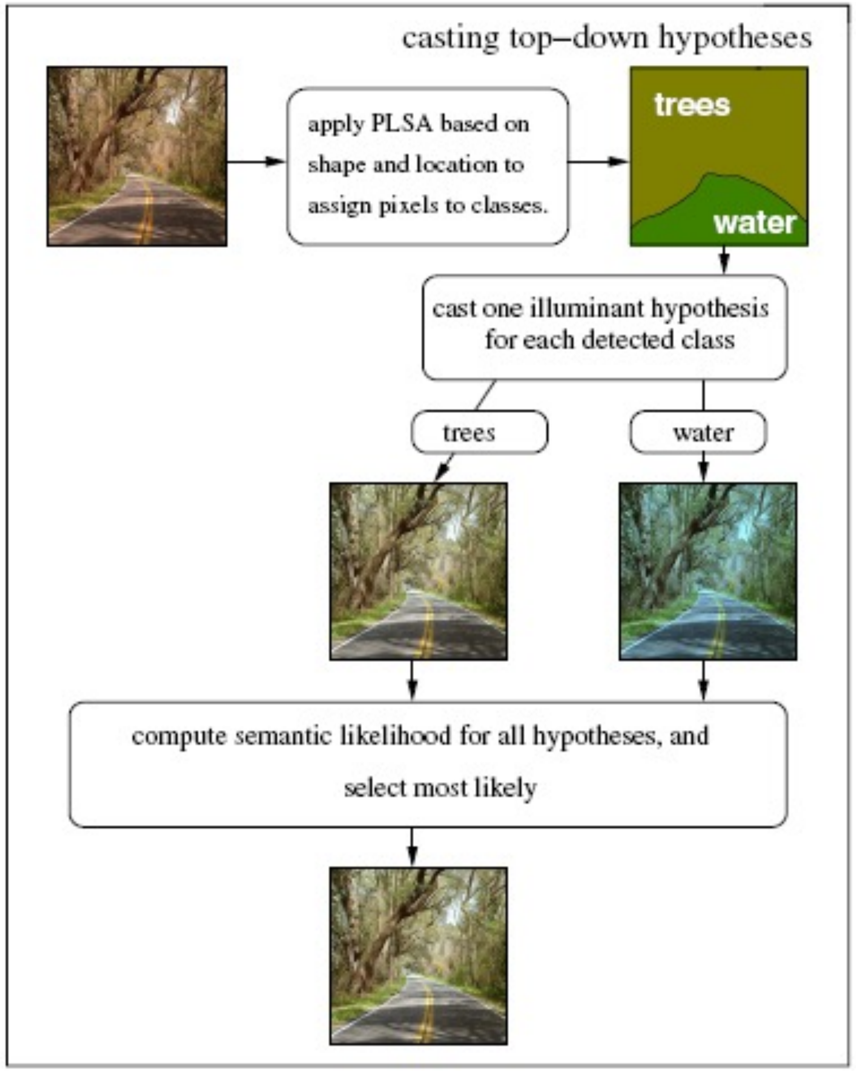
$$R' = R\cdot\frac{Avg}{\overline{R}}; \quad G' = G\cdot\frac{Avg}{\overline{G}}; \quad B' = B\cdot\frac{Avg}{\overline{B}}$$

Slide by S. Lazebnik

«Серый мир» - примеры



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

J. Van de Weijer, C. Schmid and J. Verbeek, Using High-Level Visual Information for Color Constancy, ICCV 2007.

Slide by S. Lazebnik

Шумоподавление и свёртка изображения

Шумоподавление

Шум фотоаппарата

- Изображения обычно содержат «шум», т.е. значение пикселя отличается от истинного
- Нужно получить «чистое» изображение без шума («подавить шум»)

Виды шума

Original

Gaussian noise

Salt and pepper noise

Impulse noise

- **Гауссов** (*Gauss noise*)
  - Аддитивный шум
  - $Image[i,j] = Image_{true}[i,j] + Noise[i,j]$
  - Колебания яркости, распределенные по нормальному закону
  - $Noise[i,j] \sim N(\mu, \sigma)$

- **Потеря информации (***data drop-out noise***)**
  - **Соль и перец**: случайные черные и белые пиксели
  - **Импульсный:** случайные белые пиксели

Source: S. Seitz

Усреднение нескольких кадров

«Временная фильтрация» - усреднение пикселей между несколькими кадрами

…

Зашумленные изображения

Усреднение по 10 изображениям

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

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

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

Какой вид шума можем подавить?

Шумоподавление в одном кадре

Как быть, если изображение только одно?

Можем усреднить пиксели в окрестности

Пространственная фильтрация

- Вычислим новое значение y_{ij} для пиксела x_{ij} как функцию от его локальной окрестности:

$$y_{ij} = f([x_{kl}]), x_{kl} \in neighbour(x_{ij})$$

- Например, взвешенное среднее по всем пикселам из окрестности
- Веса обозначаются как *ядро фильтра*
- Примером ядра для усреднения является “box” фильтр

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

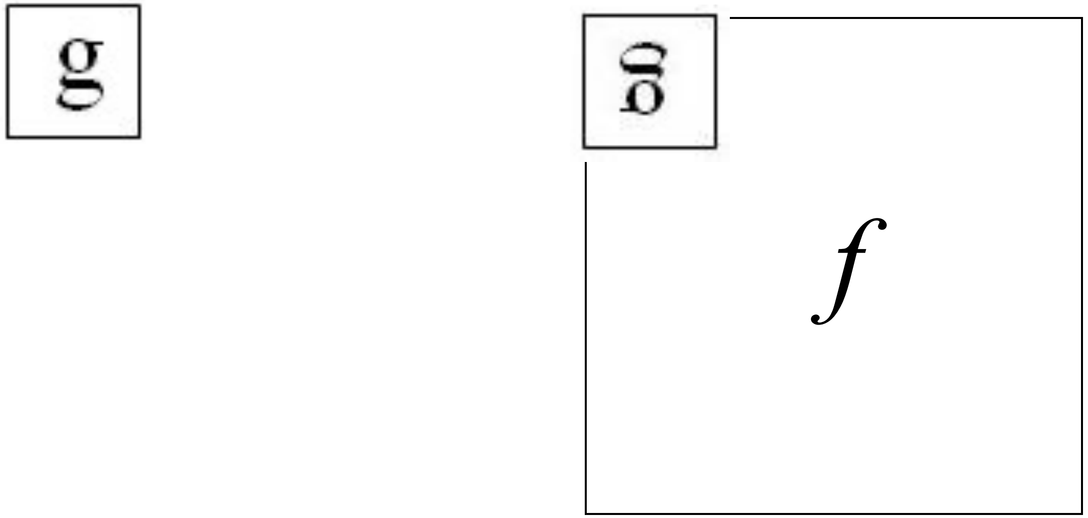
“box filter”

Source: D. Lowe

Свертка

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

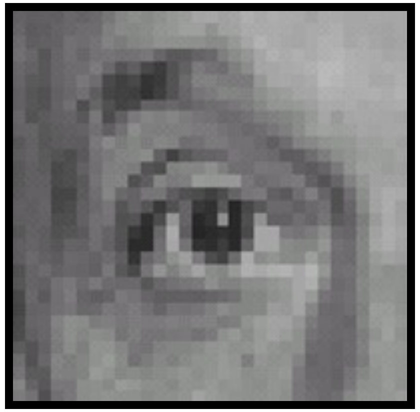
$$(f * g)[m,n] = \sum_{k,l} f[m-k, n-l]\, g[k,l]$$



Свёртка – ядро фильтра «перевёрнуто»!

Source: F. Durand

Примеры фильтров



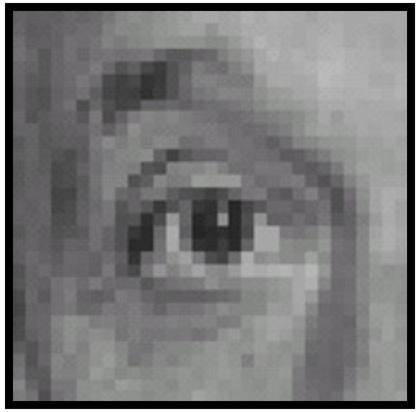
Исходное

0	0	0
0	1	0
0	0	0

?

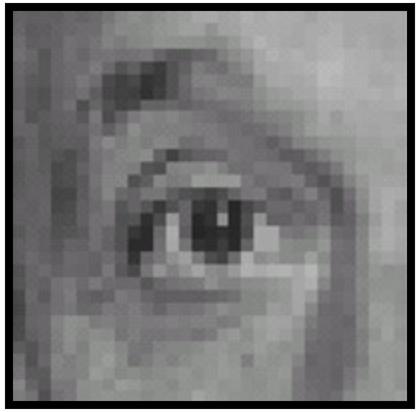
Source: D. Lowe

Примеры фильтров



Исходное

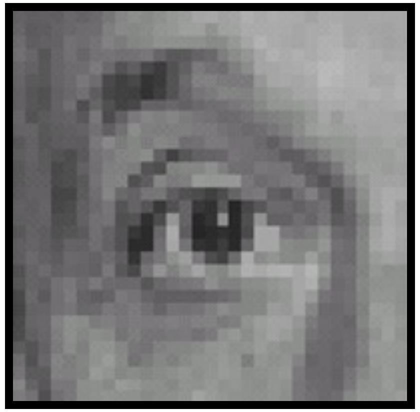
0	0	0
0	1	0
0	0	0



Результат

Source: D. Lowe

Примеры фильтров

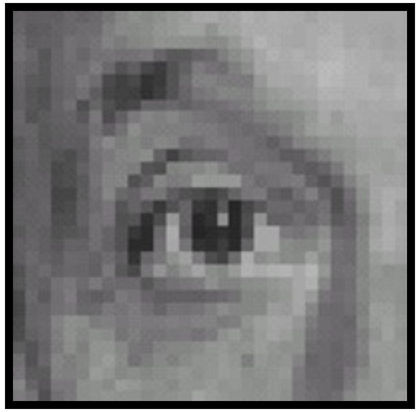


Исходное

0	0	0
0	0	1
0	0	0

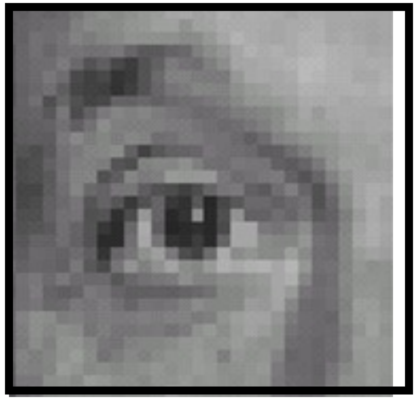
?

Примеры фильтров



Исходное

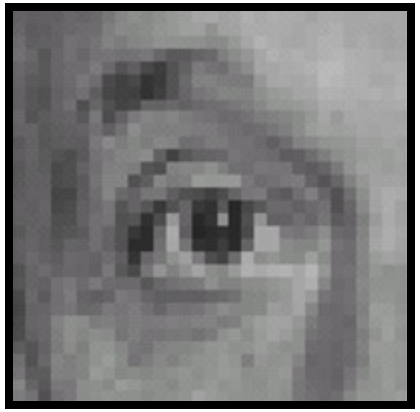
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$



Сдвиг на 1 пиксель

Source: D. Lowe

Примеры фильтров



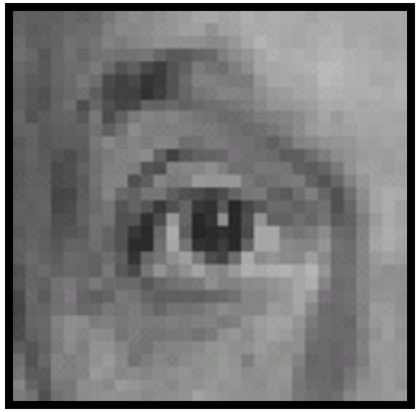
Исходное

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

?

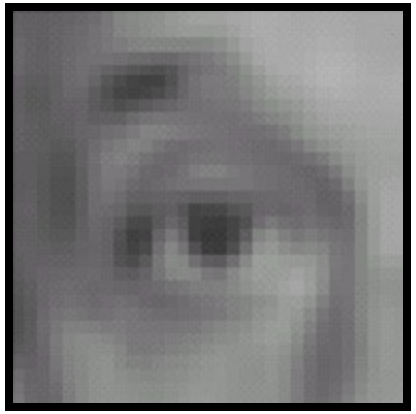
Source: D. Lowe

Примеры фильтров



Исходное

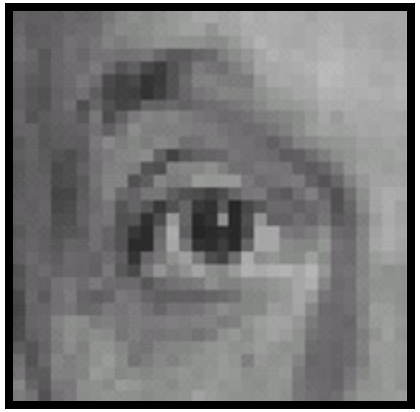
$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$



Результат

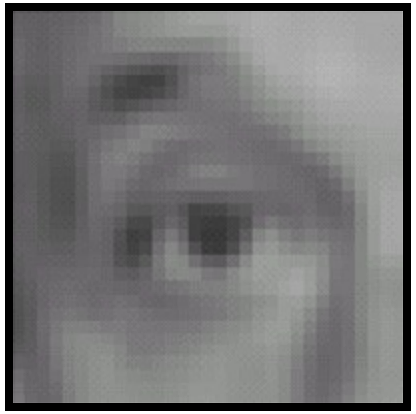
Source: D. Lowe

Примеры фильтров



Исходное

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



Результат

Любой фильтр со всеми положительными весами и суммой весов = 1 будет «сглаживать» (blur) изображение, при этом подавляя шум

Source: D. Lowe

Примеры фильтров

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

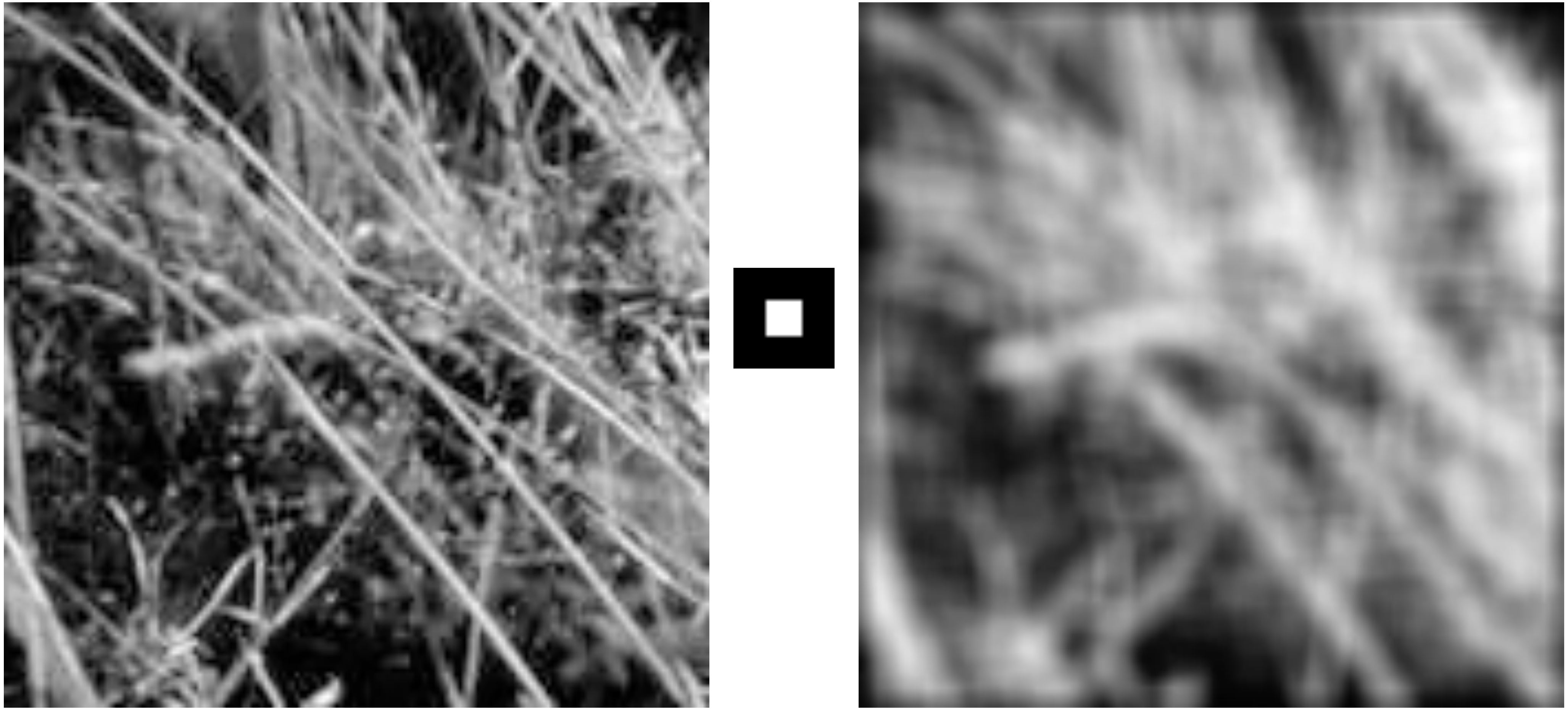
$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Примеры фильтров

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

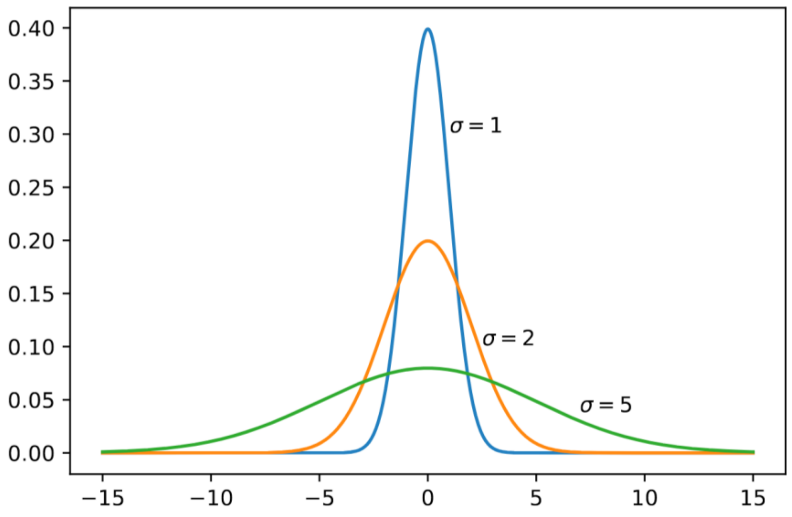
Вернёмся к шумоподавлению

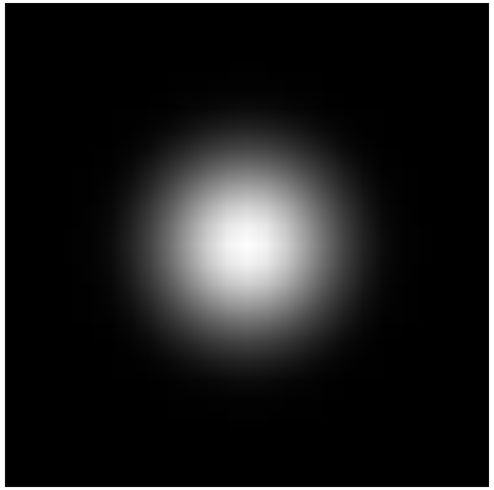
При сглаживании с box-фильтром на изображении могут образовываться паразитные линии



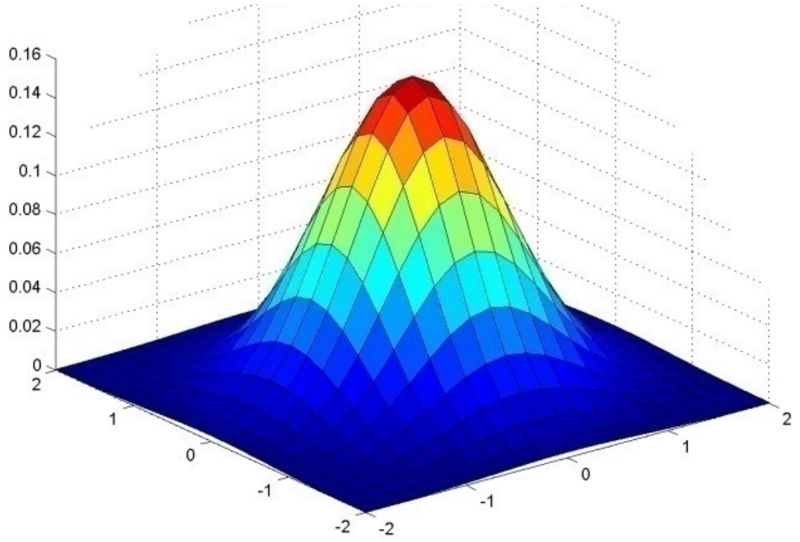
Source: D. Forsyth

Фильтр Гаусса





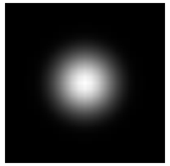
$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

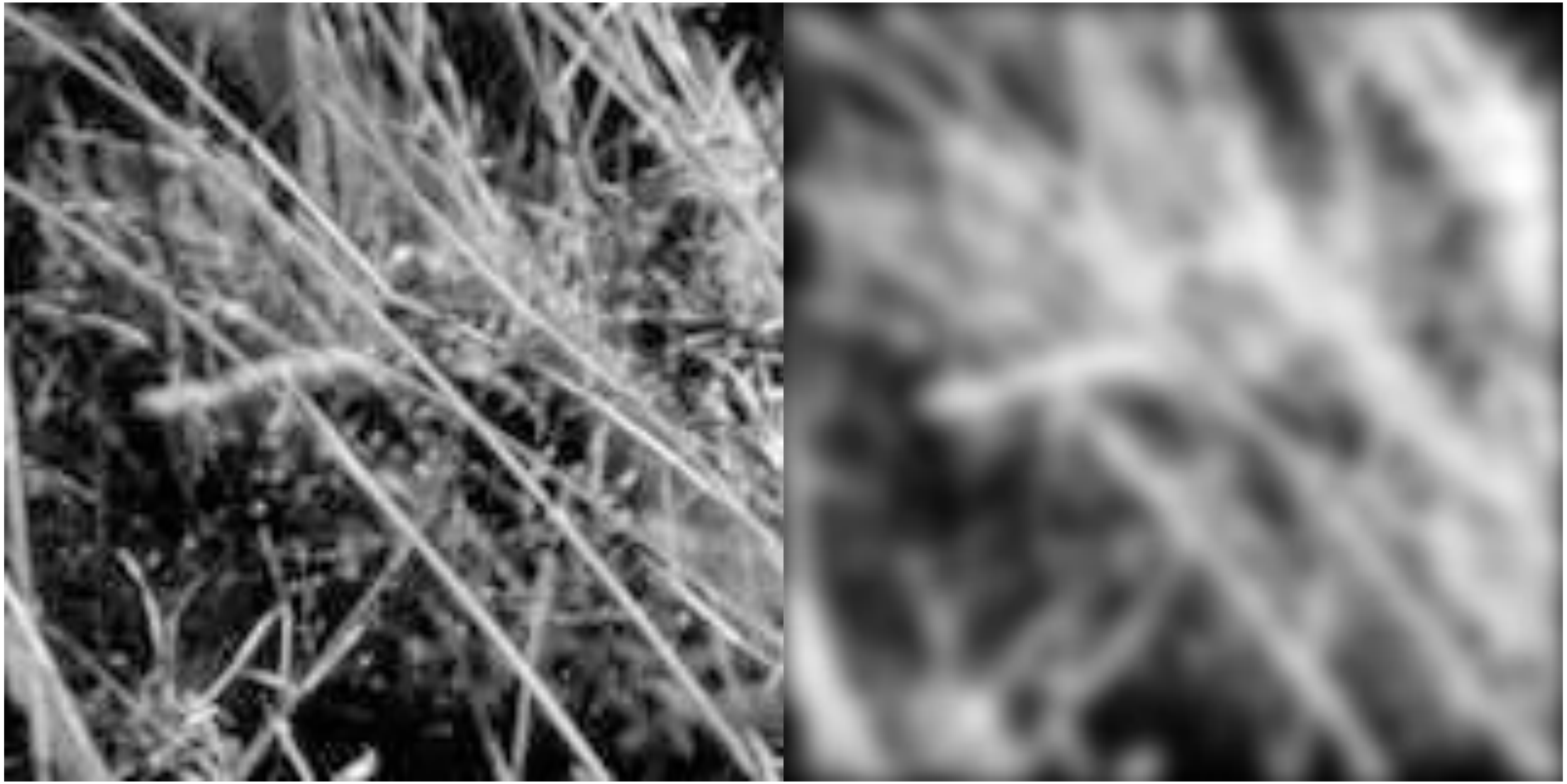


- Радиус r фильтра равен 3σ
- Размер фильтра 2r+1

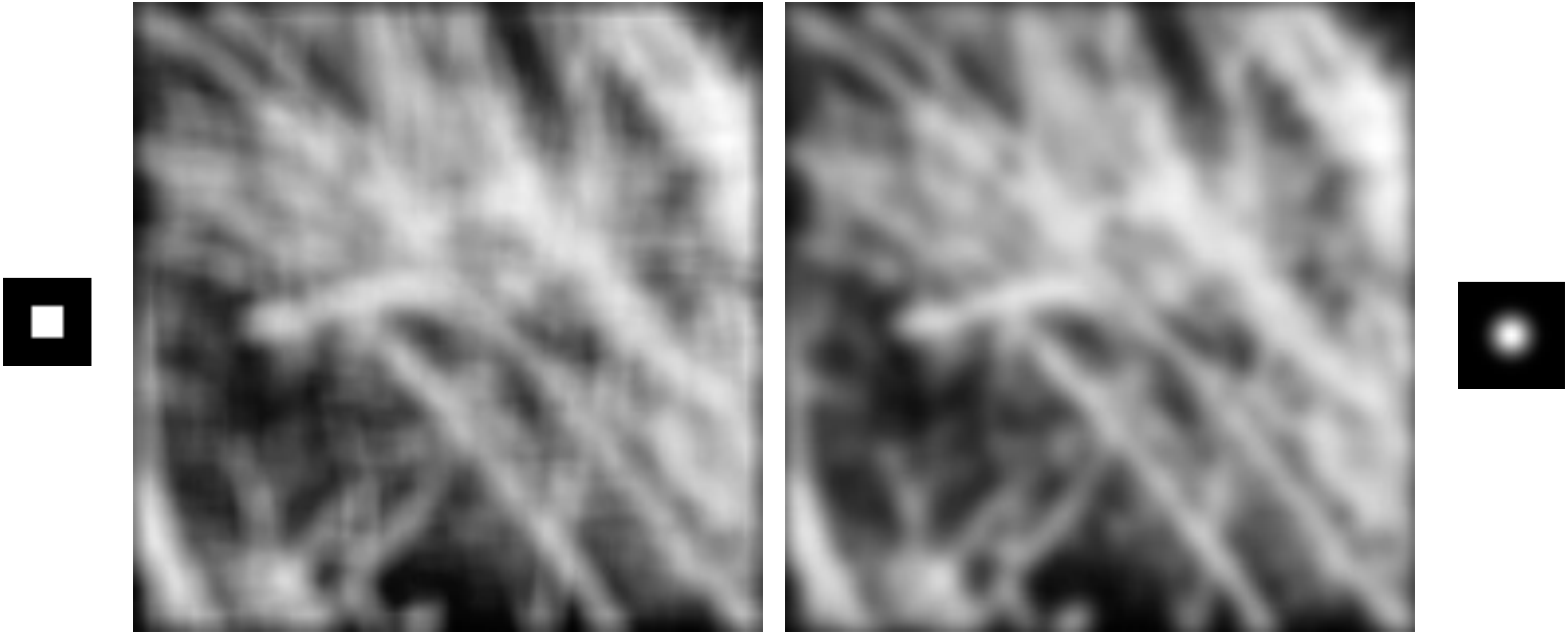
Source: C. Rasmussen

Сглаживание фильтром гаусса

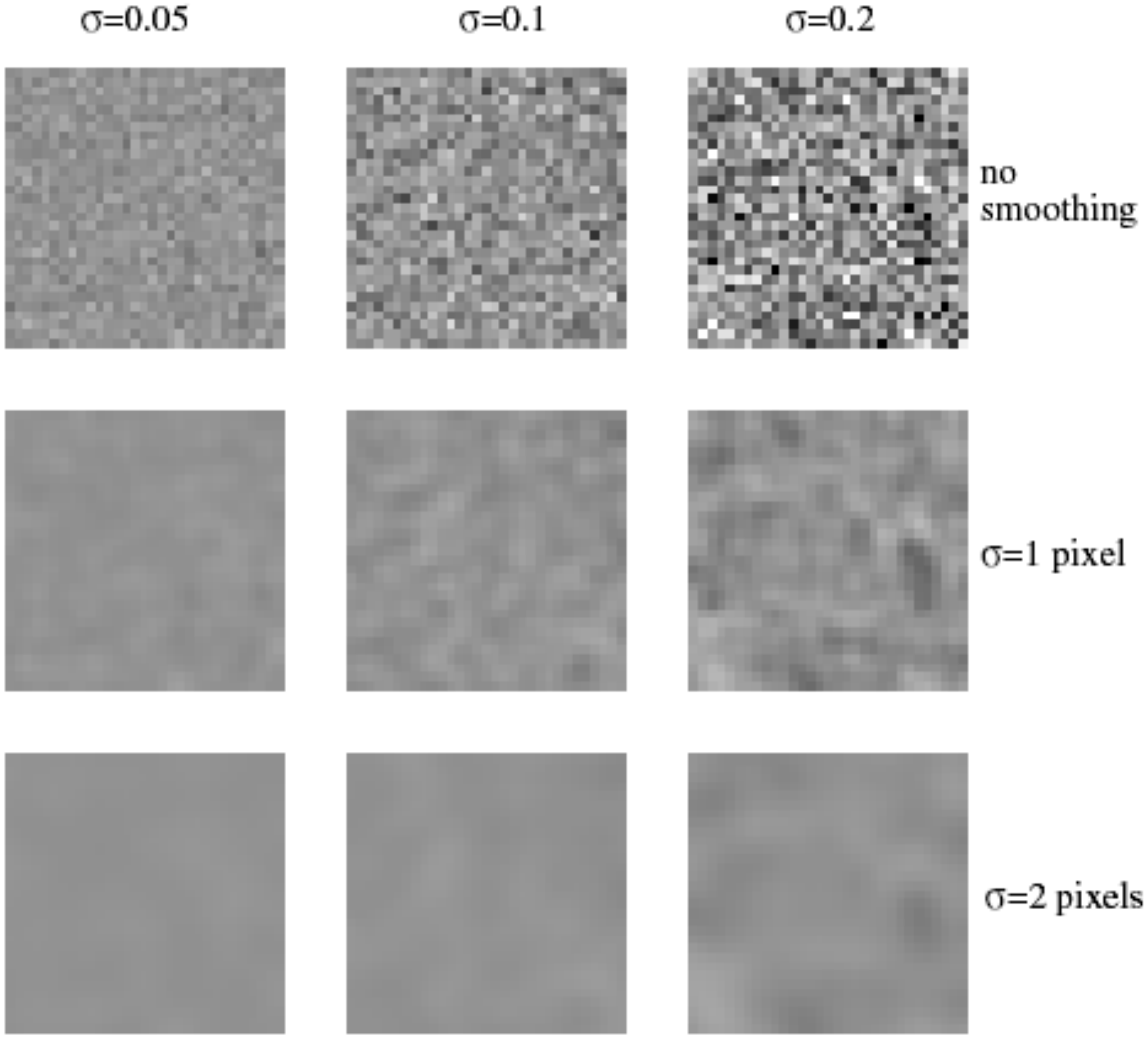




Сравнение бокс и гаусс-фильтра



Подавление шума с помощью фильтра Гаусса



Сглаживающие фильтры подавляют шум, но размывают изображение

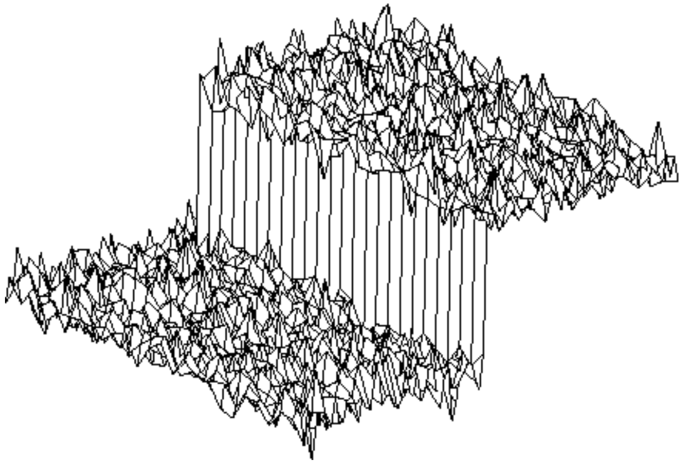
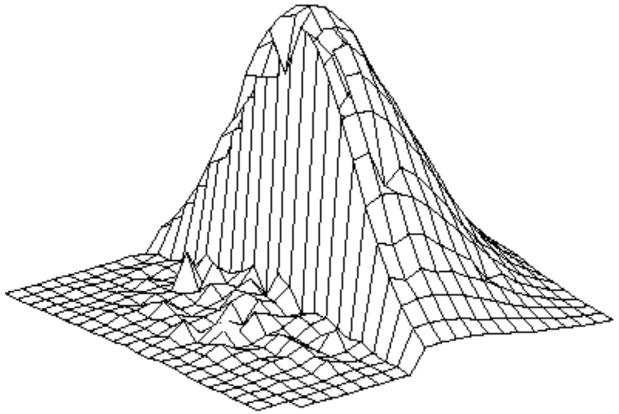
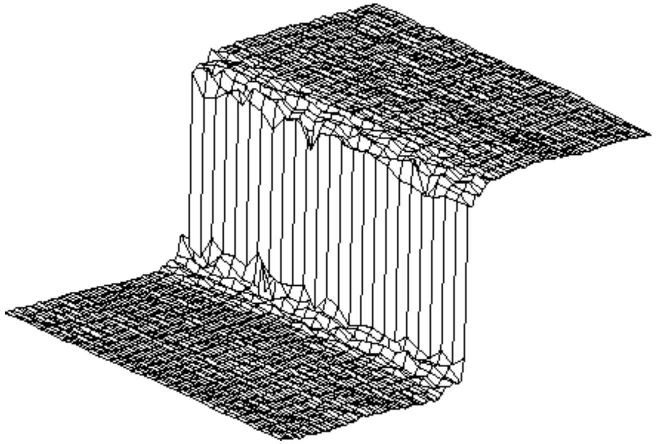
Чем больше размер ядра фильтра, тем сильнее размытие

Slide by S. Lazebnik

Билатеральный фильтр

- В операции свёртки каждый пиксель обрабатывается одинаково
- Но мы можем сделать параметры фильтра зависящими от изображения
- Билатеральный фильтр:

$$y_{ij} = \sum x_{i-k,j-l} \underbrace{w(x_{i-k,j-l} - x_{i,j})}_{\substack{\text{Вес, пропорциональный} \\ \text{близости пикселов по цвет}}} \underbrace{d(x_{i-k,j-l}, x_{i,j})}_{\substack{\text{Параметры} \\ \text{фильтра Гаусса}}}$$

Виды шума

Original

Gaussian noise

Salt and pepper noise

Impulse noise

- **Гауссов** (*Gauss noise*)
  - Аддитивный шум
  - Image(i,j) = true(i,j) + noise(i,j)
  - Колебания яркости, распределенные по нормальному закону
  - $Noise(i,j) \sim N(\mu,\sigma)$

- **Потеря информации (***data drop-out noise***)**
  - **Соль и перец**: случайные черные и белые пиксели
  - **Импульсный:** случайные белые пиксели

Source: S. Seitz

Подавление шума «соль и перец»

3x3

5x5

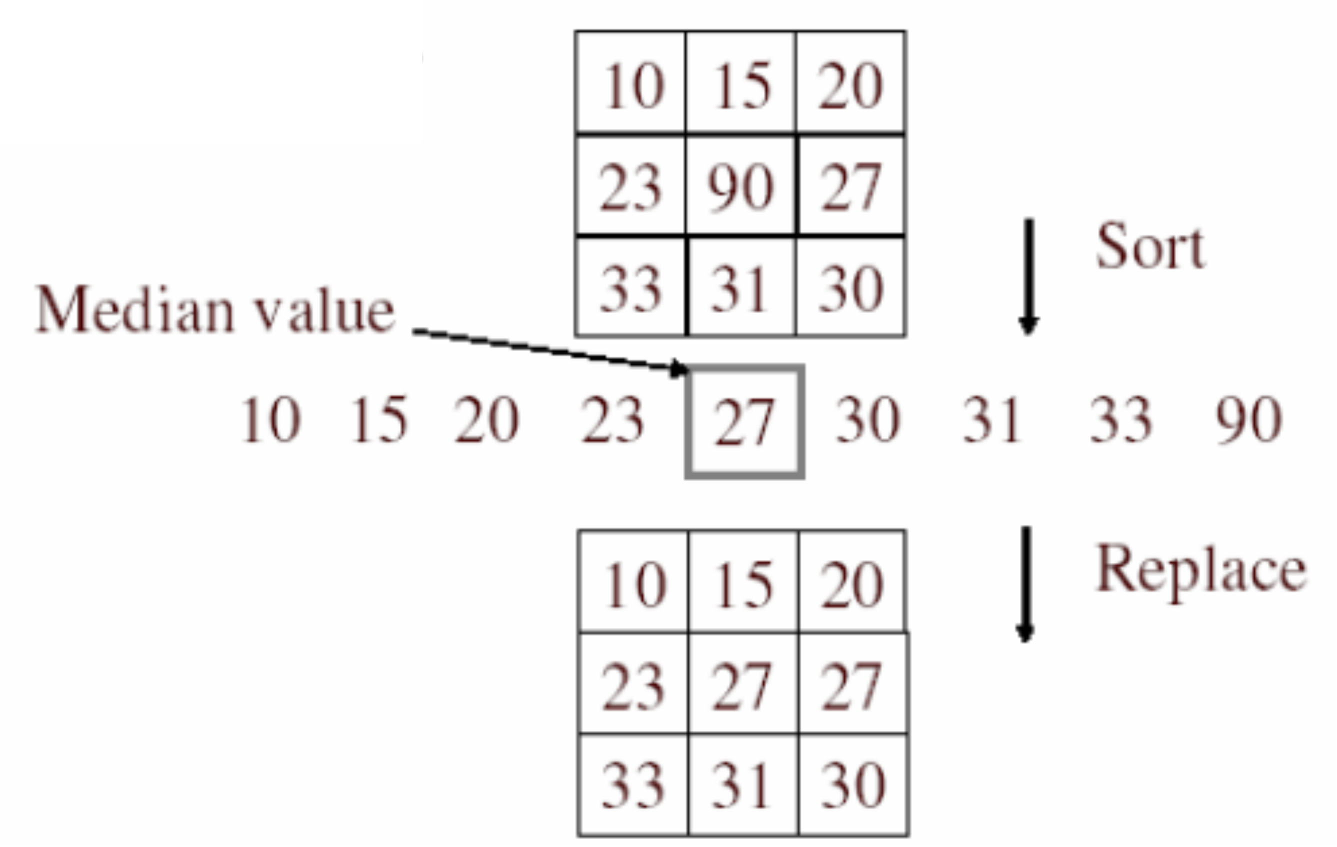
7x7

Применим фильтр Гаусса

Чем результат плох?

Медианный фильтр

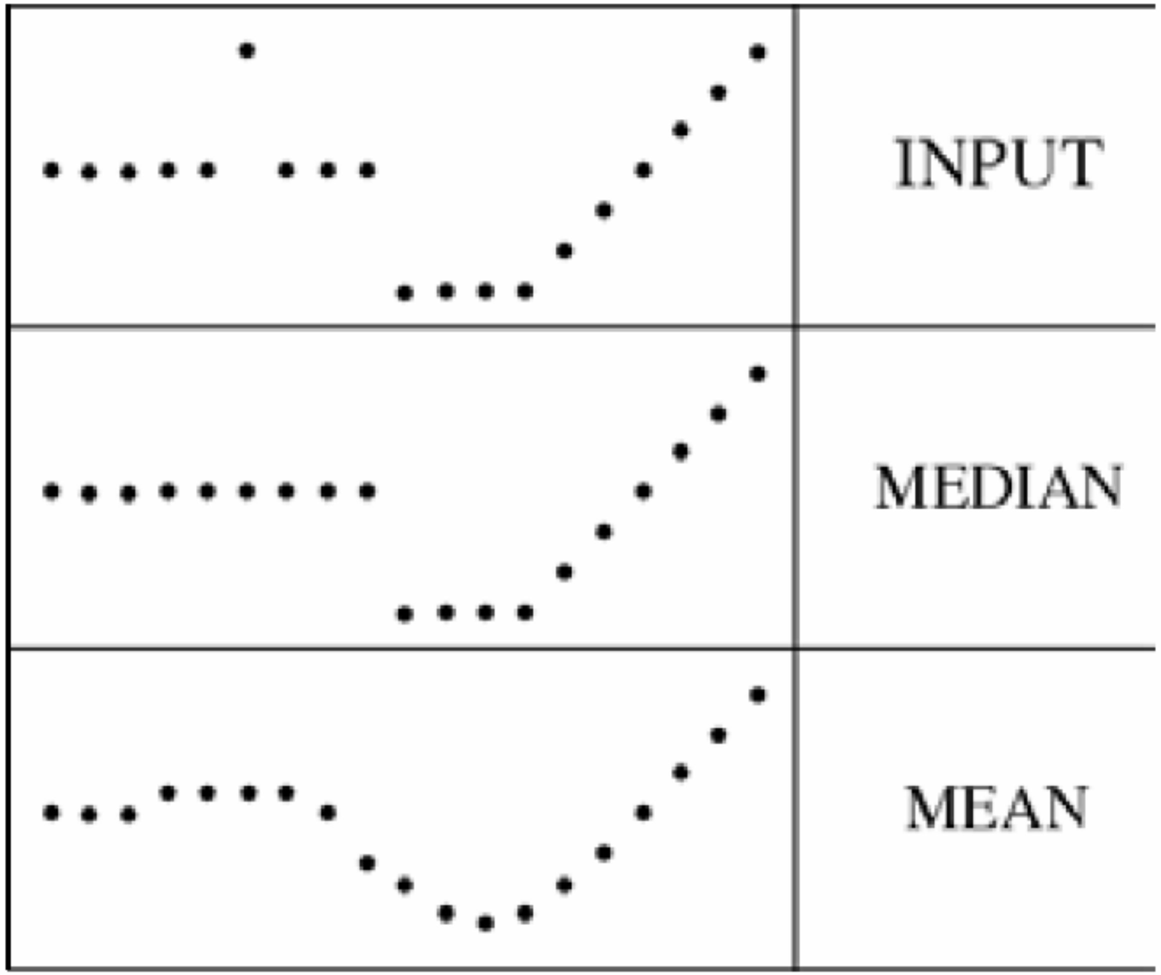
- Выбор медианы из выборки пикселей по окрестности данного



Source: K. Grauman

Свойства медианного фильтр

- В чем отличие медианного фильтра от фильтра гаусса?
- Не порождает новые значения, а берёт существующие
- Сохраняет границы
- «Устойчив» к выбросам (outliers)



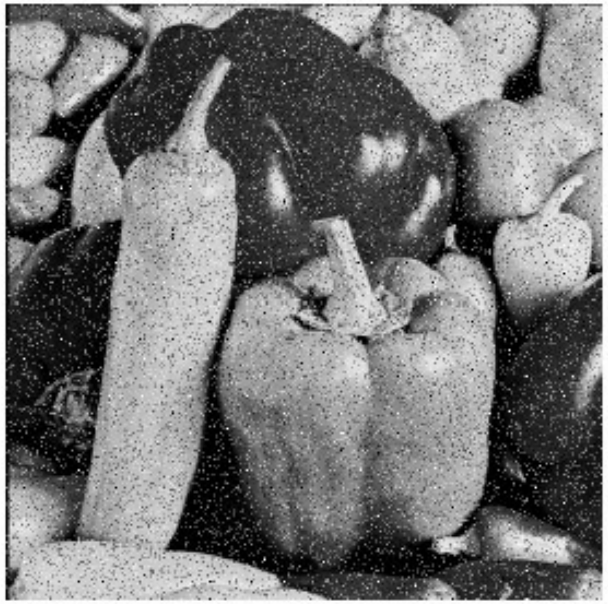
Фильтры размером в 5 пикселей

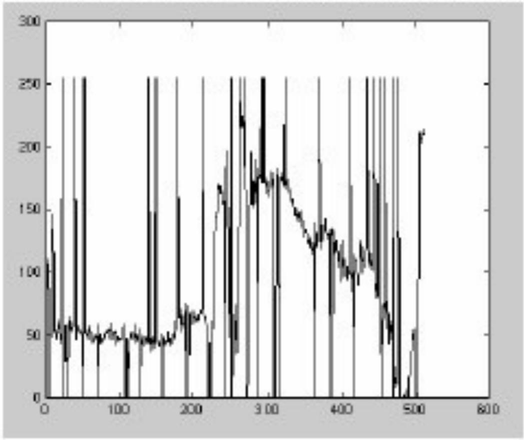
Source: K. Grauman

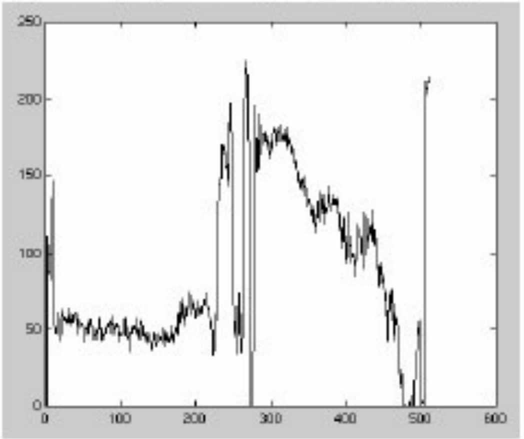
Пример применения медианного фильтра

Шум «соль и перец»

Медианная фильтрация







Source: M. Hebert

Сравнение работы фильтров на импульсном шуме

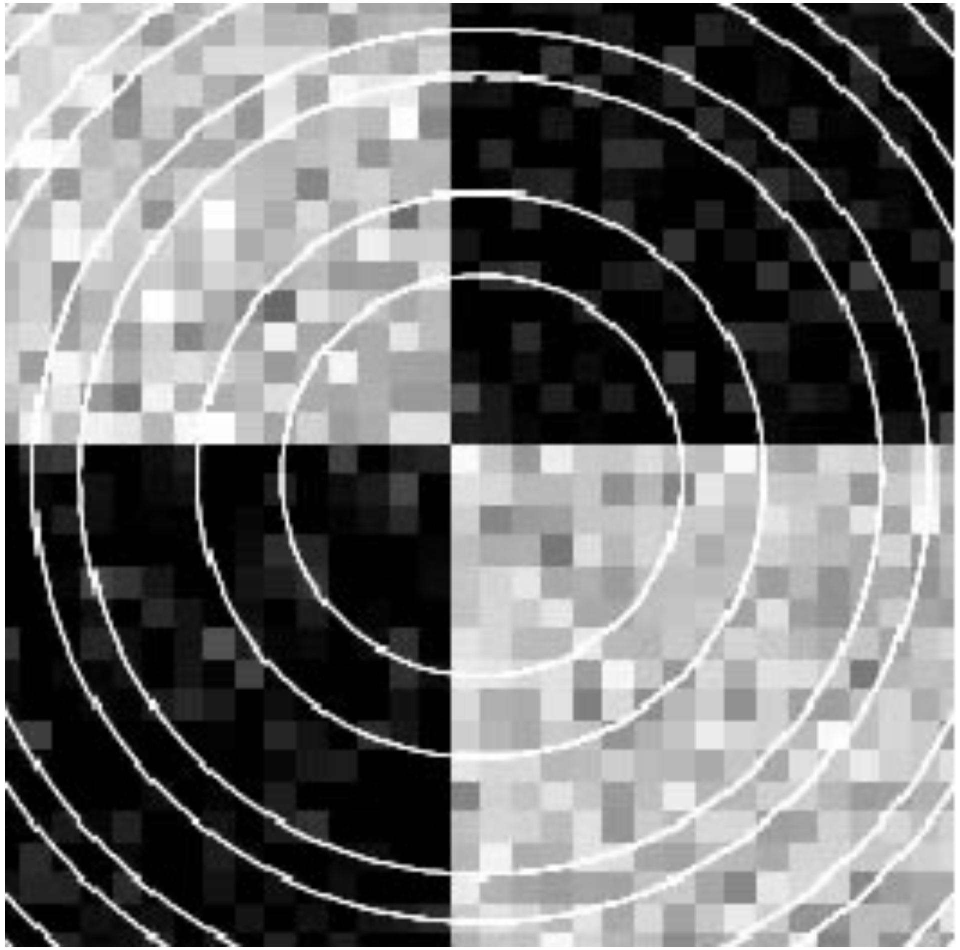
3x3 | 5x5 | 7x7

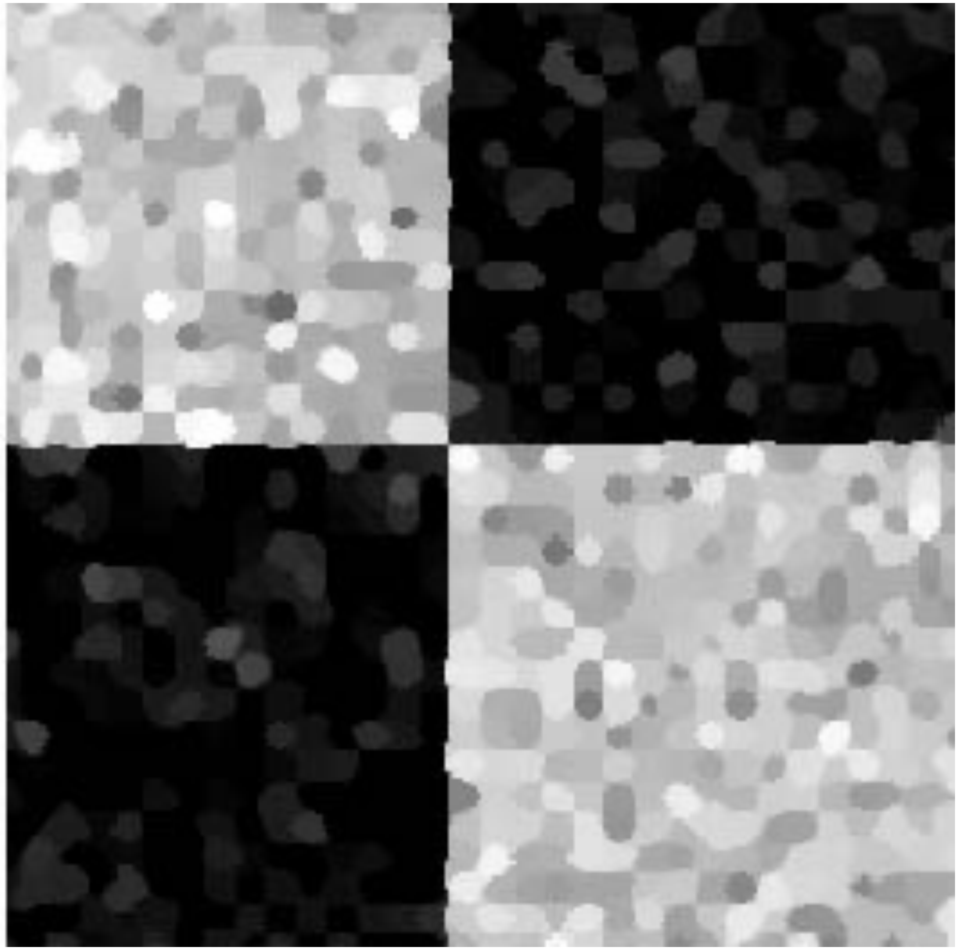
Гауссов

Медианный

Slide by S. Lazebnik

Применение медианного фильтра





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

Ещё про свёртку и приложения фильтров

Линейные фильтры и свёртка

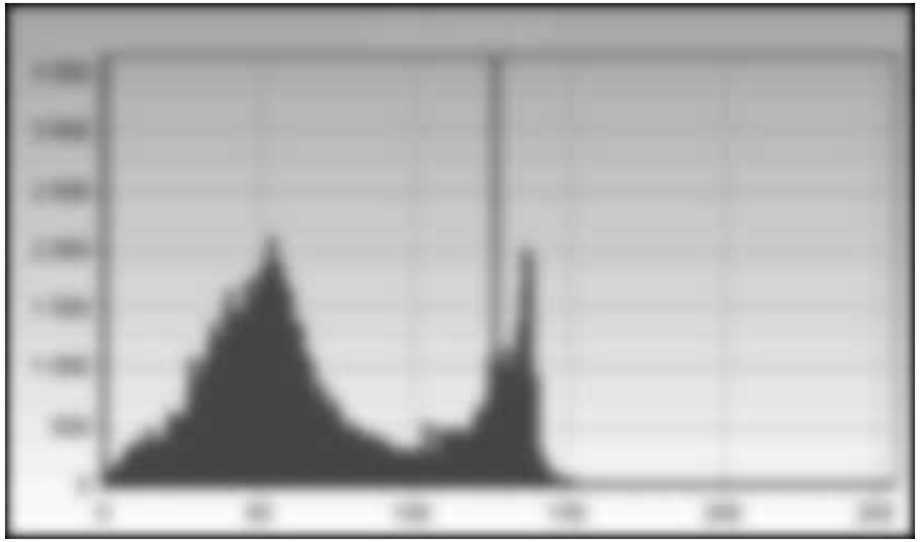
- Пространственный фильтр называется **линейным**, если выполняются свойства **линейности**:
  - filter(f_1 + f_2) = filter(f_1) + filter(f_2)
  - filter(af_1) = a filter(f_1)
- Фильтр называются инвариантным к сдвигу, если выполняется следующее условие:
  filter(shift(f)) = shift(filter(f))
- **Теорема**: 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

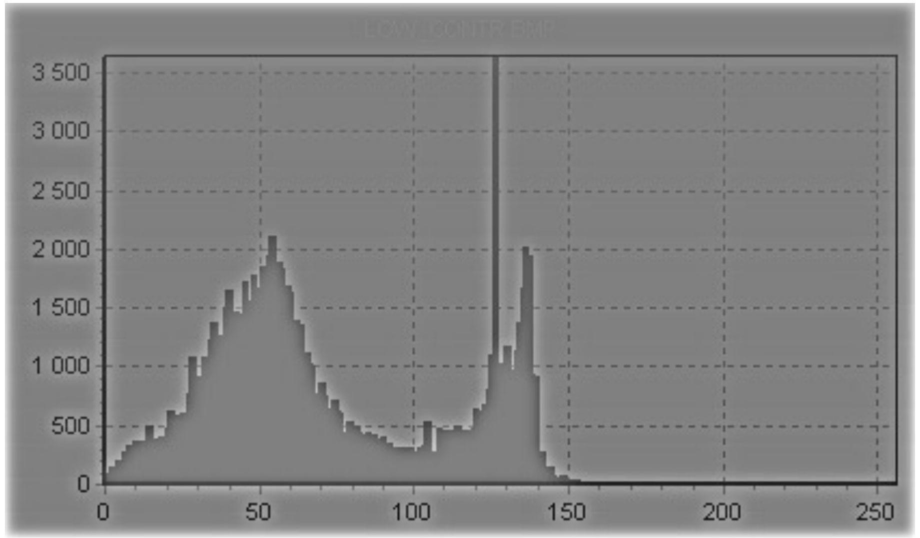
Свойства свёртки

- Коммутативность: $I * K = K * I$
  - Нет никакой разницы между изображением и ядром фильтра
- Ассоциативность: $(I * K_1) * K_2 = I * (K_1 * K_2)$
  - Последовательное применение фильтров эквивалентно применению 1 фильра
- Дистрибутивность по сложению: $\mathrm{I} * (\mathrm{K}_1 + \mathrm{K}_2) = (\mathrm{I} * \mathrm{K}_1) + (\mathrm{I} * \mathrm{K}_2)$
- Домножение на скаляр: $aI * \mathrm{K} = \mathrm{I} * \mathrm{aK} = \mathrm{a}(\mathrm{I} * \mathrm{K})$
- Свёртка с единичным ядром: $I * E = I, E = [\ldots, 0,0,1,0,0, \ldots]$

Slide by S. Lazebnik

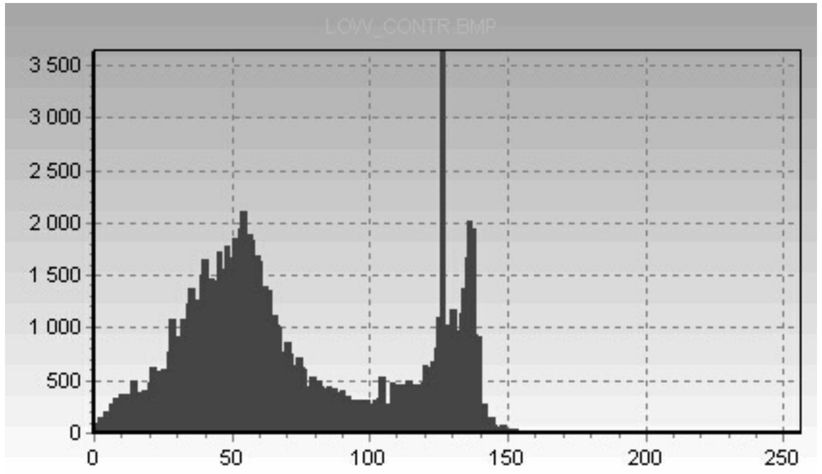
Маленькая экскурсия к Фурье





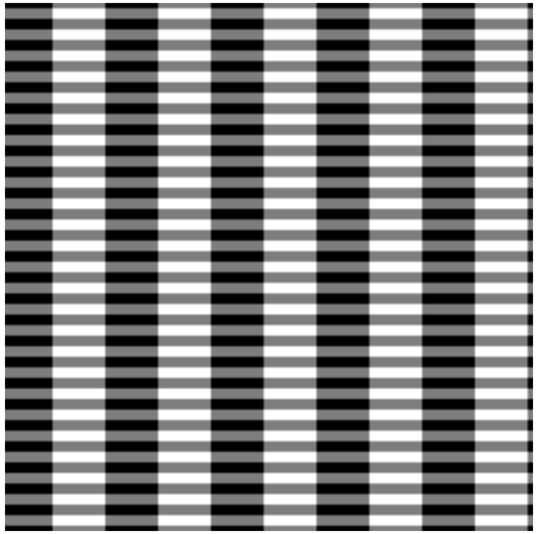
Низкие частоты

Высокие частоты

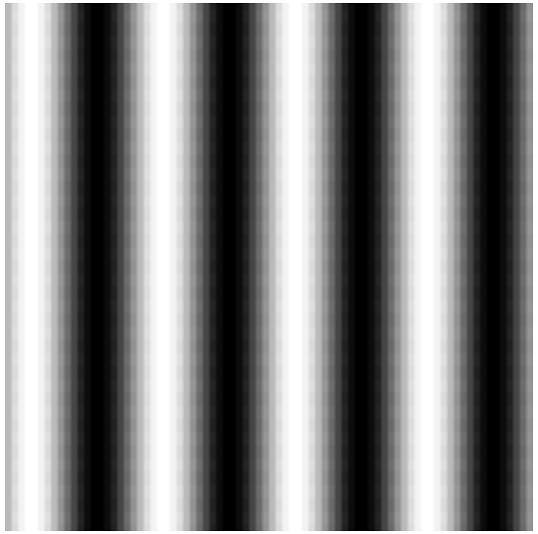


Свойство фильтра Гаусса

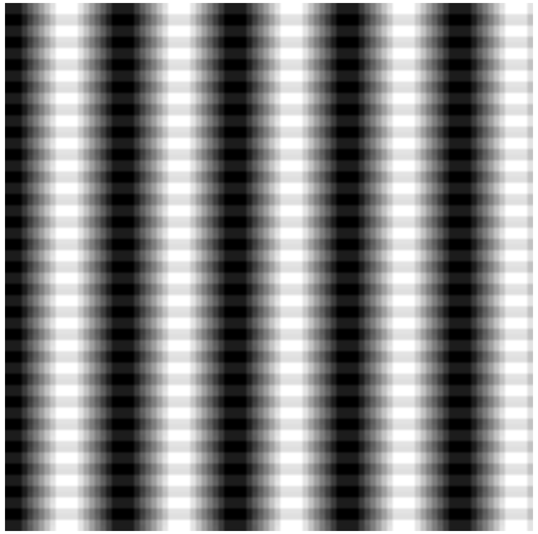
Важное свойство фильтра Гаусса – он является фильтром низких частот, подавляя высокие частоты в сигнале.



Исходное изображение



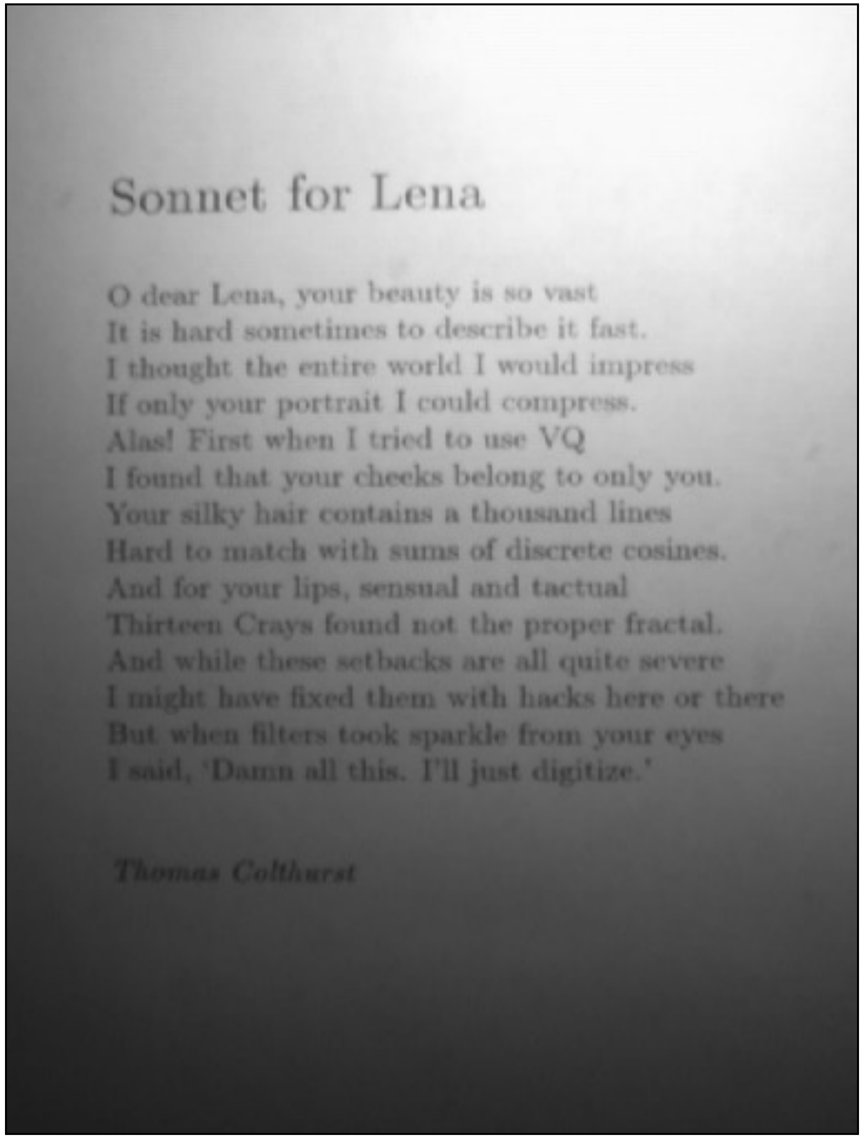
Фильтр Гаусса с Sigma = 4



Усреднение по 49 пикселям (7x7)

Именно поэтому он сглаживает изображение

Пример другого применения фильтра Гаусса



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 sums of discrete cosines.
And for your lips, sensual and tactual
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, 'Damn all this. I'll just digitize.'

Thomas Colthurst

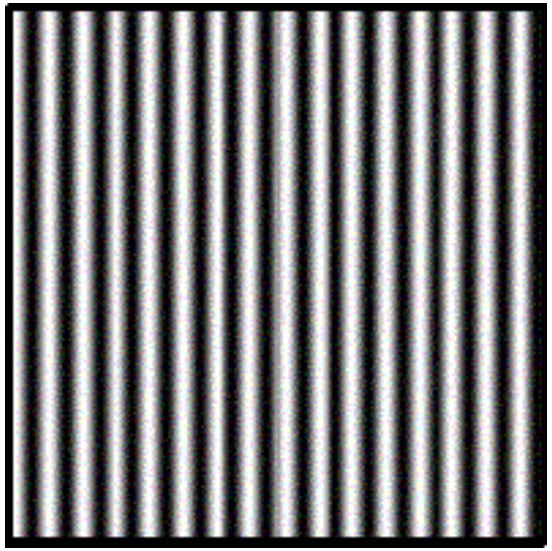
Что можно сказать
про это изображение?

Компенсация разности освещения

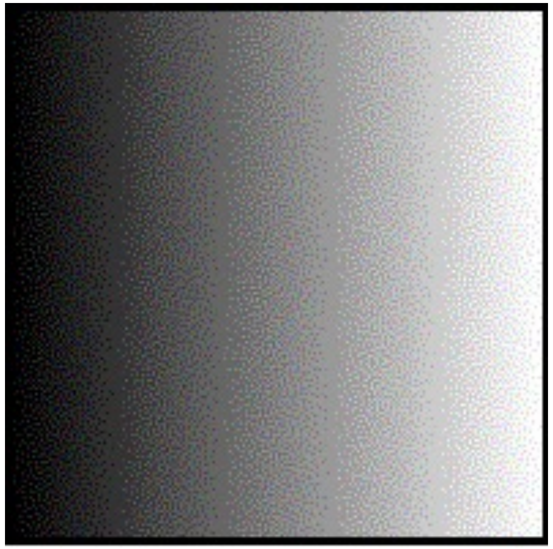
Идея:

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

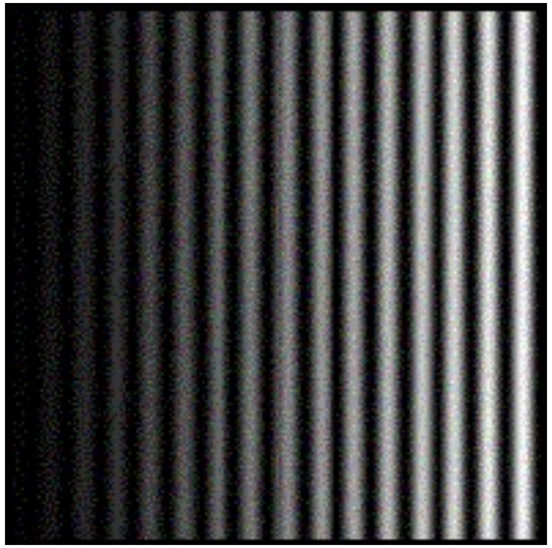
Плавные изменения яркости относятся к освещению, резкие - к объектам.



объект $r(i,j)$



освещение $l(i,j)$



Изображение освещенного объекта $I(i,j)$

Алгоритм Single scale retinex (SSR)

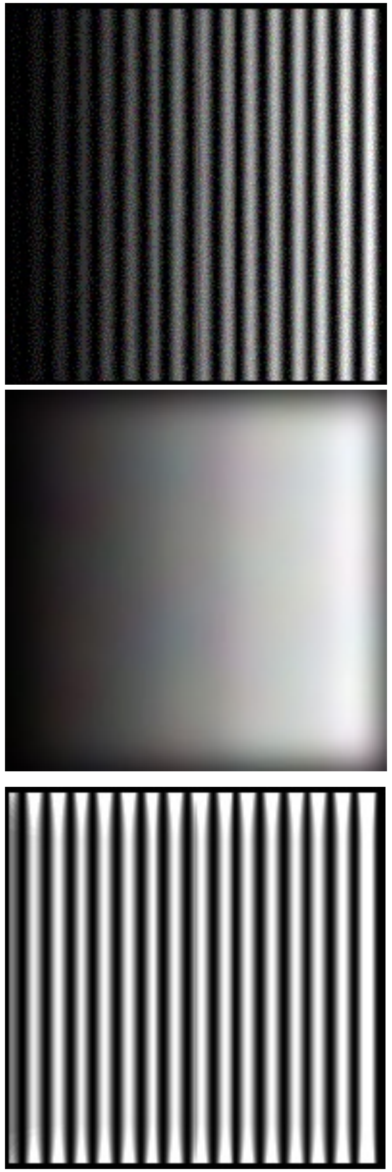
- Получить приближенное изображение освещения путем низочастотной фильтрации

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

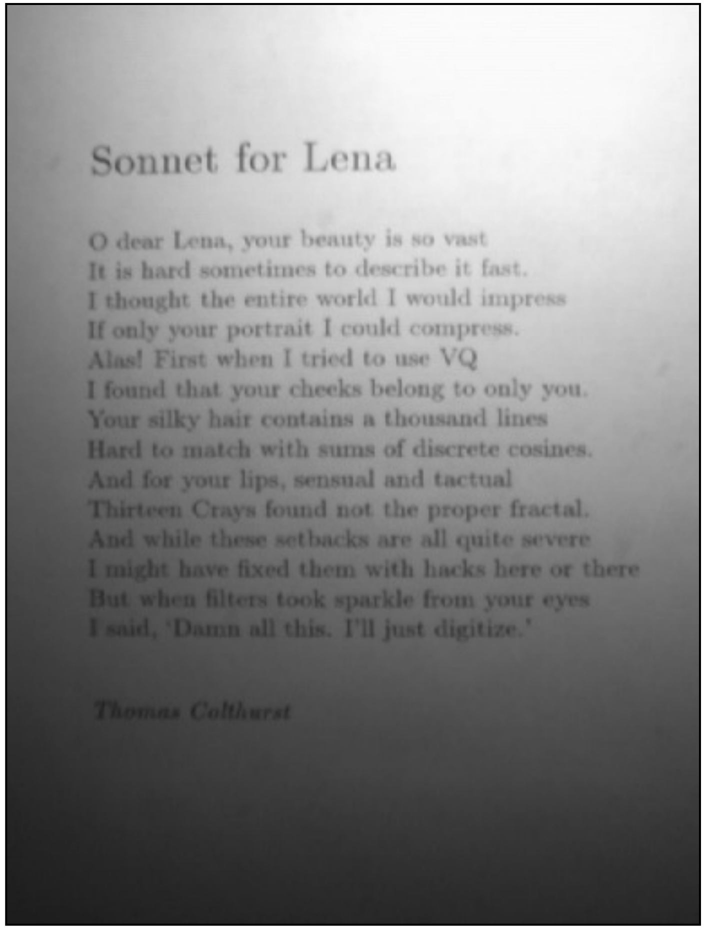
- Восстановить изображение по формуле

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

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



Пример компенсации разности освещения



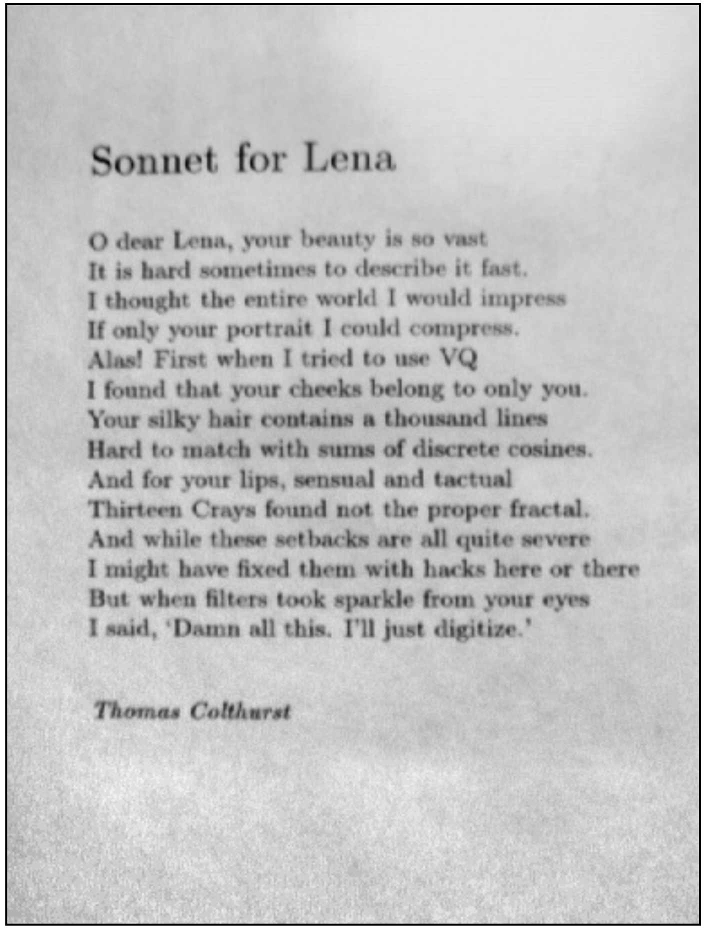
Sonnet for Lena

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 sums of discrete cosines.
And for your lips, sensual and tactual
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, 'Damn all this. I'll just digitize.'

Thomas Colthurst

/

=



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 sums of discrete cosines.
And for your lips, sensual and tactual
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, 'Damn all this. I'll just digitize.'

Thomas Colthurst

Gauss 14.7 пикселей

Повышение резкости (sharpening)

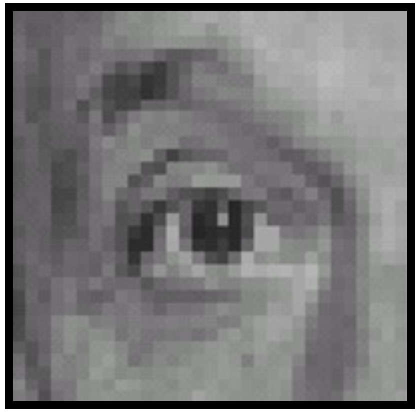
 $-$  $=$ 

 $+\alpha$  $=$ 

Насколько это корректно и к каким проблемам может привести?

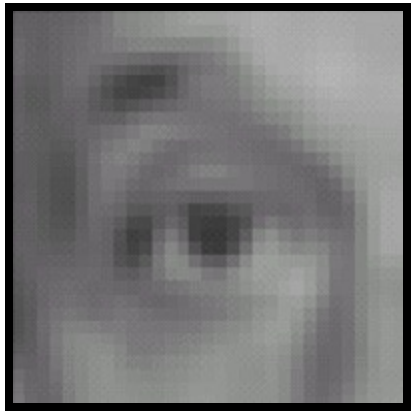
Быстрая фильтрация

Box-фильтр



Исходное

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



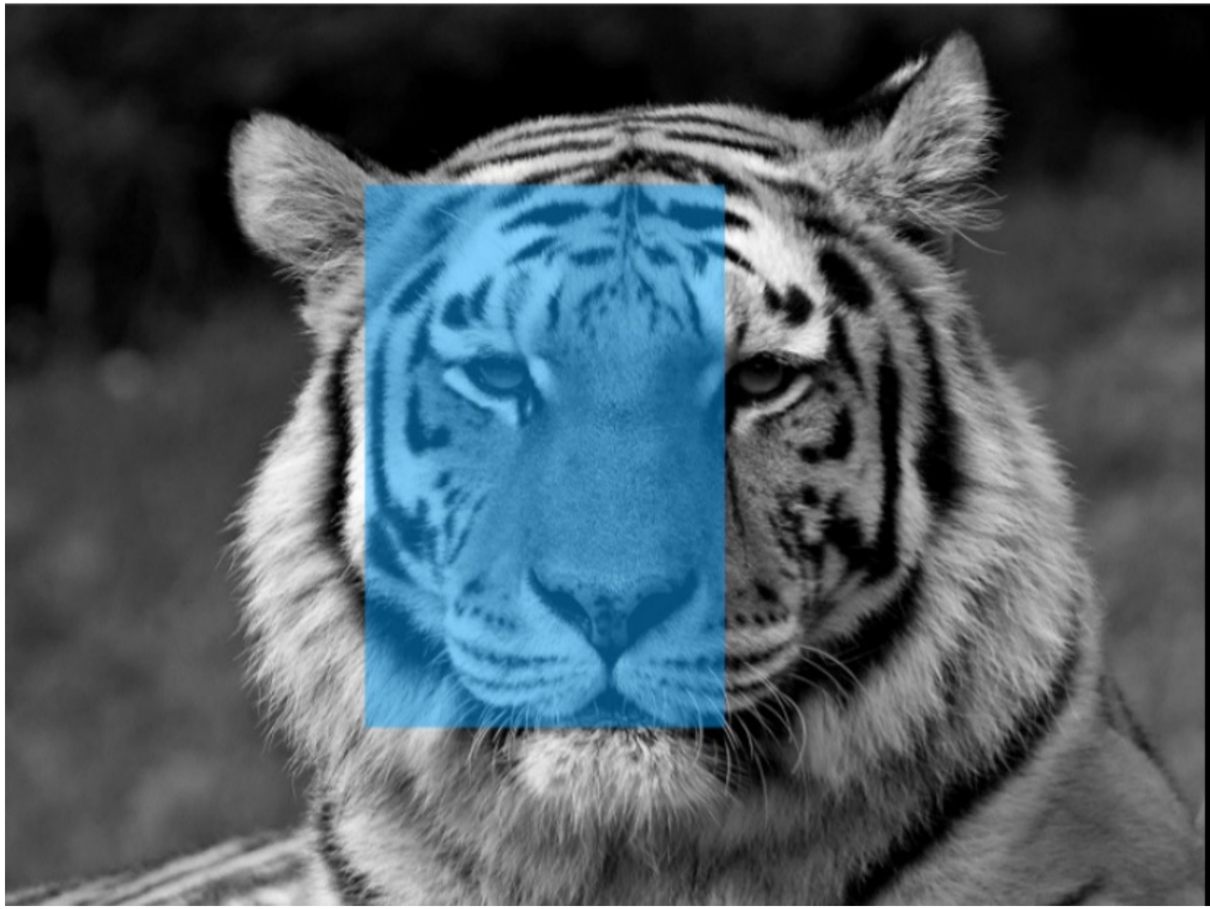
Результат

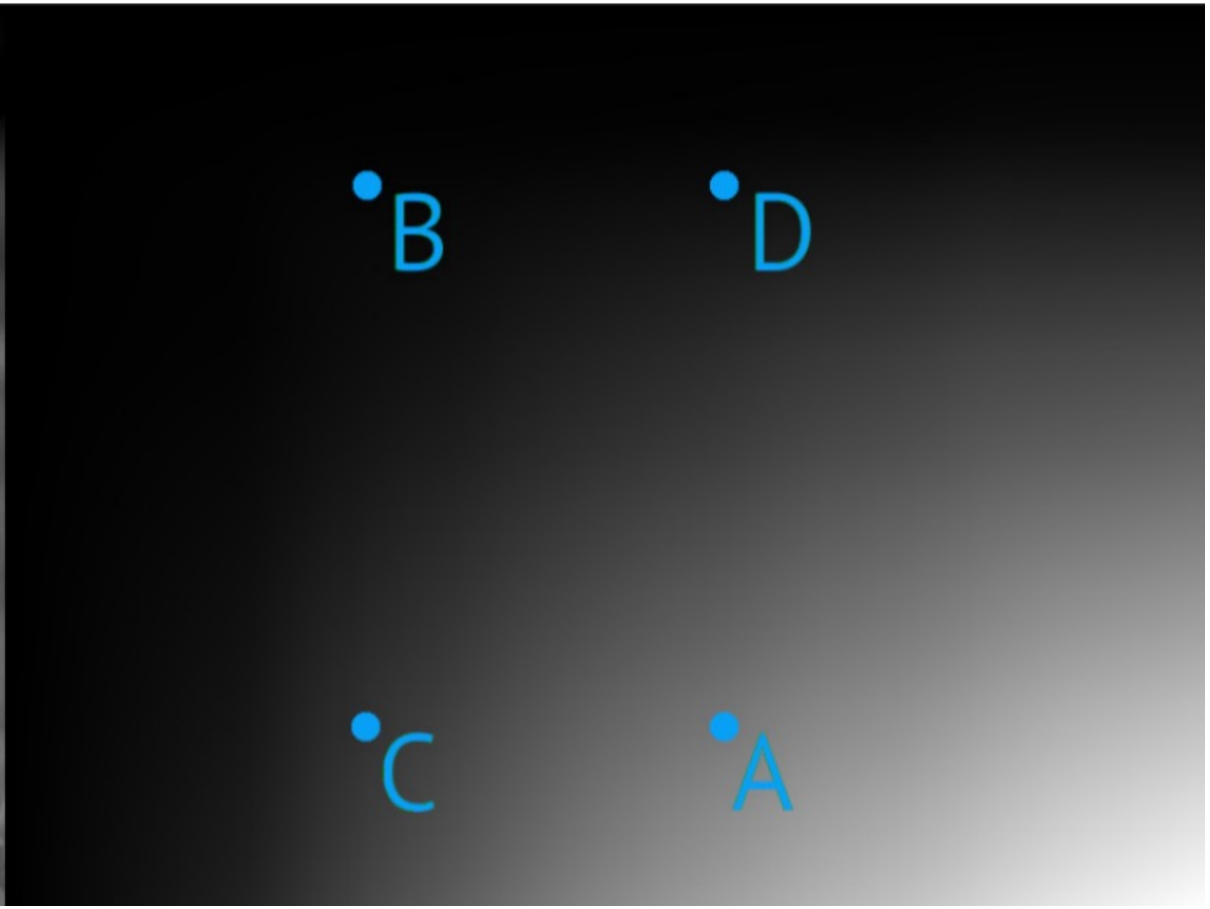
Source: D. Lowe

Быстрый box-filter

$$S[x,y] = \sum_{i=0}^{x} \sum_{j=0}^{y} I[i,j]$$

Быстрый box-filter





$$
\begin{aligned}
Sum(x_1, y_1, x_2, y_2) &= A + B - C - D \\
&= S[x_2, y_2] + S[x_1 - 1, y_1 - 1] - S[x_1 - 1, y_2] - S[x_2, y_1 - 1]
\end{aligned}
$$

Сепарабельные фильтры

- 2D фильтр называется сепарабельным, если его можно разложить в произведение 1D свёрток
- Фильтр Гаусса сепарабелен!

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

- Вычислительная сложность свёртки $\mathrm{N}^2 K^2$
- Вычислительная сложность сепарабельного фильтра $\mathrm{N}^2 K$

Аппроксимация с сепарабельным фильтром

- Фильтр сепарабельный, если ранг ядра фильтра = 1

$$\mathrm{K} = \mathrm{vh}^{\mathrm{T}}$$

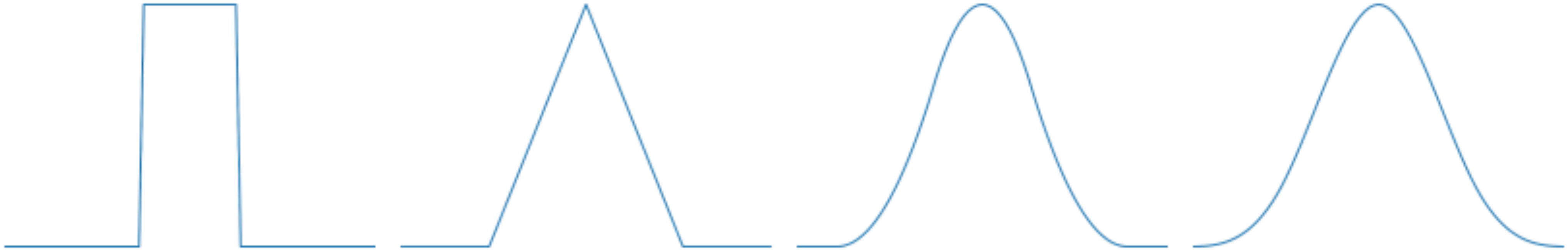
- Несапарабельные фильтры можно факторизовать с помощью SVD

$$\mathrm{K} = \sum_i \sigma_i u_i v_i^T$$

- И аппроксимировать с помощью последовательности сепарабельных фильтров

$$\mathrm{K}_1 = \sqrt{\sigma_1} u_1 * \sqrt{\sigma_1} v_1^T, \mathrm{K}_2 = \sqrt{\sigma_2} u_2 * \sqrt{\sigma_2} v_2^T, \ldots$$

Аппроксимация фильтра Гаусса



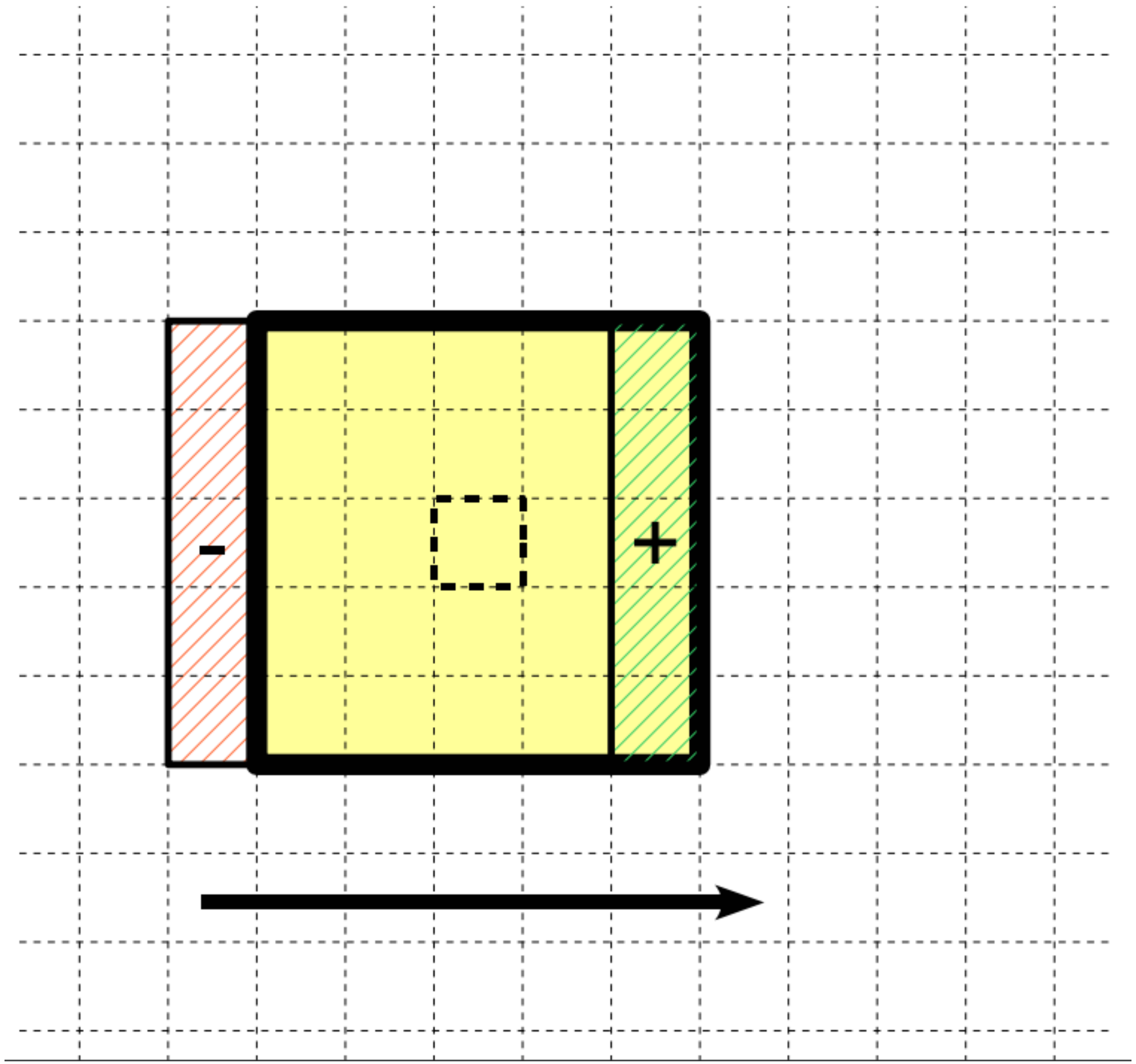
Фильтр Гаусса с параметром σ можно аппроксимировать через N бокс-фильтров ширины $\sigma\sqrt{12/N}$

На практике достаточно N=3. Ошибка составляет около 3%

Быстрая медианная фильтрация

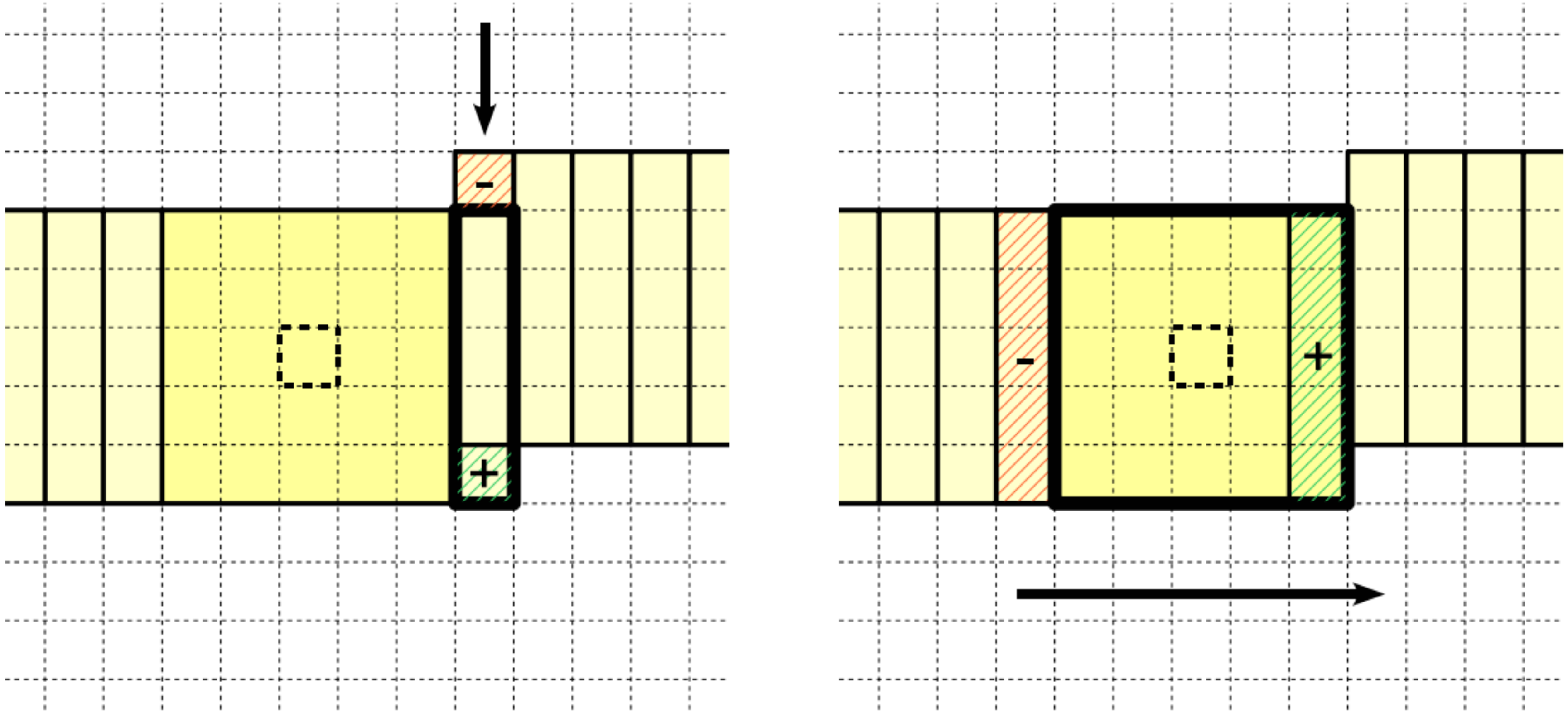
- Вычислительная сложность медианной фильтрации $\mathrm{N}^2 K^2 \log(K)$ при использовании быстрой сортировки
- Huang et al. 1979: $\mathrm{N}^2 K$
- Perreault el al. 2007: N^2

Быстрая медианная фильтрация (линейное время)



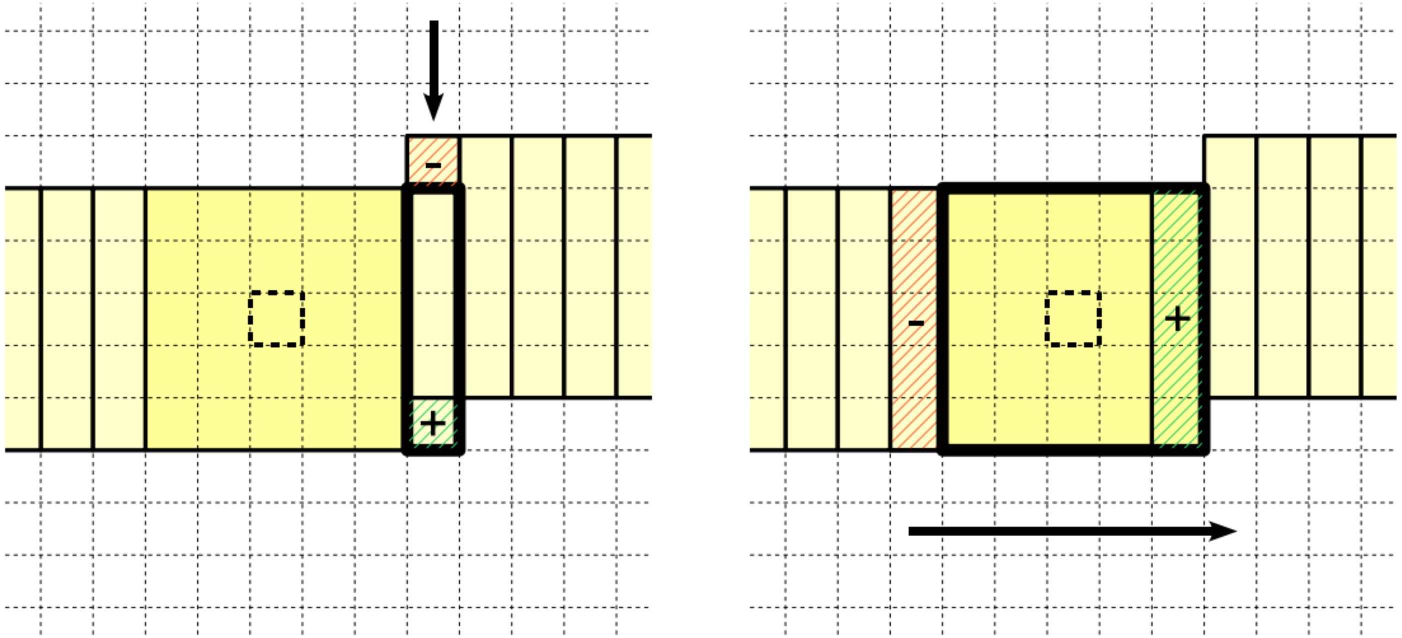
- Поддерживаем гистограмму яркостей для текущего окна
- При сдвиге вычитаем из неё пиксели одного столбца, и добавляем пиксели второго столбца
- Это порядка k операций

Быстрая медианная фильтрация (константное время)



- Поддерживаем ещё и гистограммы всех столбцов изображения, вычитая и добавляя её из гистограммы фильтра
- При обработке следующих строк, нам нужно пересчитывать гистограммы столбцов за 2 операции (вычитаем и добавляем пиксель)

Дополнительные программные трюки



- 16bits для корзин гистограммы, использование векторных операций
- Разбиваем изображение на вертикальные полосы для лучшего использования кэша при расчёте гистограм, но приходится граничные эффекты учитывать
- Поддерживаем 2 вида гистограм – 16 и 256 корзин

Извлечение информации из изображения
и детектор Кэнни

Пример

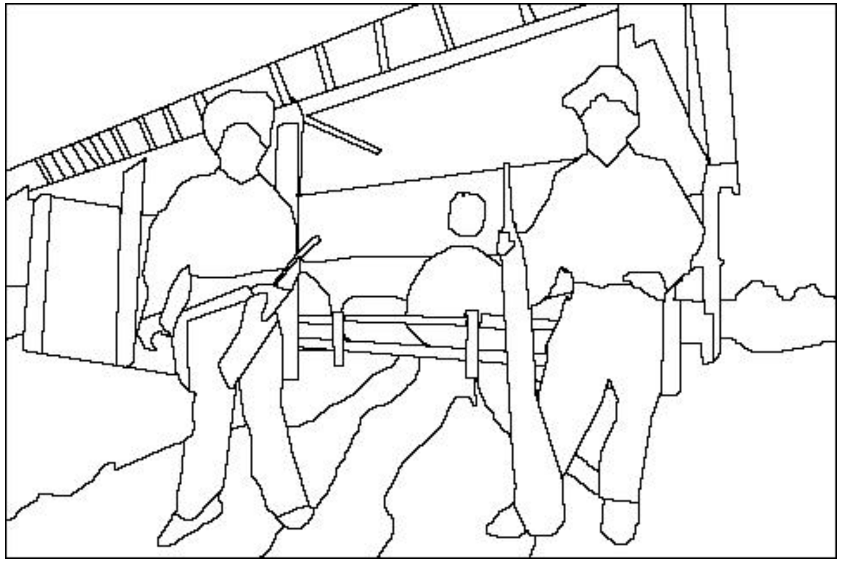
Что можно сказать про изображение?

Что такое край (edge)?

Разметка человеком

изображение

разметка вручную



Berkeley segmentation database:

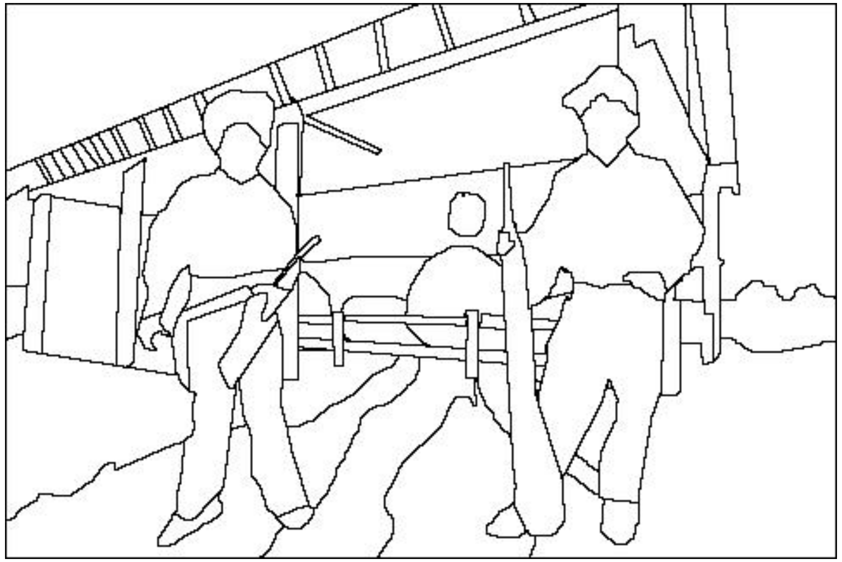
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

Края обычно соответствует границам визуально отличающихся друг от друга областей изображения

Края и области

изображение

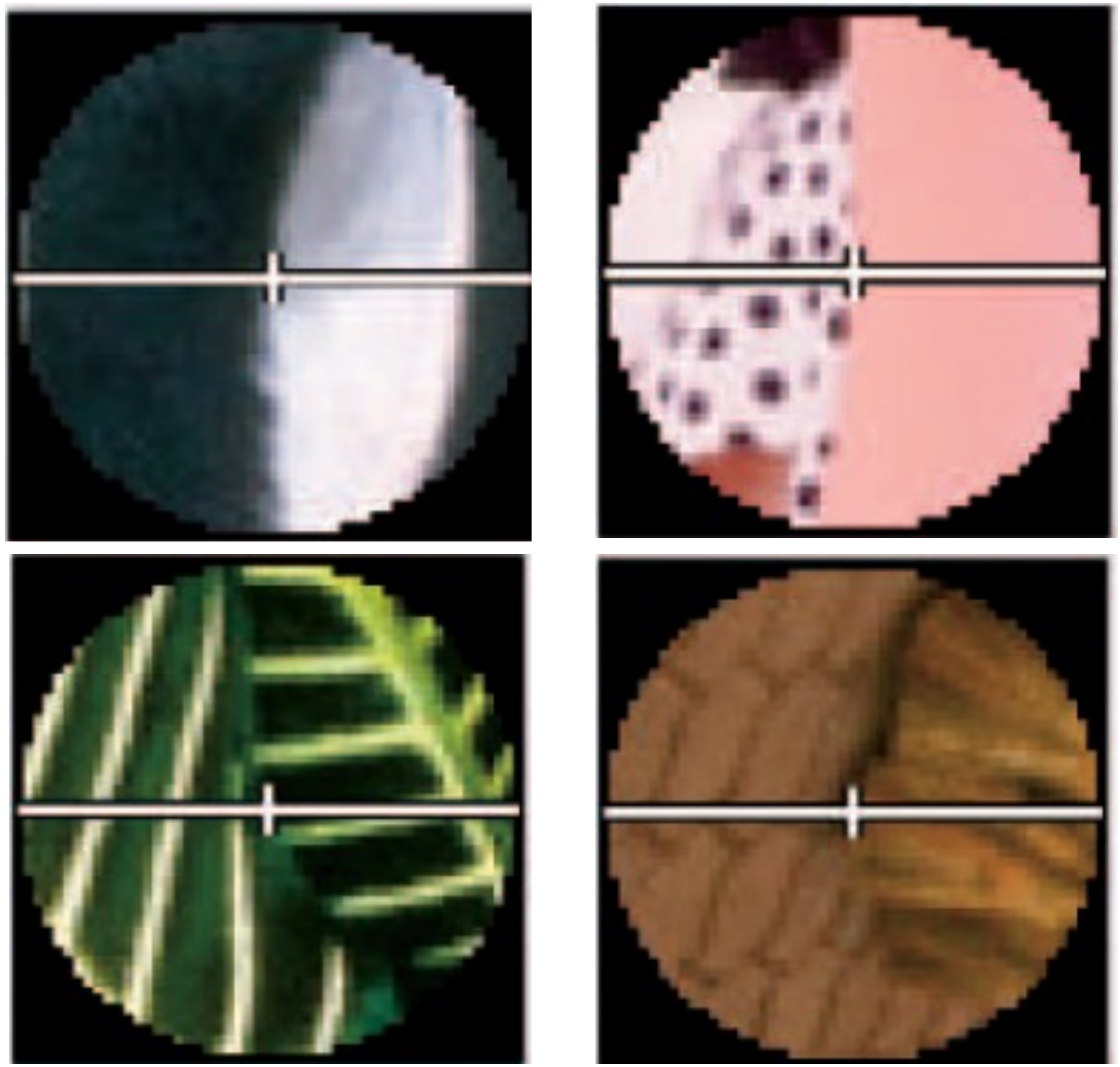
разметка вручную



Смежные задачи:

- Нахождение краёв в изображении
- Разделение изображения на визуально однородные области (визуальная сегментация)

Примеры краёв

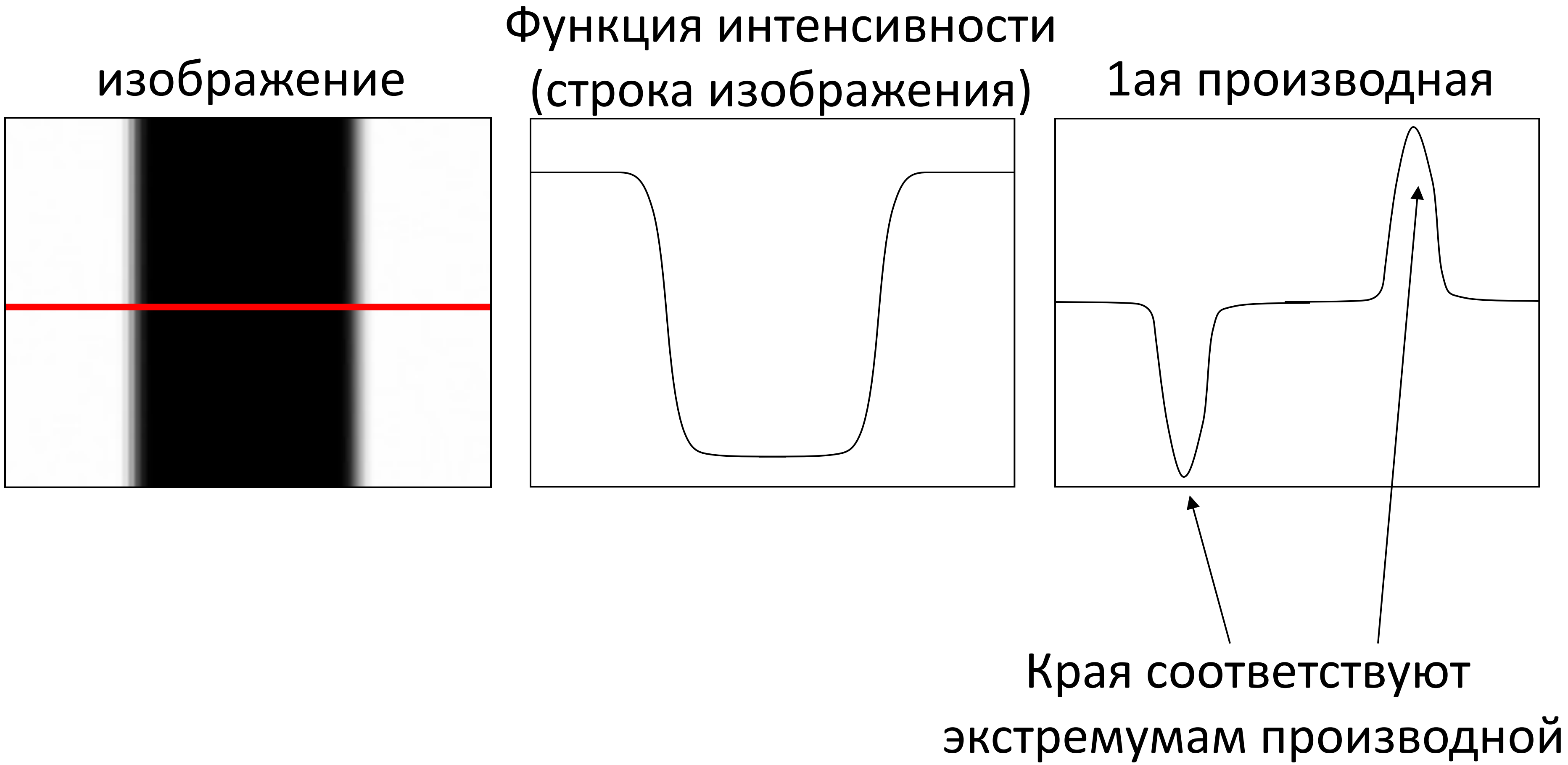


По каким признакам мы определяем наличие границы?

Какой признак простейший?

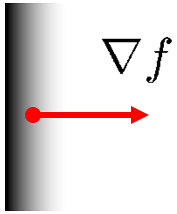
Базовое определение края

- Край – это точка резкого изменения значений функции интенсивности изображения

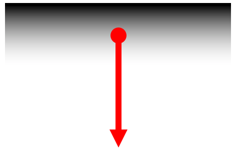


Градиент изображения

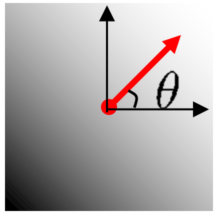
- Градиент изображения: $$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

- 

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



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



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

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:

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

Source: Steve Seitz

Дифференцирование и свёртка

- Для функции 2х переменных, f(x,y):

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

- 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

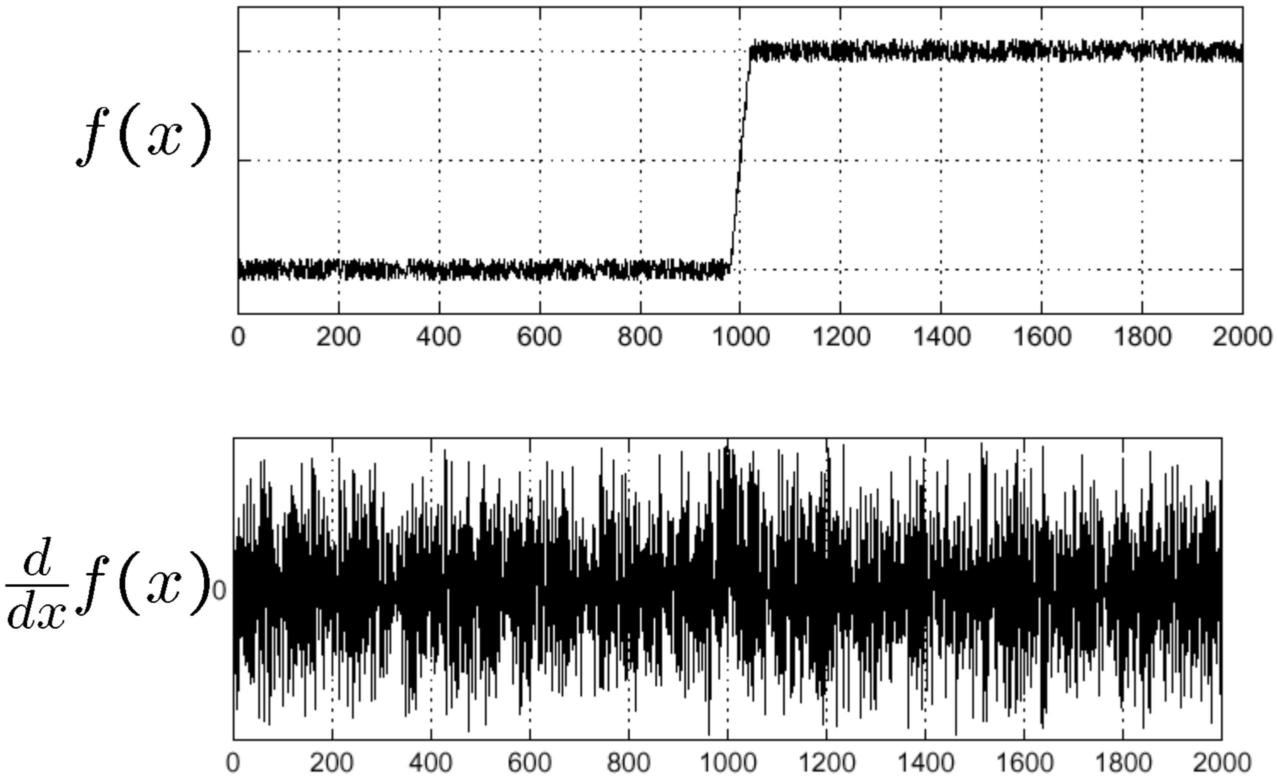
-1	**1**

Простейший фильтр

Source: D. Forsyth, D. Lowe

Влияние шума

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



Край исчез

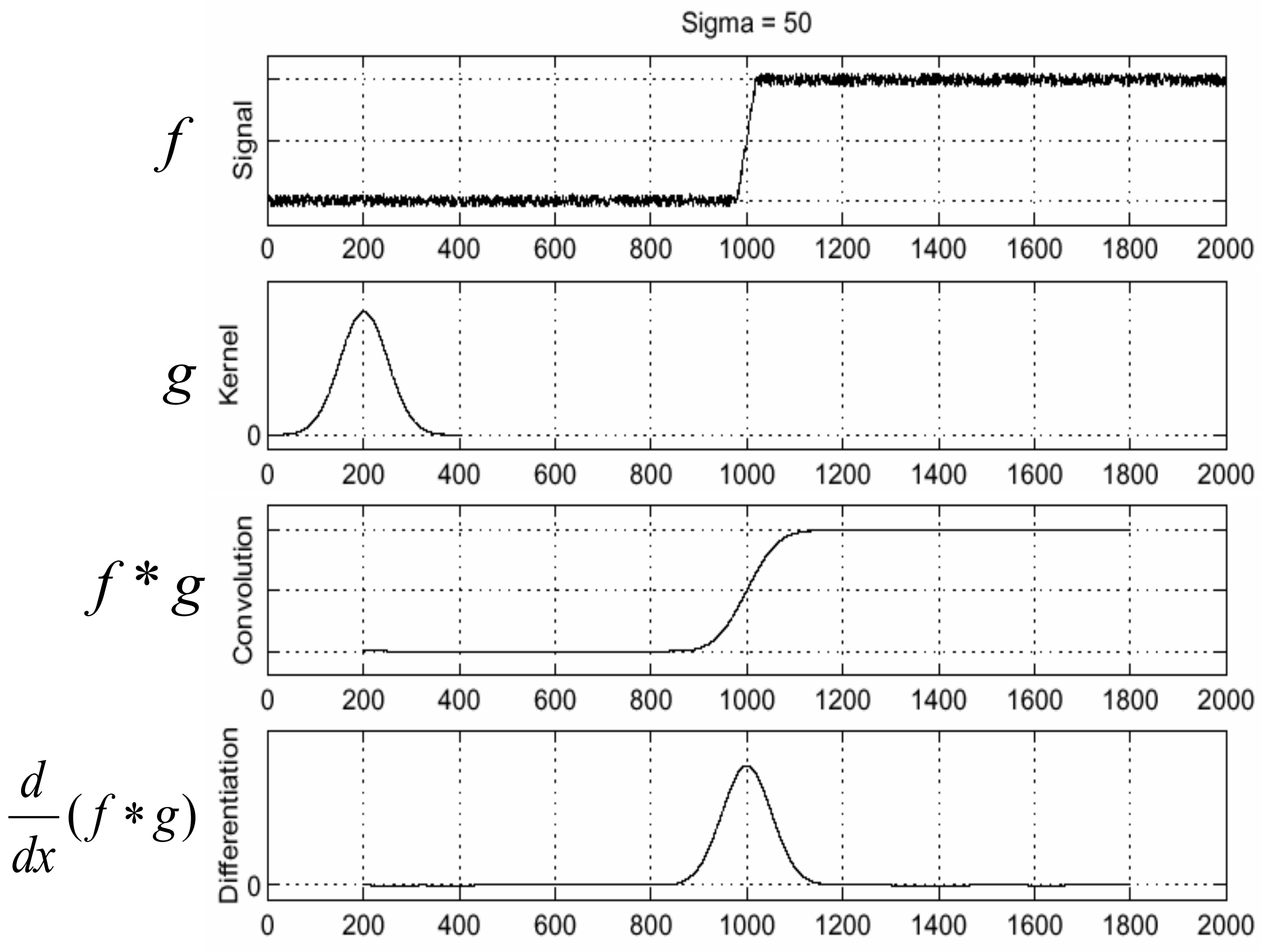
Source: S. Seitz

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

Source: D. Forsyth

Предобработка (сглаживание)



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

Source: S. Seitz

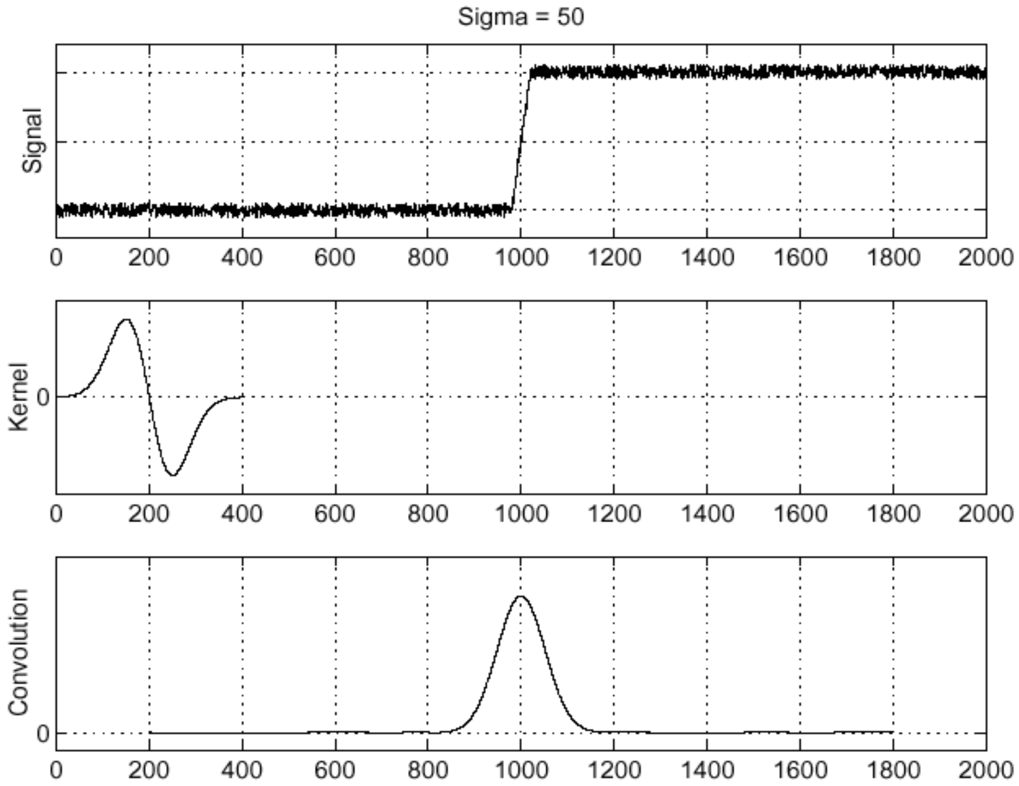
Свойства свертки

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

f

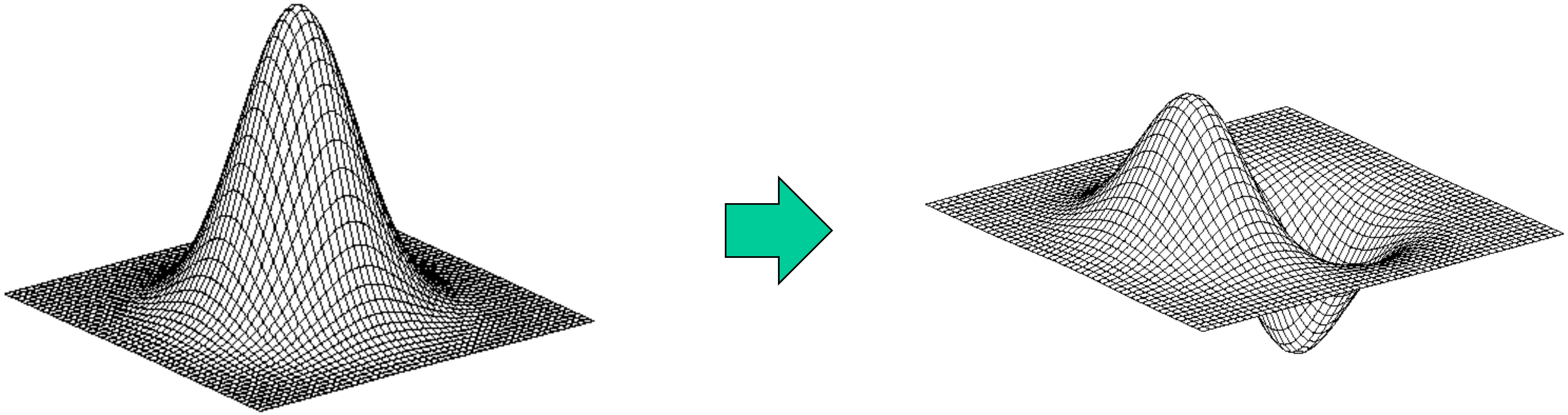
$\frac{d}{dx} g$

$f * \frac{d}{dx} g$

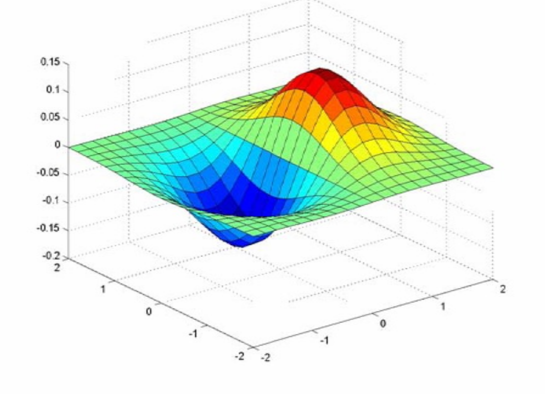


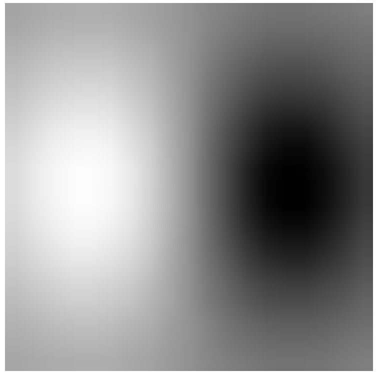
Source: S. Seitz

Производная фильтра Гаусса

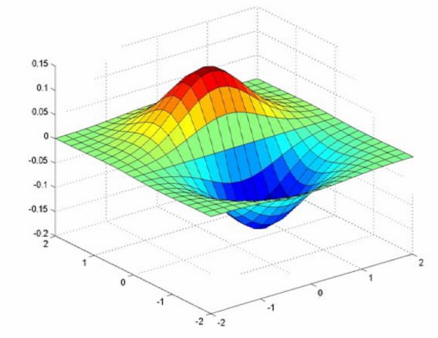


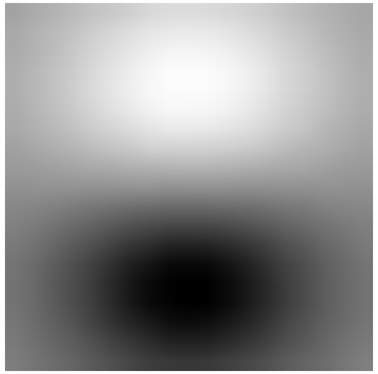
По x





По y:





Slide by S. Lazebnik

Известные фильтры

Несколько фильтров, по разному оценивающие производные по направлению и интегрирующие шумоподавление:

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

Робертса

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

Превитт

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

Собель 3x3

$$\begin{bmatrix} +3 & +10 & +3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix} \quad \begin{bmatrix} +3 & 0 & -3 \\ +10 & 0 & -10 \\ +3 & 0 & -3 \end{bmatrix}$$

Scharr фильтр

$$\begin{bmatrix} 1 & 2 & 0 & -2 & -1 \\ 4 & 8 & 0 & -8 & -4 \\ 6 & 12 & 0 & -12 & -6 \\ 4 & 8 & 0 & -8 & -4 \\ 1 & 2 & 0 & -2 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Собель 5x5

Примеры карт силы краев

Робертса

Превитт

Собеля

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 с использованием гистерезиса (двойного порога)
   - Выше верхнего порога – «сильные края»
   - Ниже нижнего порога – шум (отсекаем)
   - Посередине – потенциальные края

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

Посмотрим на примере Lena

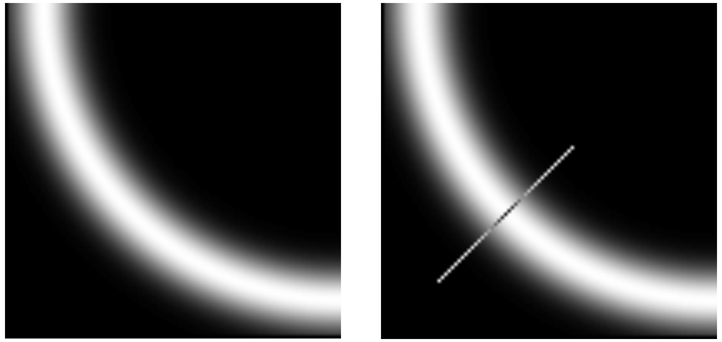
Оригинальное изображение
Lena (Lenna)

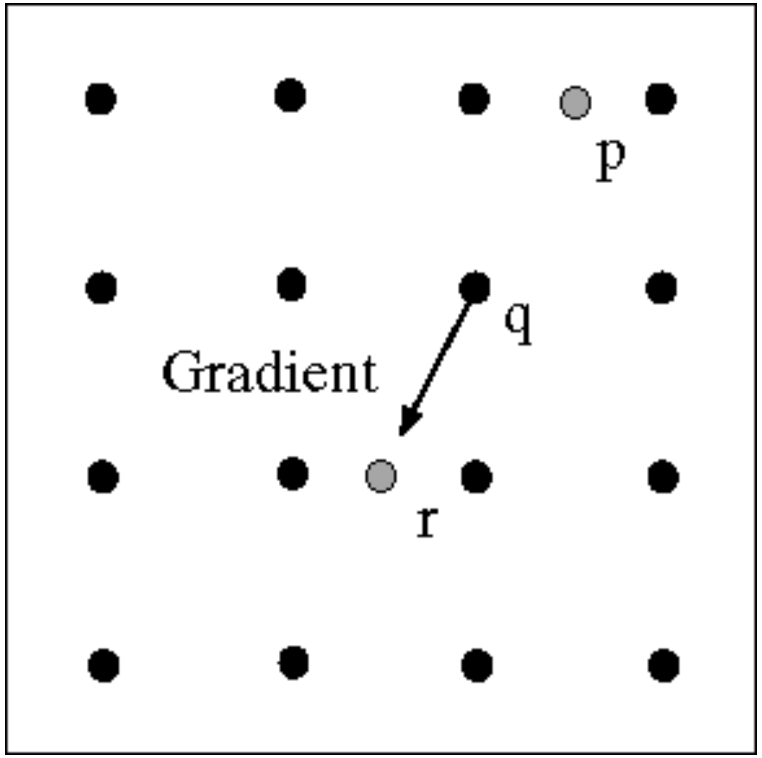
- Обрезанная (512x512) часть изображения с разворота Playboy, Nov 1972
- Самый популярный, но не первый случай использования Playboy в обработке изображений (первый в 1961)
- Пригласили на 50ую конференцию Society for Imaging Science and Technology (IS&T) in 1997

Пример

Норма градиента

Поиск локальных максимумов





Несколько вариантов фильтров:

- Фильтр 3x3. Пиксель оставляем, если градиент в нём больше всех остальных в окрестности.
- Фильтр по направлению градиента. Оставляем градиент q, если значение больше p и r. Значения в p и r интерполируем.

Source: D. Forsyth

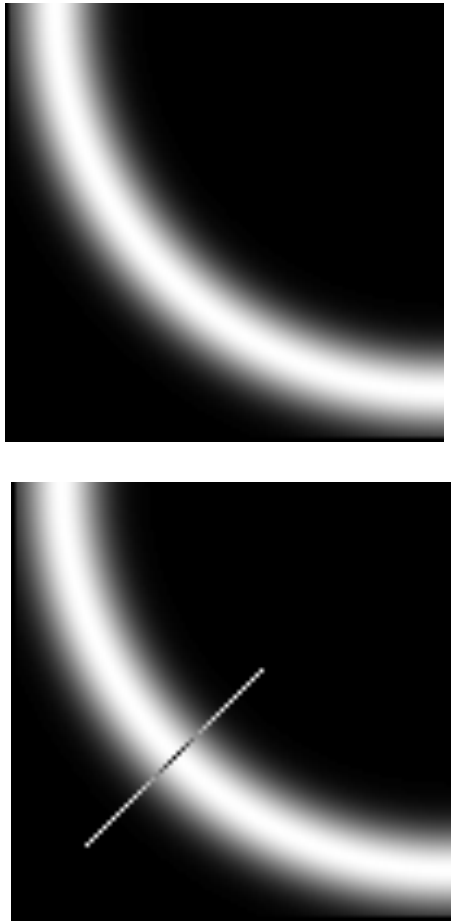
Результат работы

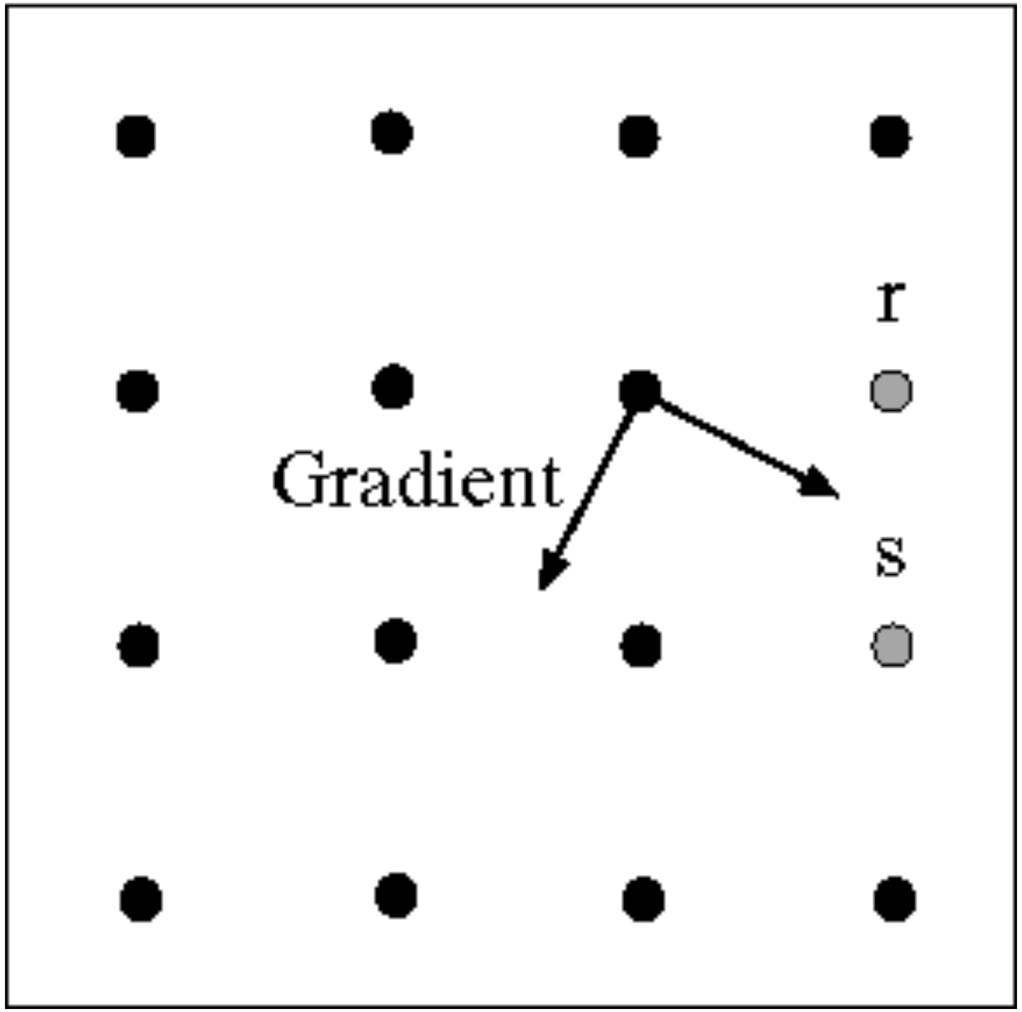
Прослеживание с гистерезисом

- Выбираем два порога «верхний» и «нижний»
- Размечаем все пиксели изображений (оставшиеся)
  - Выше верхнего – «сильные края»
  - Выше нижнего – «слабые края»
  - Ниже нижнего - шум
- Идея прослеживания:
  - Оставляем «слабые» края только в том случае, если они связаны с сильными краями

Source: S. Seitz

Прослеживание границ





- Пусть отмеченная точка – «слабый край», который ещё не проверен.
- Будем искать «сильного» соседа, переходя по соседним неподавленным краям вдоль границы

Source: D. Forsyth

Эффект гистерезиса

Исходное изображение

Высокий порог
(сильные края)

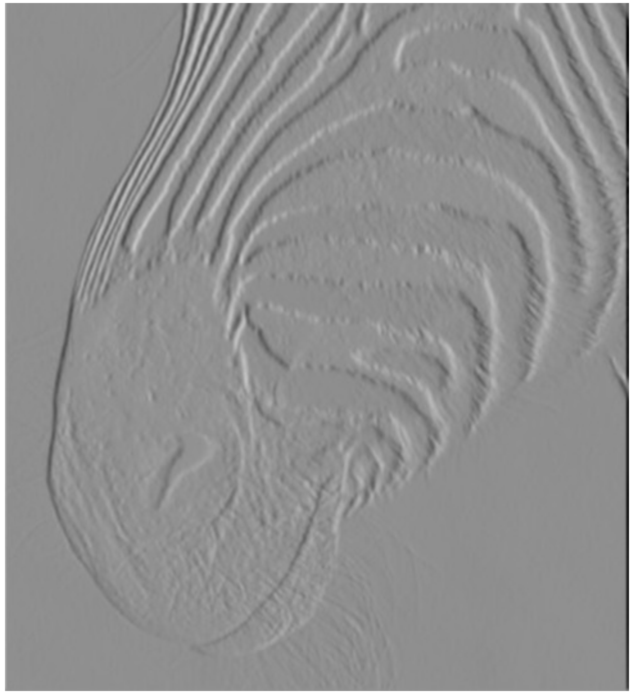
Низкий порог
(слабые + сильные края)

Порог по гистерезису

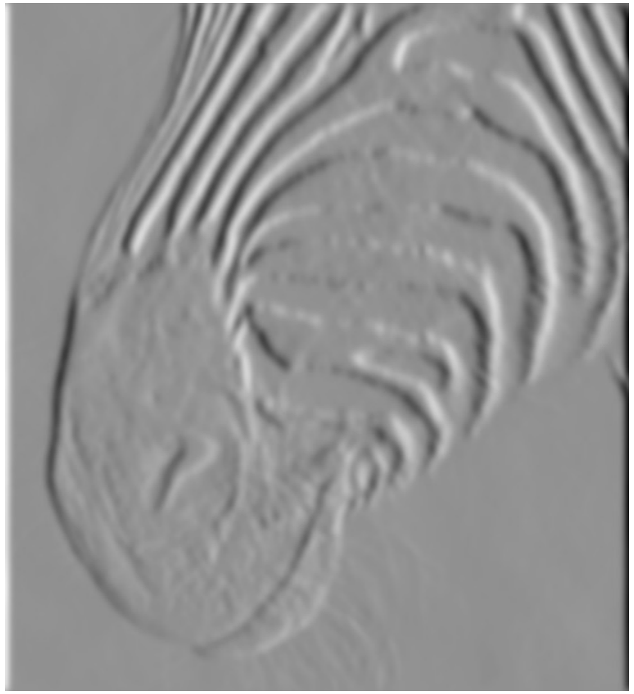
Source: L. Fei-Fei

Сглаживание и локализация

Применим сглаженные производные разного размера:



1 pixel



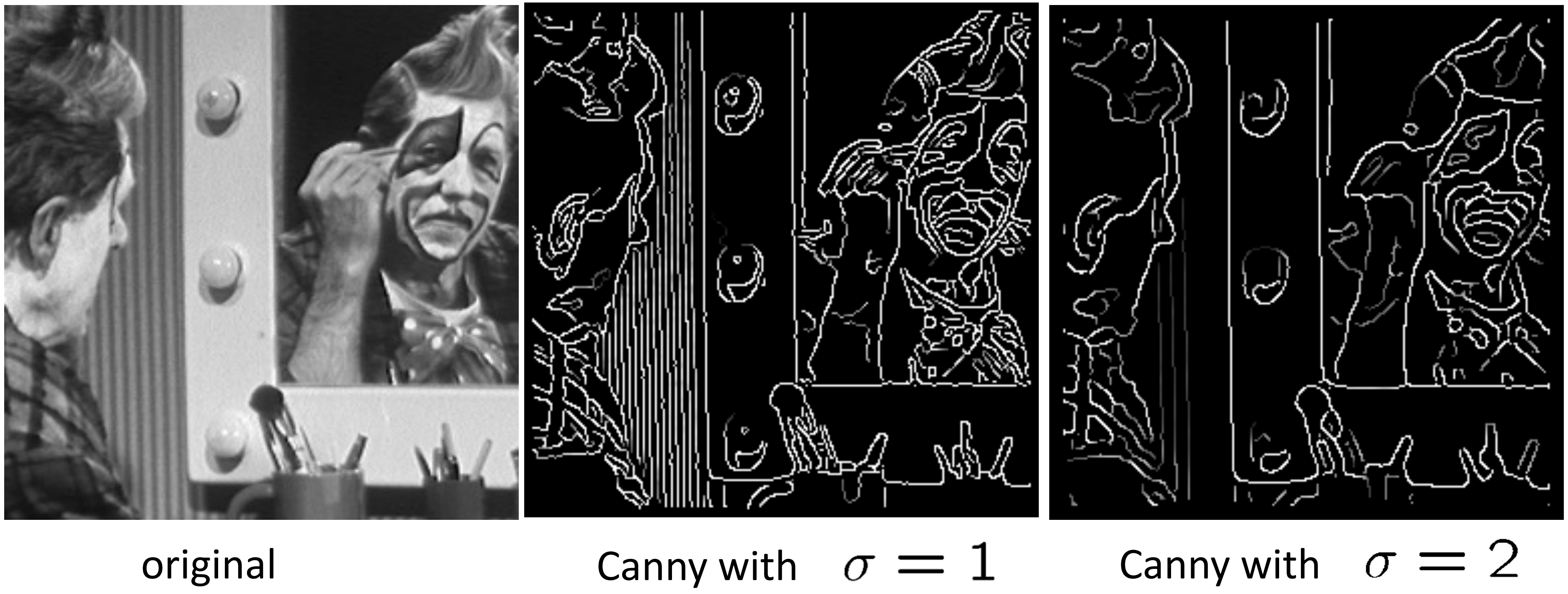
3 pixels

7 pixels

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

Source: D. Forsyth

Влияние σ в Cany (Размер ядра размытия)



Выбор σ зависит от задачи

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

Source: S. Seitz

На лекции рассмотрели

- Коррекцию яркости и цветопередачи
- Линейную фильтрацию (свёртку) изображения, которая позволяет решать целый ряд задач – шумоподавление, оценка градиента, оценки карты освещённости
- Медианный фильтр для подавления импульсного шума
- Быстрая реализация фильтров
- Выделение краёв изображения и метод Canny