



# Основы обработки изображений

Яндекс



## Sonnet for Lena

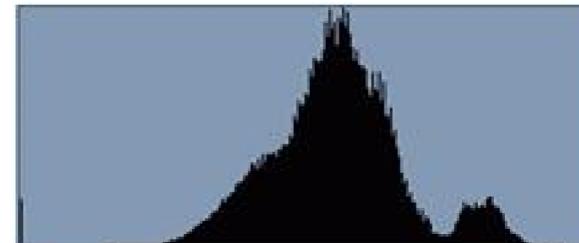
O dear Lena, your beauty is so vast  
It is hard sometimes to describe it fast.  
I thought the entire world I would impress  
If only your portrait I could compress.  
Alas! First when I tried to use VQ  
I found that your cheeks belong to only you.  
Your silky hair contains a thousand lines  
Hard to match with sums of discrete cosine.  
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 locks 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 cosine.  
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 locks here or there  
But when filters took sparkle from your eyes  
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst



Антон Конушин



# Обработка изображений

---

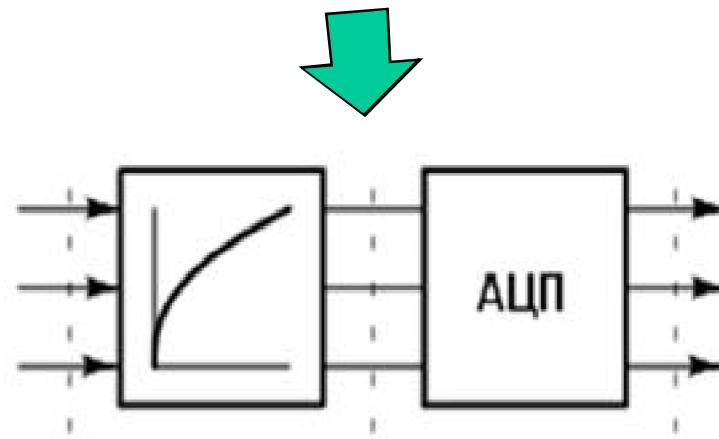
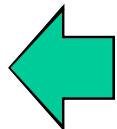
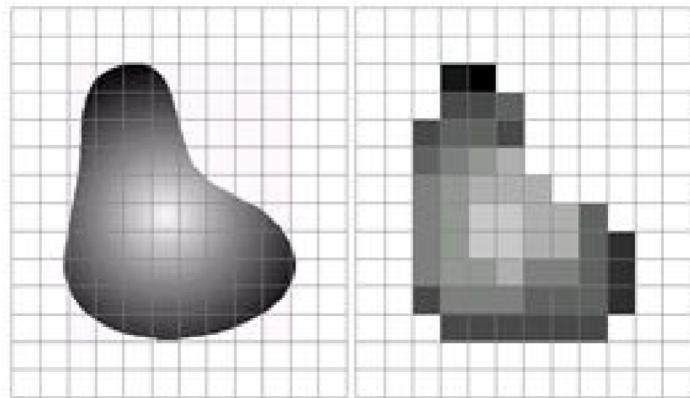
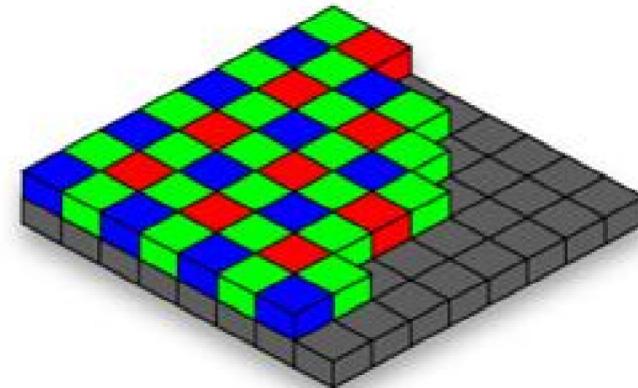
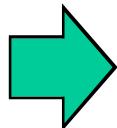


- Семейство методов и задач, где входной и выходной информацией являются изображения.
  
- Зачем обрабатывать?
  1. Улучшение изображения для восприятия человеком
    - цель – чтобы стало «лучше» с субъективной точки зрения человека
  2. Улучшение изображения для восприятия компьютером
    - цель – упрощение последующего распознавания
  3. Развлечение (спецэффекты)
    - цель – получить эстетическое удовольствие от красивого эффекта



# Цифровое изображение

Яндекс



Вспоминаем процесс получения цифрового изображения...



# Что может получиться плохо?

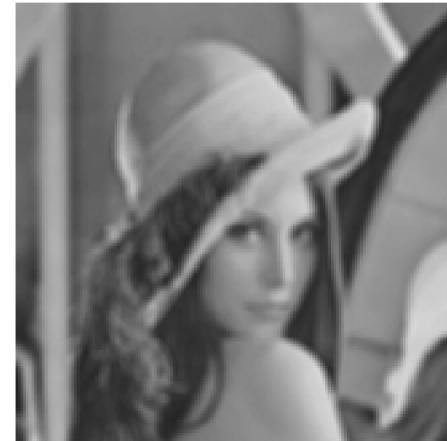
Яндекс



Темное или слабоконтрастное

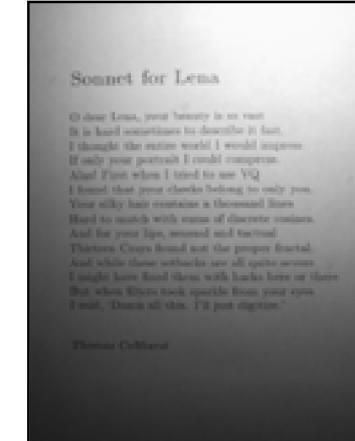


Неправильные цвета



Шумное

Нерезкое



Неравномерно освещённое



# Постоянство цвета и освещенности

Яндекс



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

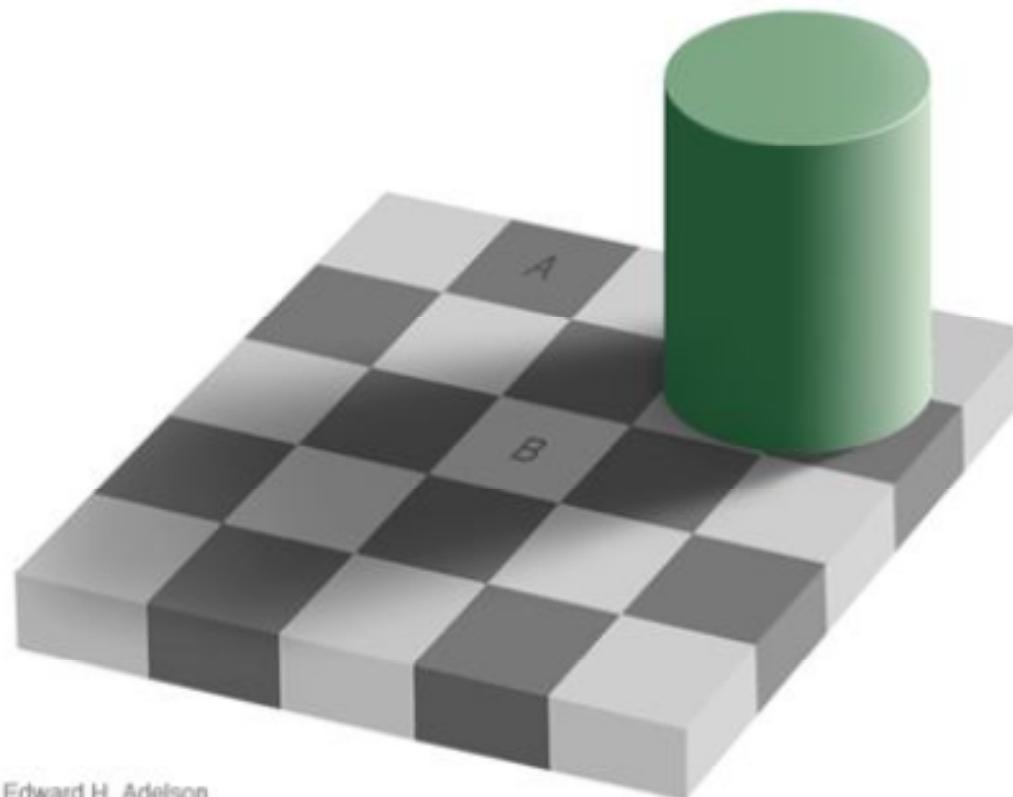
Пример:  
Белый цвет на  
свету и в тени

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



# Постоянство яркости

Яндекс



Edward H. Adelson

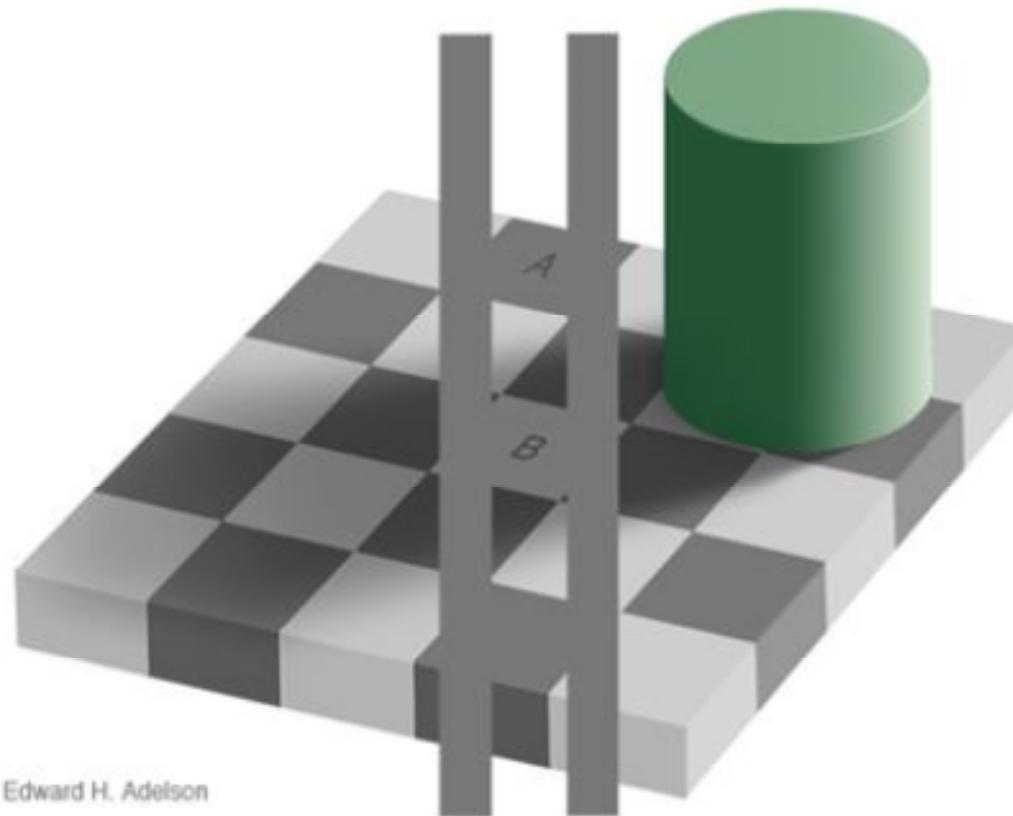
[http://web.mit.edu/persci/people/adelson/checkershadow\\_illusion.html](http://web.mit.edu/persci/people/adelson/checkershadow_illusion.html)

Slide by S. Lazebnik



# Постоянство яркости

Яндекс



Edward H. Adelson

[http://web.mit.edu/persci/people/adelson/checkershadow\\_illusion.html](http://web.mit.edu/persci/people/adelson/checkershadow_illusion.html)

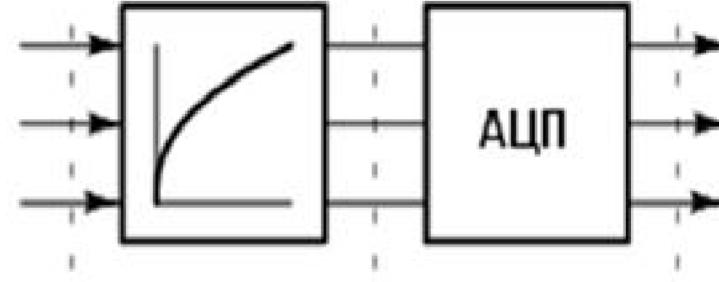
Slide by S. Lazebnik



## Почему может получиться плохо?

Яндекс

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





# Что такое гистограмма?

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



0 255



0 255



# Изменение контраста изображения

---

Что может не устраивать в полученном изображении:

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

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

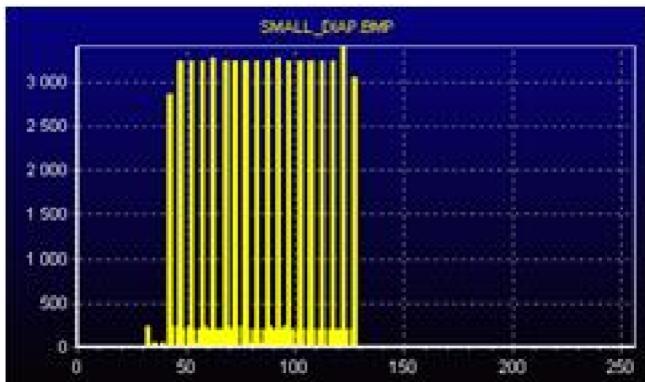
$$f^{-1}(y) = x \quad \begin{aligned} & y - \text{яркость пикселя на исходном изображении}, \\ & x - \text{яркость пикселя после коррекции}. \end{aligned}$$



# Линейная коррекция

Яндекс

Компенсация узкого диапазона яркостей –  
линейное растяжение:



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

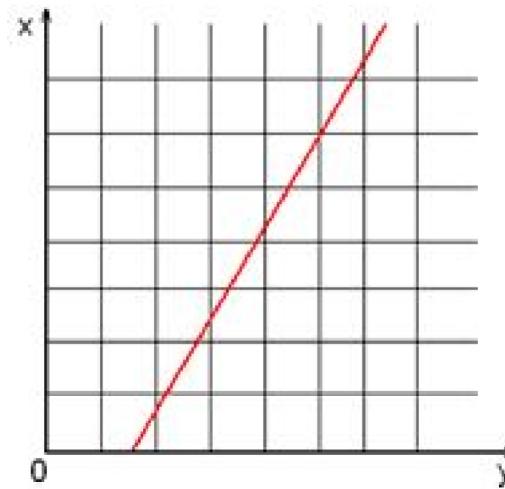


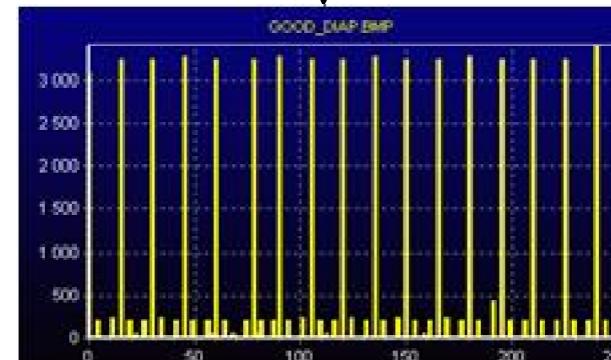
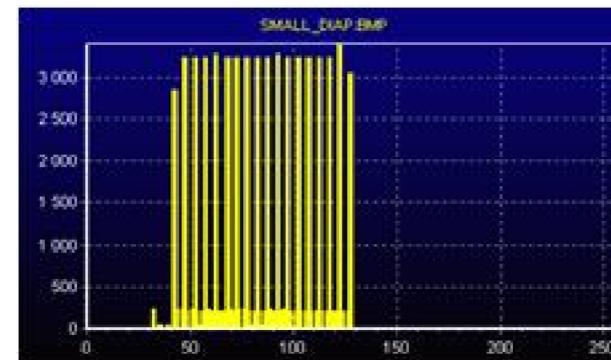
График функции  $f^{-1}(y)$



# Линейная коррекция

Яндекс

Компенсация узкого диапазона яркостей – линейное растяжение:





# 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

Яндекс

Линейное растяжение – «как AutoContrast в Photoshop»





## Линейная коррекция

Яндекс



Линейная коррекция помогает не всегда!



# Нелинейная коррекция

Яндекс

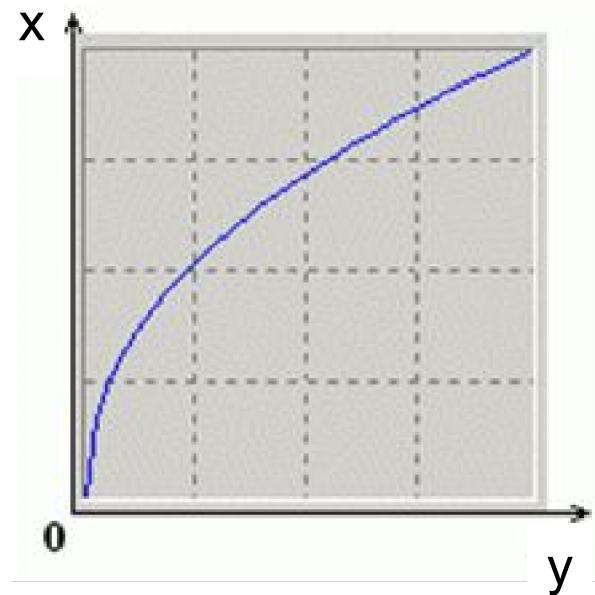


График функции  $f^{-1}(y)$



# Нелинейная коррекция

---

Нелинейная компенсация недостаточной контрастности

Часто применяемые функции:

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

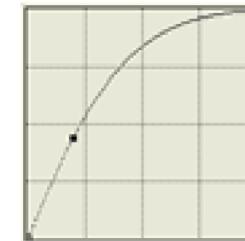
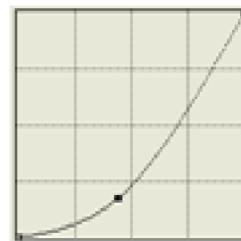
$$y = c \cdot x^\gamma$$

- Логарифмическая
  - Цель – сжатие динамического диапазона при визуализации данных

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



# Гамма-коррекция



Графики функции  $f^{-1}(y)$



# Нелинейная коррекция

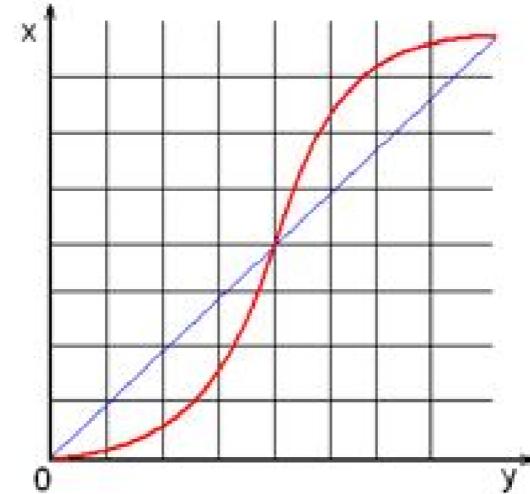
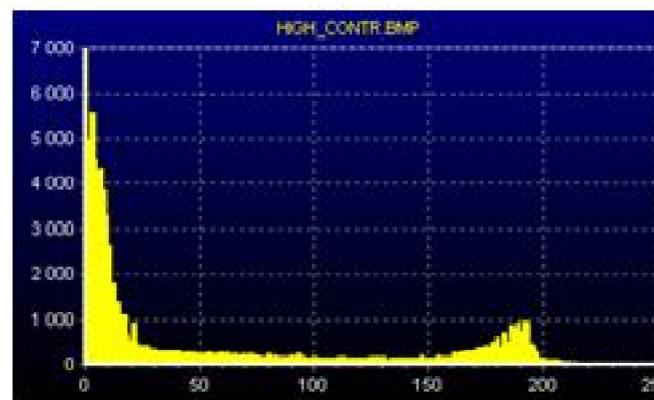
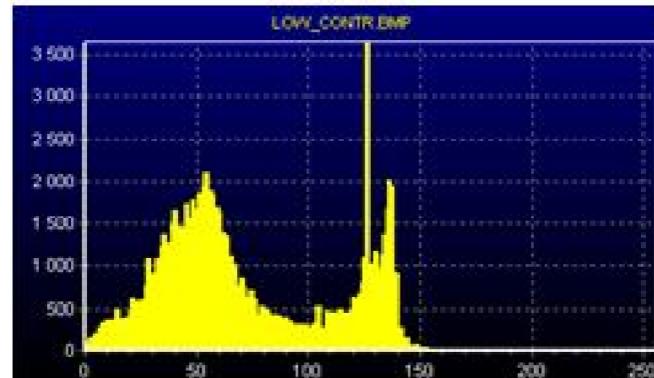


График функции  $f^{-1}(y)$



# Цветовой баланс («баланс белого»)

Яндекс

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

AWB	Auto White Balance
Custom	
K	Kelvin
	Tungsten
	Fluorescent
	Daylight
	Flash
	Cloudy
	Shade

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?



# 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

Яндекс





## «Серый мир» - примеры

Яндекс





# «Серый мир» - примеры

Яндекс





# 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 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



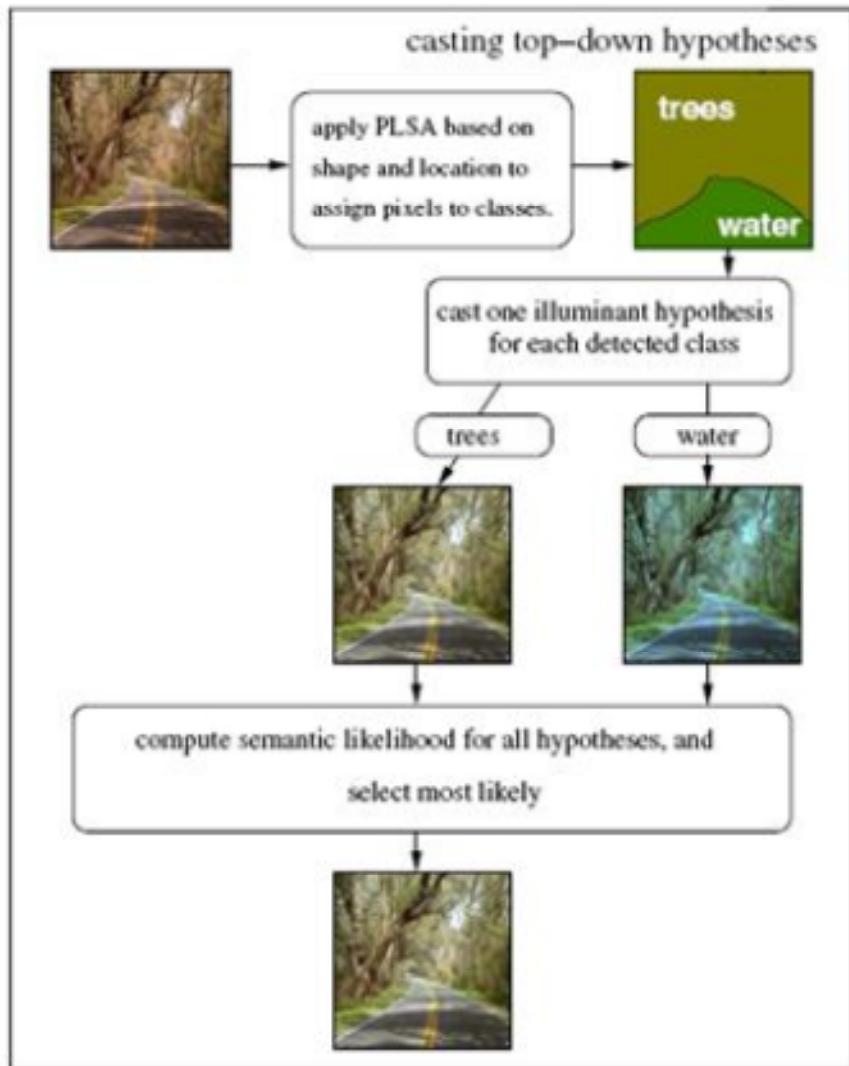


# Растяжение контрастности





# 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 «типовых» изображениях



# 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



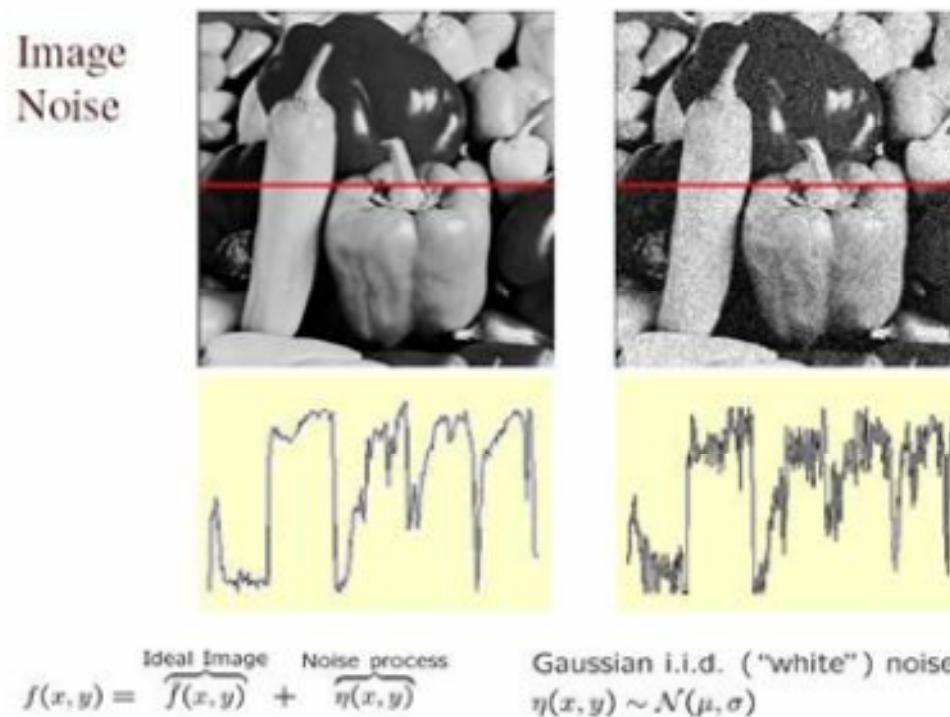
Gaussian noise

- **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



Source: M. Hebert

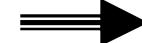


# Усреднение нескольких кадров

Яндекс



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:

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

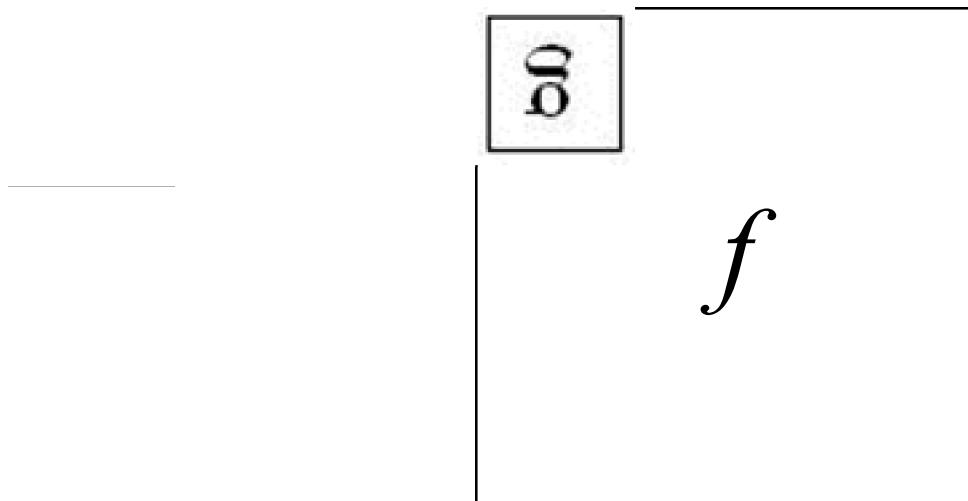
“box filter”



## Определение свертки

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

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



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



# Основные свойства

---

- **Линейность:**  $\text{filter}(f_1 + f_2) = \text{filter}(f_1) + \text{filter}(f_2)$
- **Инвариантность к сдвигу:** не зависит от сдвига 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

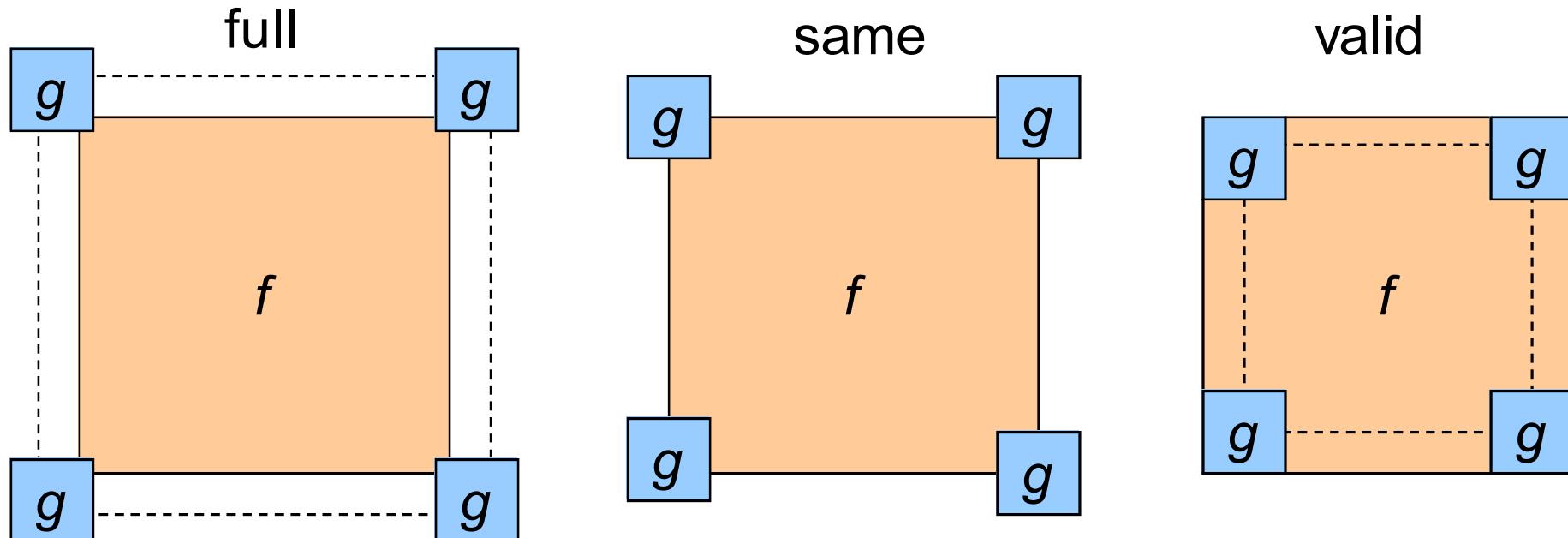
- Коммутативность:  $a * b = b * a$ 
  - Нет никакой разницы между изображением и ядром фильтра
- Ассоциативность:  $a * (b * c) = (a * b) * c$ 
  - Последовательное применение фильтров:  $((a * b_1) * b_2) * b_3$
  - Эквивалентно применению такого фильтра:  $a * (b_1 * b_2 * b_3)$
- Дистрибутивность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



# Детали реализации

Размер результирующего изображения?

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

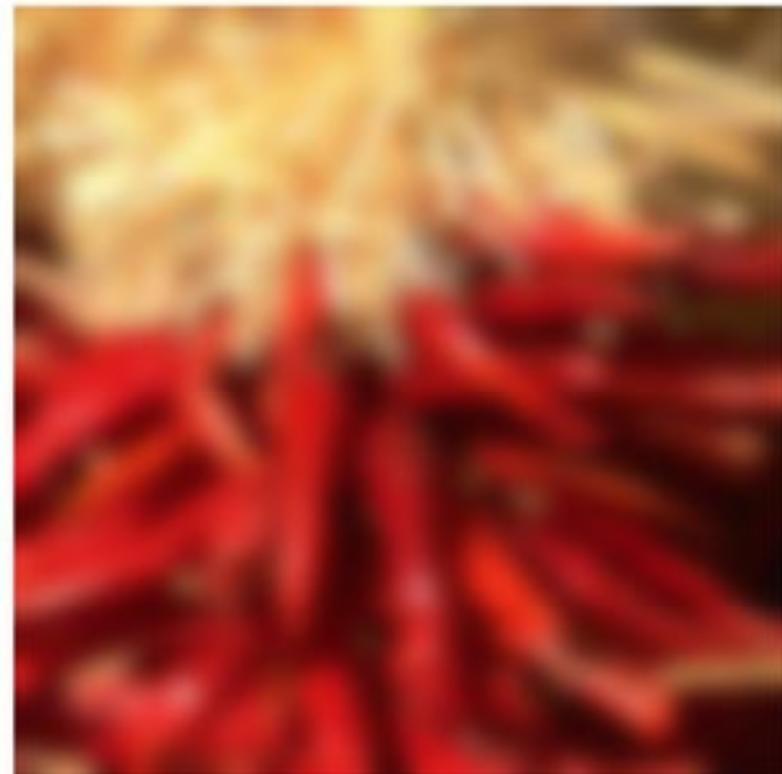




# Детали реализации

## Как происходит фильтрация по краям?

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



Source: S. Marschner



# Простейшие фильтры

Яндекс



Original

0	0	0
0	1	0
0	0	0

?



# Простейшие фильтры

Яндекс



Original

0	0	0
0	1	0
0	0	0



Filtered  
(no change)



# Простейшие фильтры

Яндекс



Original

0	0	0
0	0	1
0	0	0

?



# Простейшие фильтры

Яндекс



Original

0	0	0
0	0	1
0	0	0



Shifted left  
By 1 pixel



# Простейшие фильтры

Яндекс



Original

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

1	1	1
1	1	1
1	1	1

?



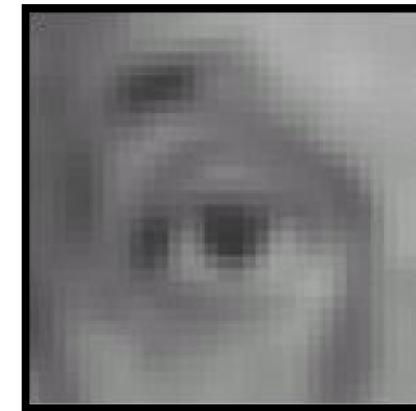
# Простейшие фильтры

Яндекс



Original

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

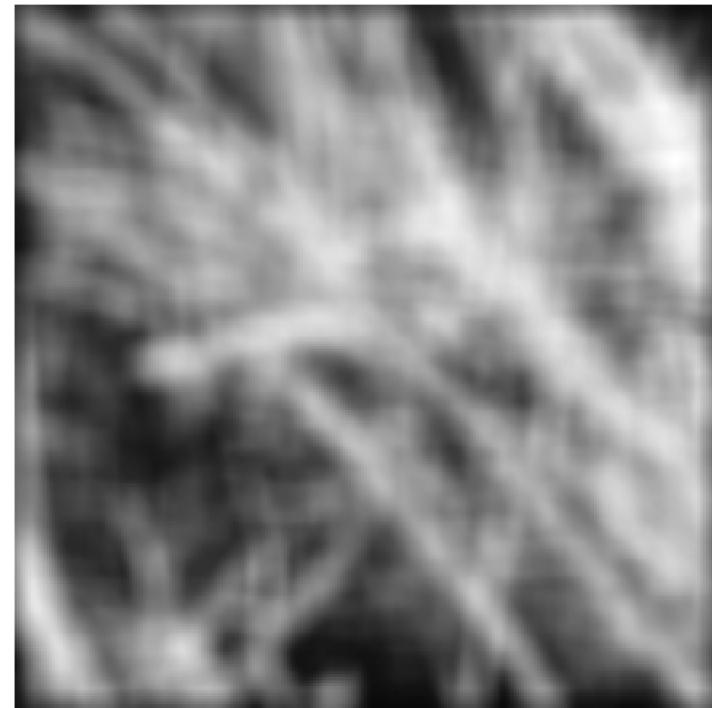


Blur (with a  
box filter)



# Сглаживание с box-фильтром

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

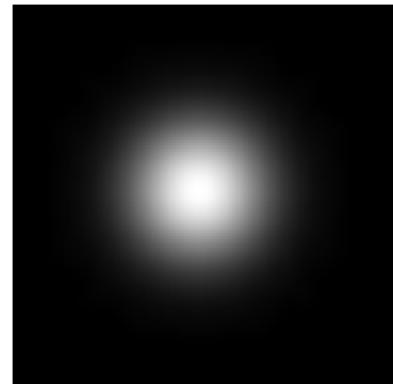
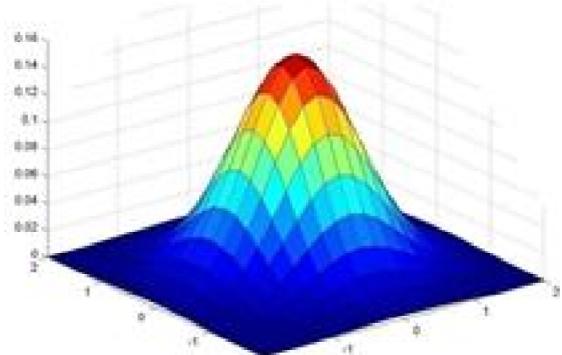




# Ядро фильтра гаусса

Яндекс

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



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

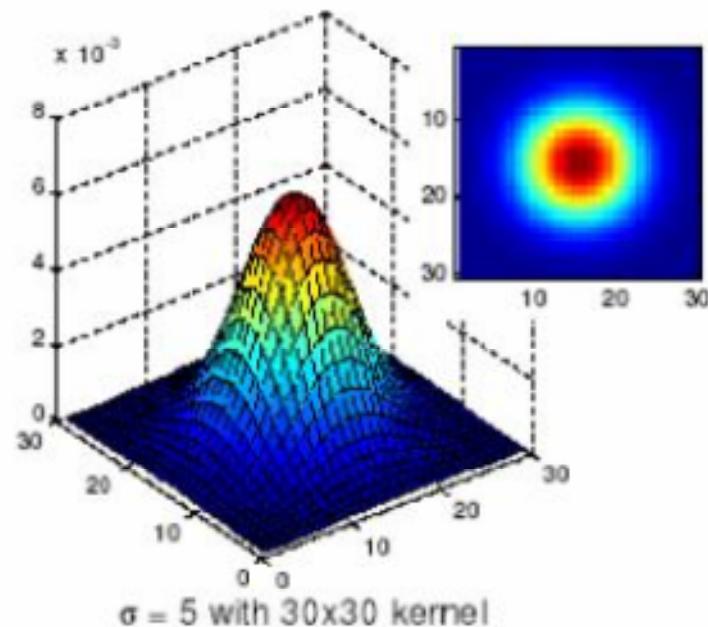
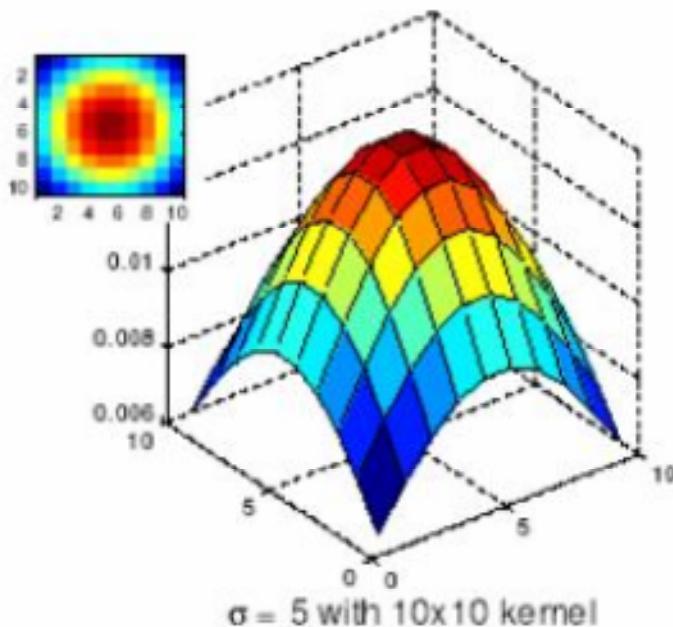
$5 \times 5, \sigma = 1$

Source: C. Rasmussen



# Выбор размера ядра

- Размер ядра дискретного фильтра ограничен

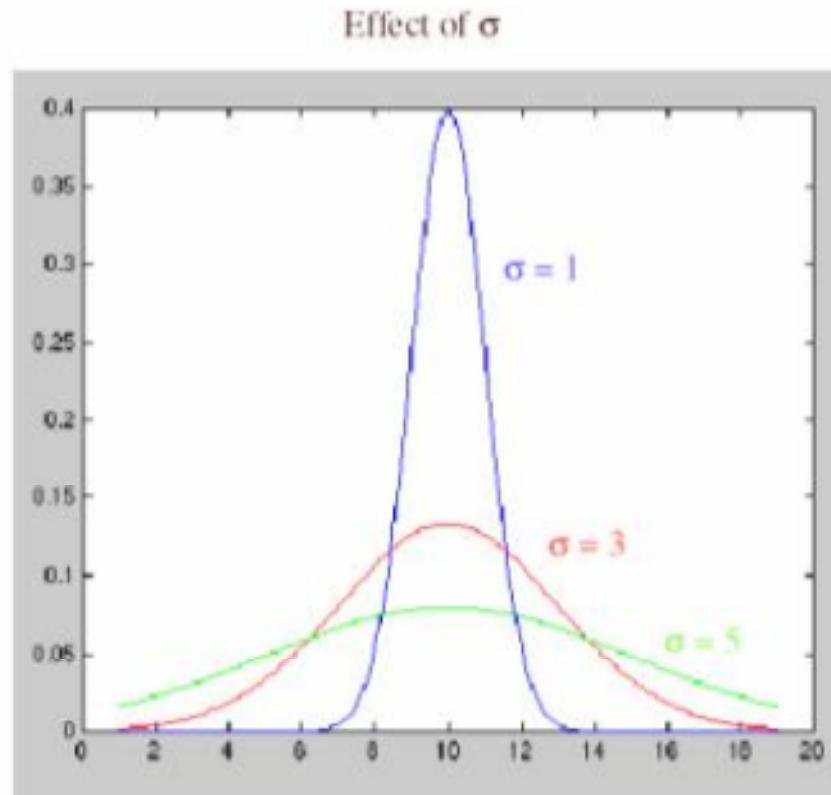




# Выбор размера ядра



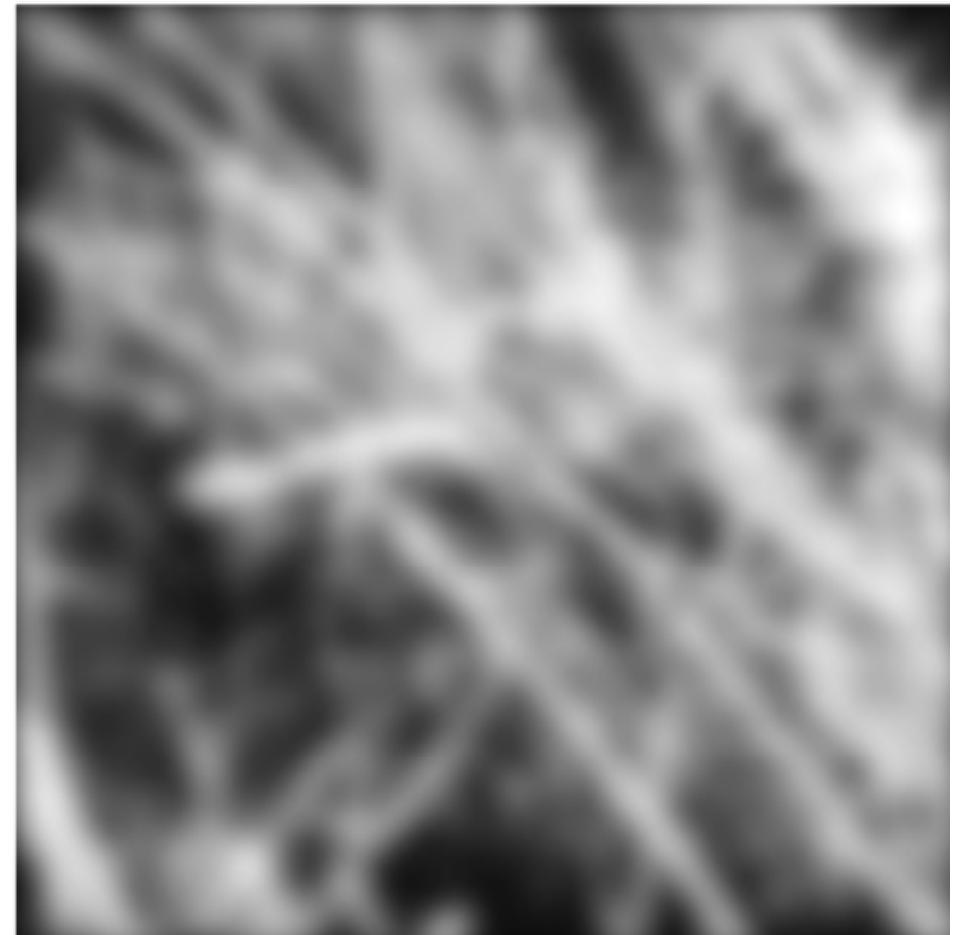
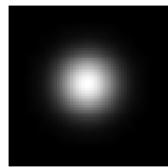
- Эмпирика: полуразмер фильтра равен  $3\sigma$





# Сглаживание фильтром гаусса

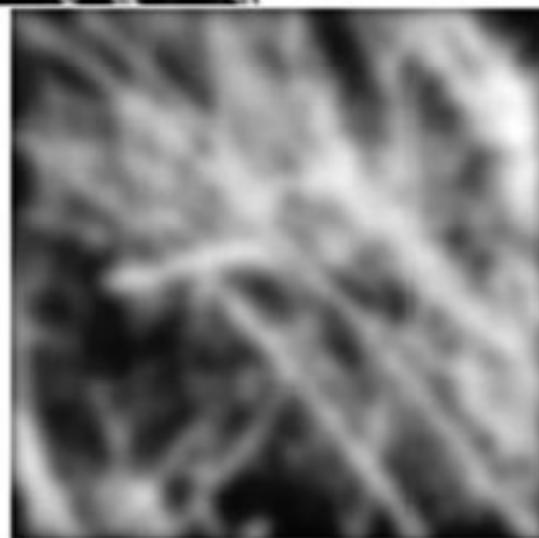
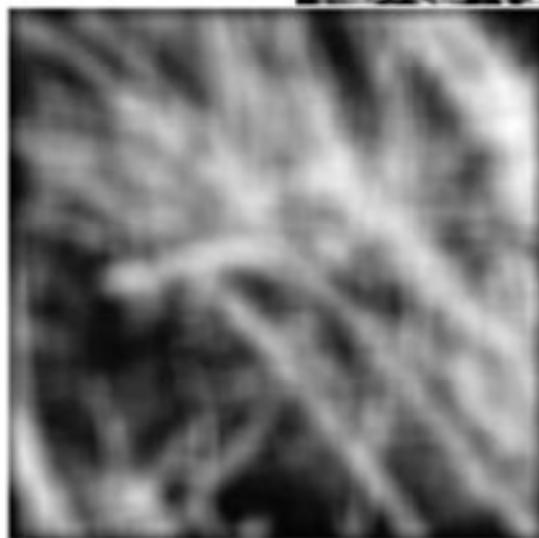
Яндекс





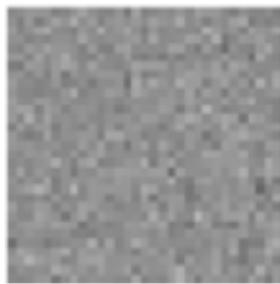
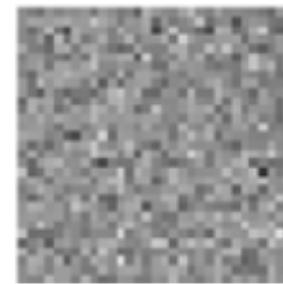
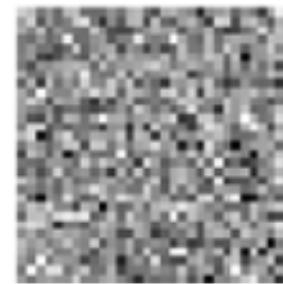
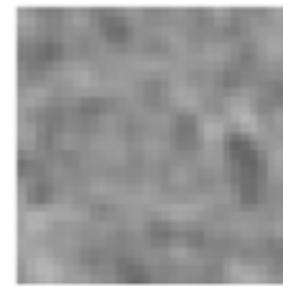
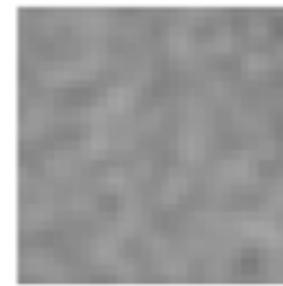
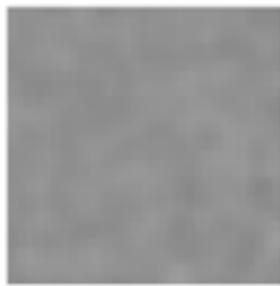
# Сравнение

Яндекс





# Подавление гауссова шума

 $\sigma=0.05$  $\sigma=0.1$  $\sigma=0.2$ no  
smoothing $\sigma=1 \text{ pixel}$  $\sigma=2 \text{ pixels}$ 

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  $\sigma$  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

Яндекс

2D свертка

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

Фильтр раскладывается  
в произведение двух 1D  
фильтров:

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

Свертка по строкам:

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

Затем свертка по столбцу:

Почему сепарабельность полезна на практике?

Source: K. Grauman



# Подавление шума «соль и перец»

Яндекс

3x3



5x5



7x7

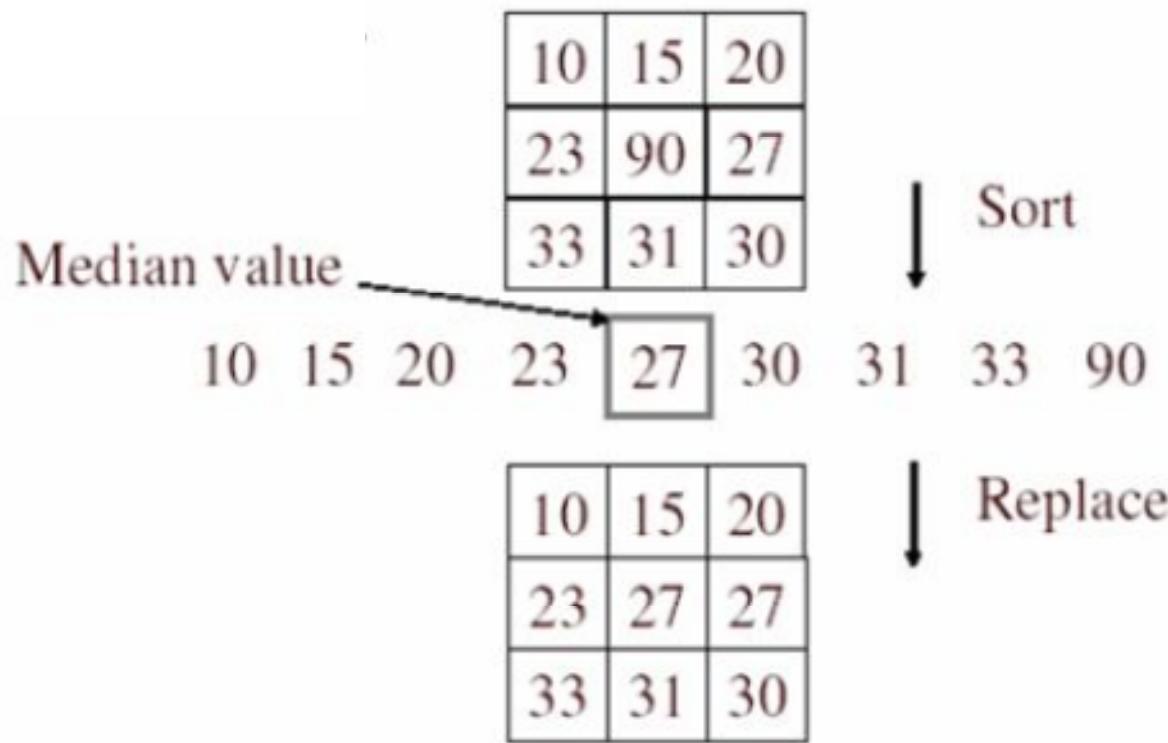


Применим фильтр Гаусса  
Чем результат плох?



# Медианный фильтр

- Выбор медианы из выборки пикселей по окрестности данного



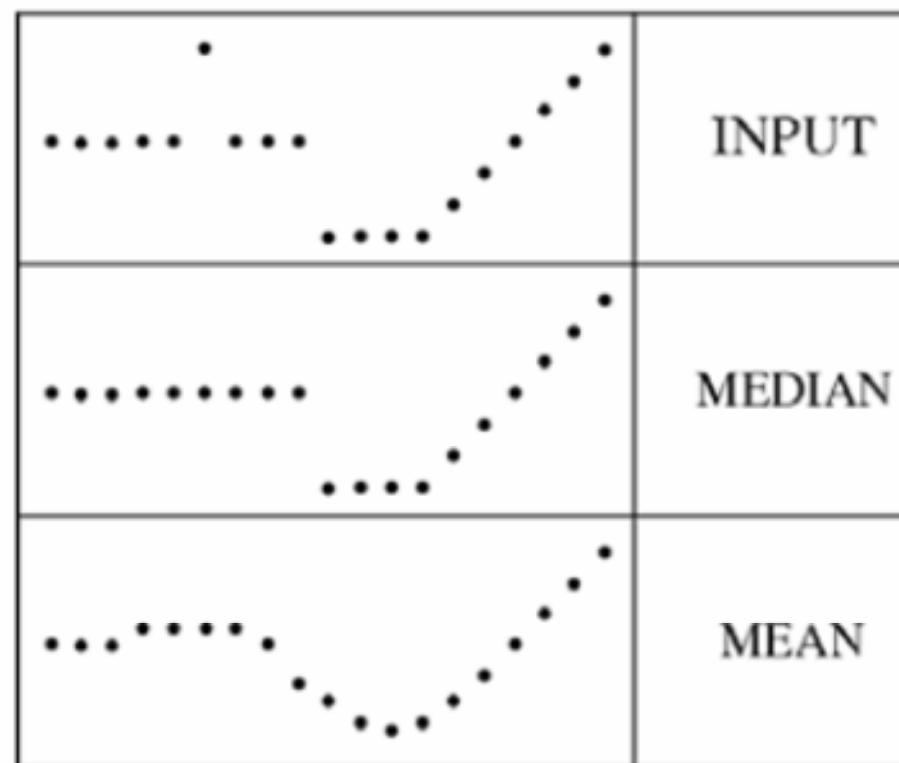
- Является ли фильтр линейным?



# Медианный фильтр

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

filters have width 5 :

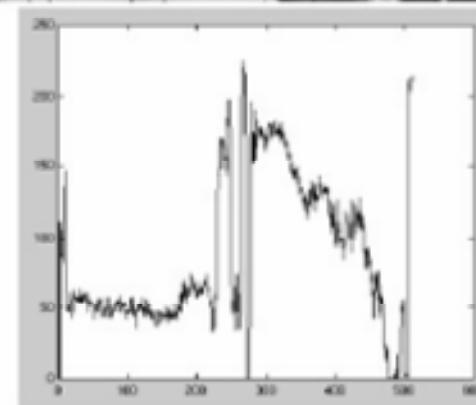
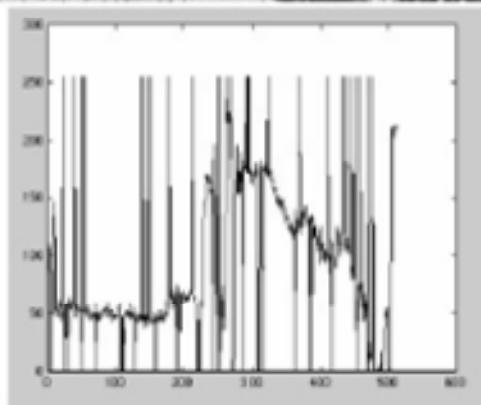


Source: K. Grauman



# Медианный фильтр

Шум «соль и перец» Медианная фильтрация



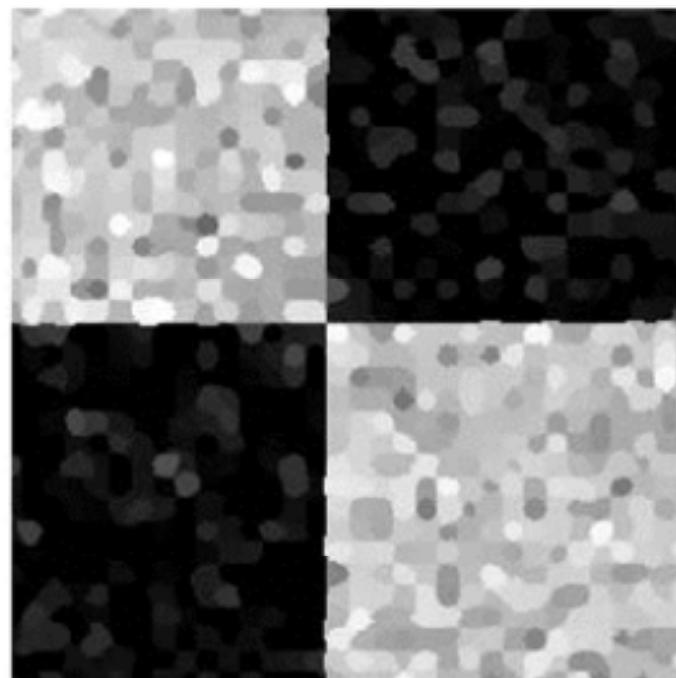
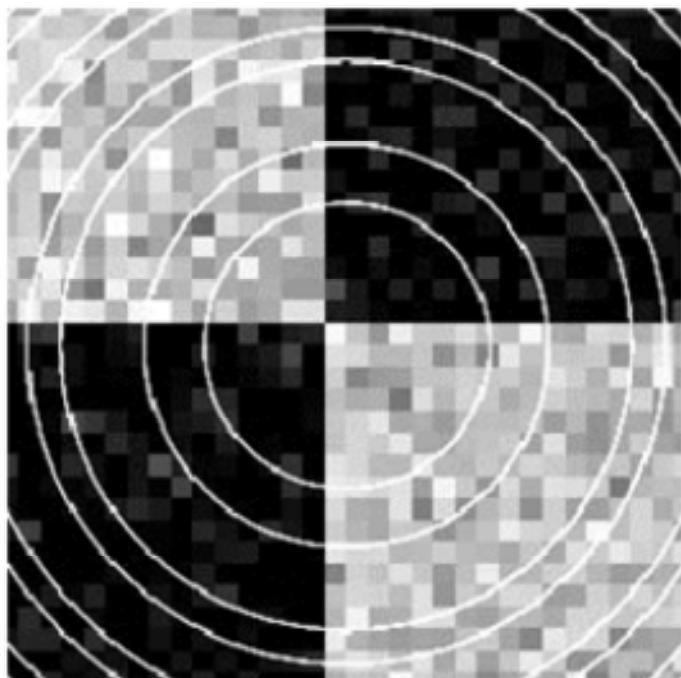
MATLAB: `medfilt2(image, [h w])`

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

Яндекс

3x3



5x5



7x7



Гауссов

Медианный





## Повышение резкости

---



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



# Повышение резкости

Яндекс

Что теряется при сглаживании?



-



=



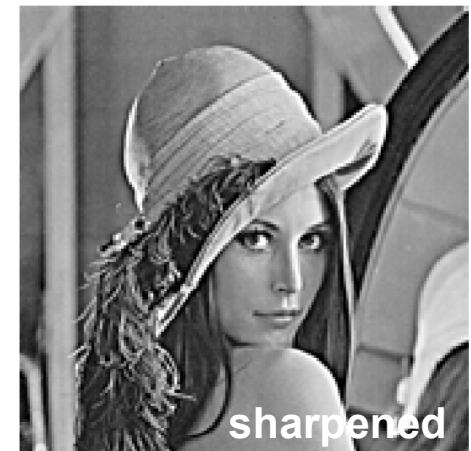
Добавим дополнительно высокие частоты:



+  $\alpha$



=





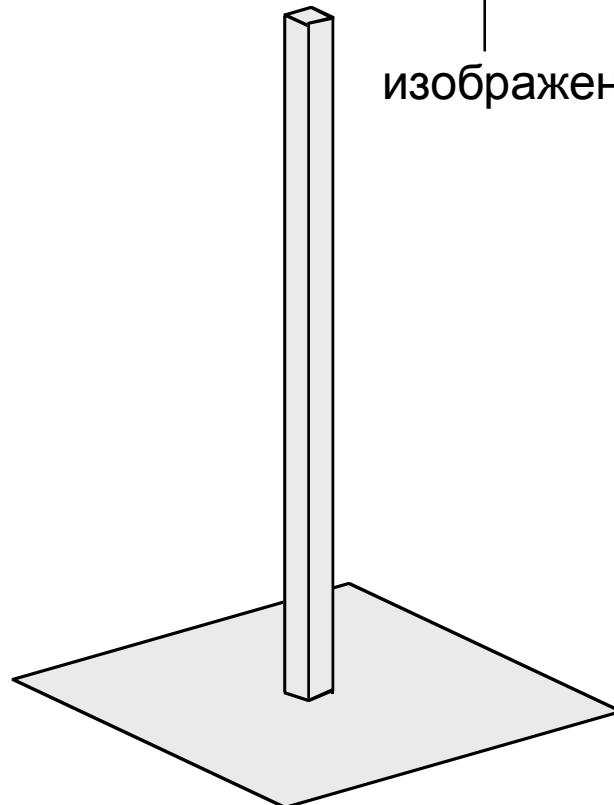
# Фильтр Unsharp

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

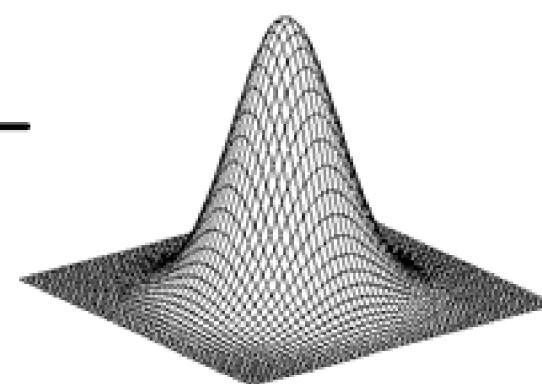
## изображение

## сглаженное изображение

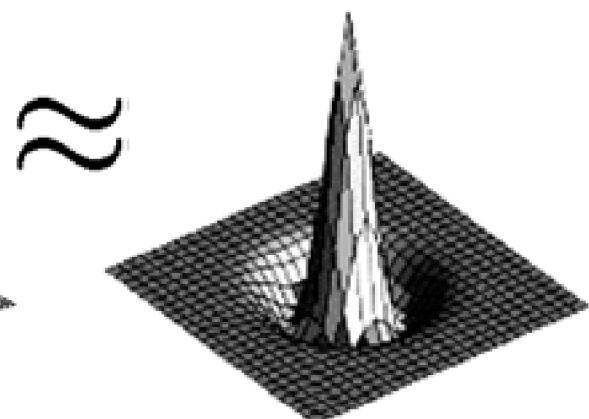
# Единичный Фильтр



## Единичный фильтр



# Гауссов



## Лапласиан гауссиана



# Пример повышения резкости

Яндекс

Ядро  
свертки

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

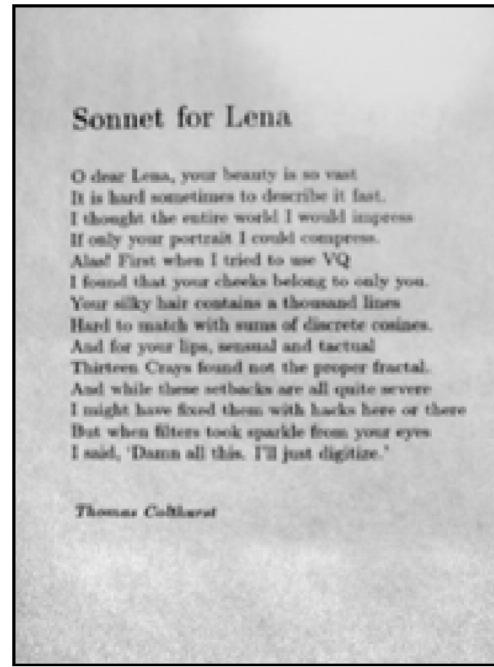
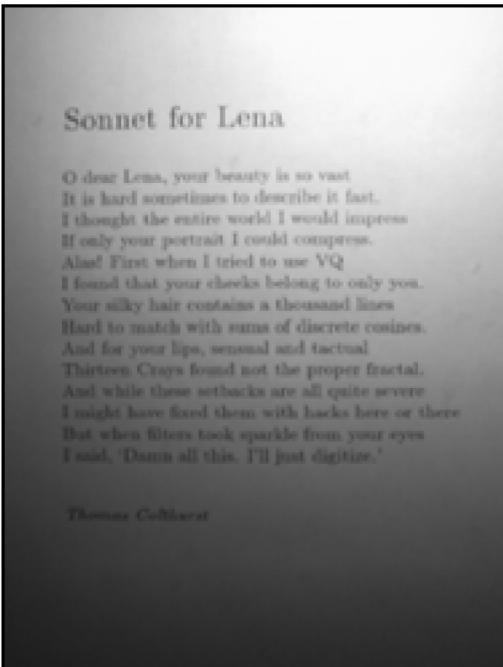




# Компенсация разности освещения

Яндекс

## Пример





# Компенсация разности освещения

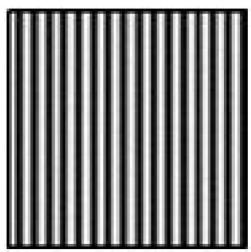
Яндекс

Идея:

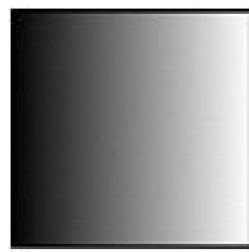
Формирование изображения:

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

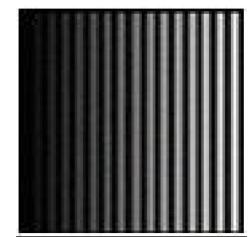
Плавные изменения яркости относятся к освещению,  
резкие - к объектам.



объект  $r(i, j)$



освещение  $l(i, j)$



Изображение  
освещенного  
объекта  $I(i, j)$