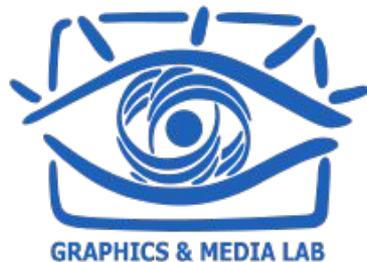


Компьютерная графика 2020



Лекция 1

Свет и цвет. Основы обработки изображений

Юрий Фетисов
Лаборатория компьютерной графики и мультимедиа

Области компьютерной графики

- Синтез изображений (компьютерная графика)
- Извлечение информации из изображений и видео (компьютерное зрение)
- Обработка изображений и видео

Области применения



Компьютерные игры



Видеоаналитика



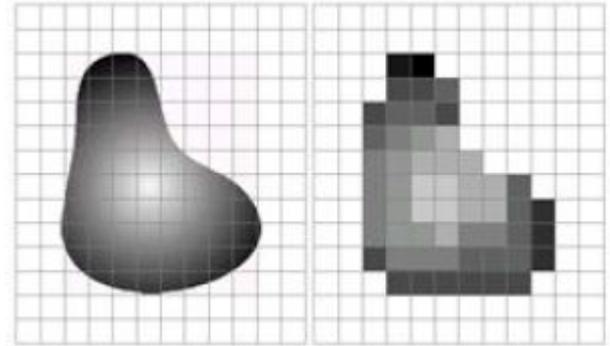
Дополненная реальность



Робототехника

Цифровое изображение

- Цифровое полутоновое изображение – матрица размера $N \times M$, элементами которой являются значения яркости света, измеренного на 2-мерной прямоугольной сетке
- Размер одного значения обычно 1 байт, т.е. $\in [0, 255]$
- Сейчас иногда используется большая точность (10-16 бит)



Что такое цвет

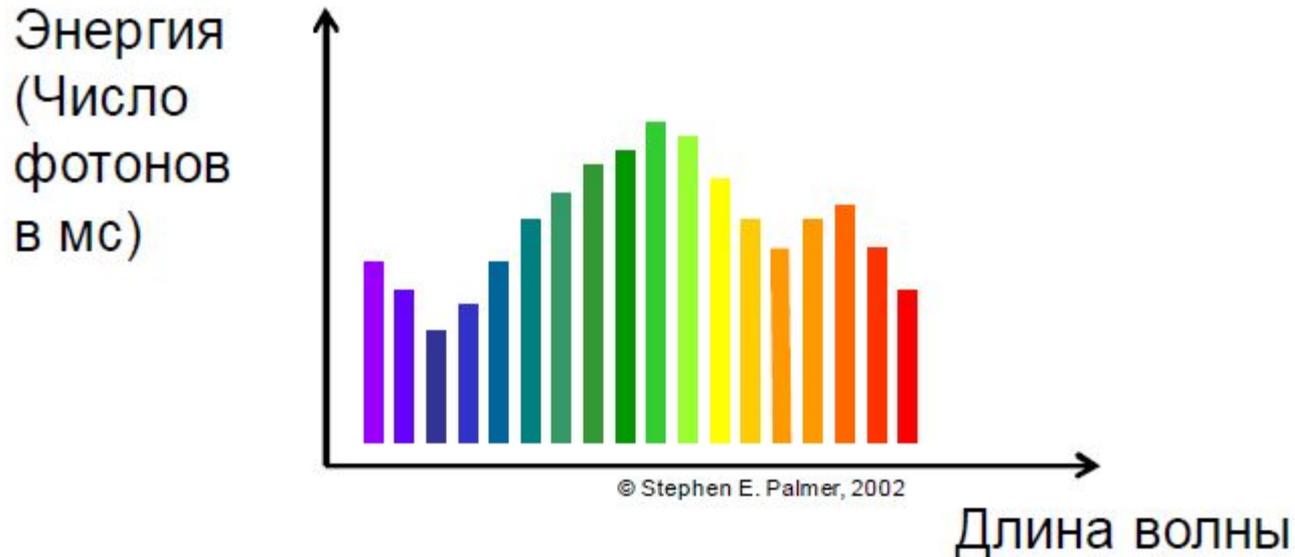
- Цвет – это психологическое свойство нашего зрения, возникающее при наблюдении объектов и света, а не физические свойства объектов и света
- Цвет – это результат взаимодействия света, сцены и нашей зрительной системы



Восприятие света человеком изучают науки фотометрия и колориметрия

Физика света

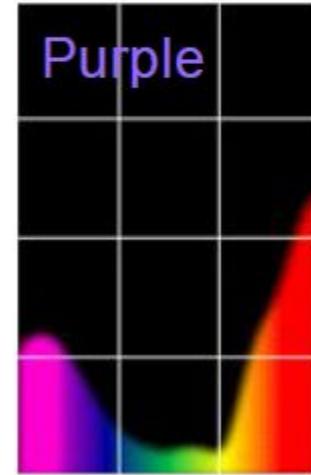
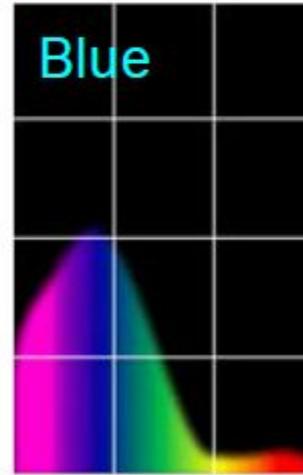
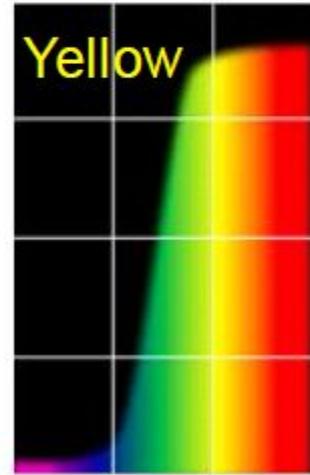
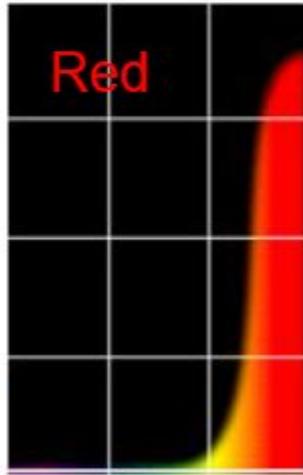
Любой источник света можно описать спектром: количество излученной энергии в единицу времени для каждой длины волны в интервале 380-720 нм



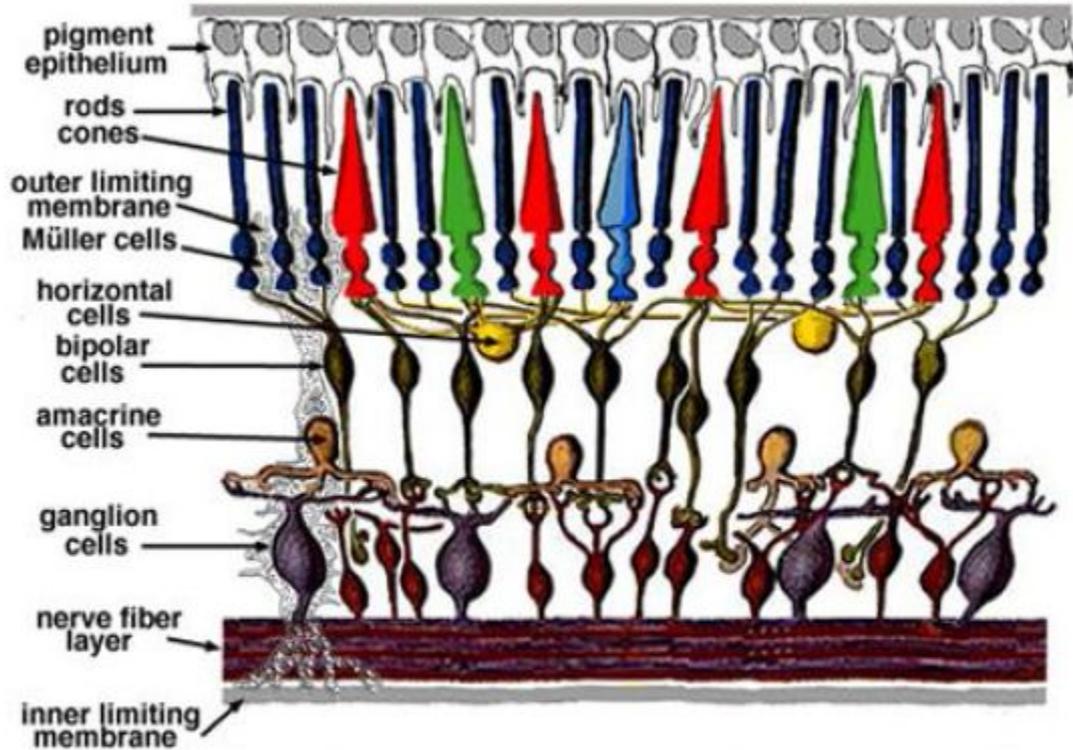
Физика света



% Отраженного света



Сетчатка глаза

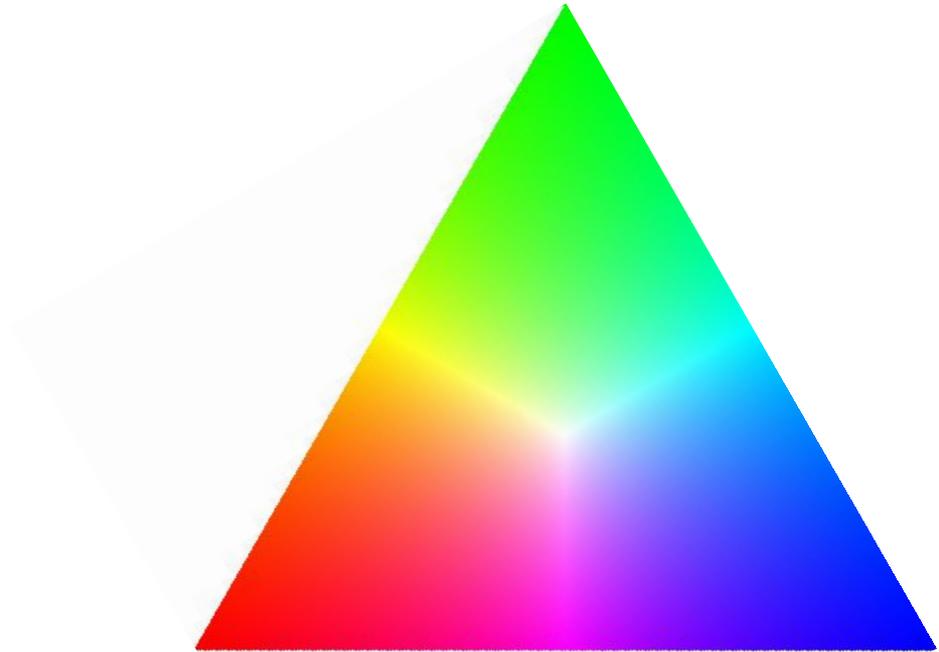
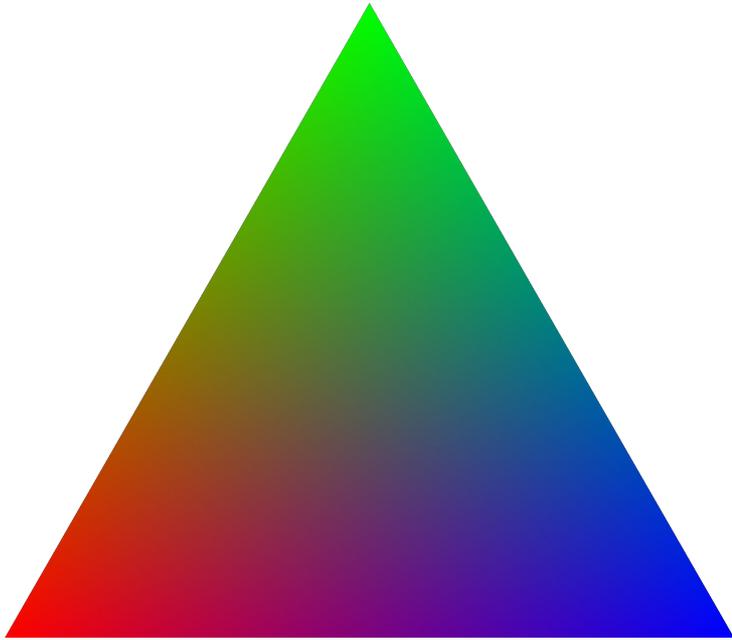


- Палочки (Rods) измеряют яркость
- Колбочки (Cones) измеряют цвет

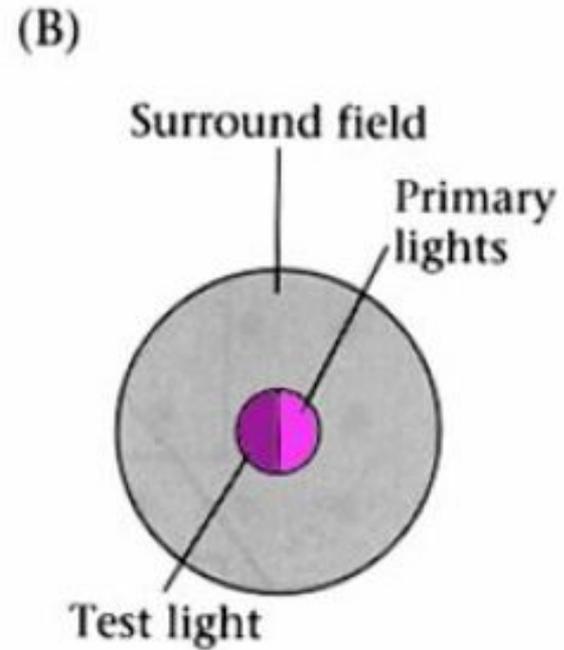
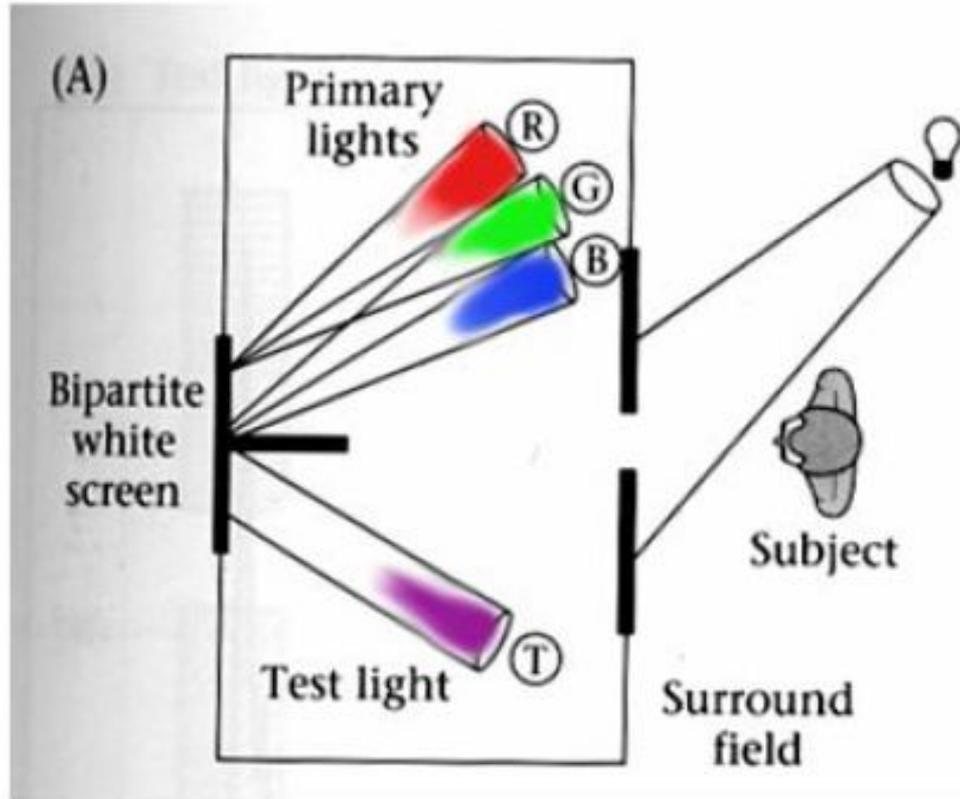


Трихроматическая теория

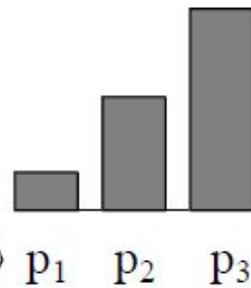
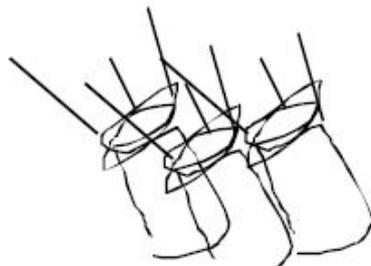
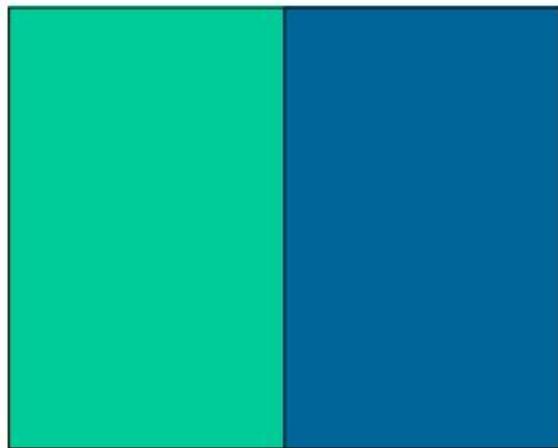
Трихроматическая теория гласит, что любой видимый цвет можно получить комбинацией трех “базовых” цветов (A, B, C). $C = aA + bB + cC$.



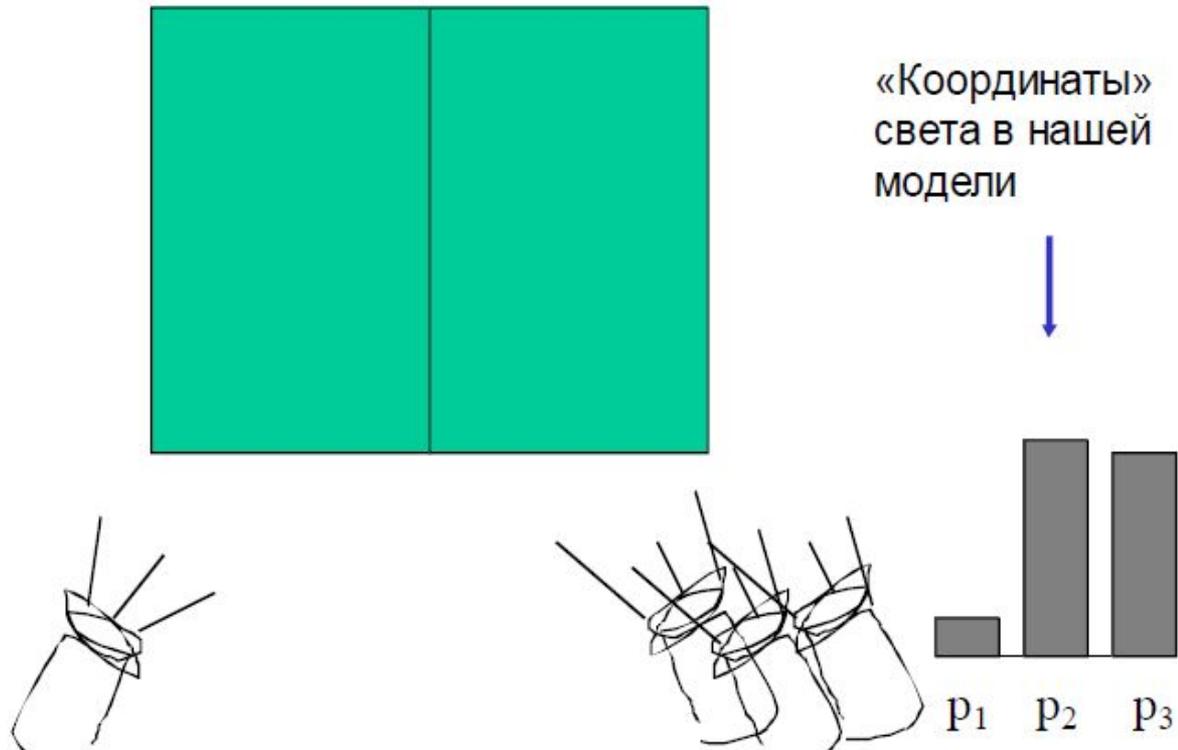
Сопоставление цветов



Сопоставление цветов



Сопоставление цветов



Закон аддитивности Грассмана

- Если наблюдатель задаст цвет лучей 1 и 2 как (R_1, G_1, B_1) и (R_2, G_2, B_2) , тогда если мы сложим источники 1 и 2, то мы их можем воспроизвести как $(R_1+R_2, G_1+G_2, B_1+B_2)$
- Соответствие цветов выполняется на всех уровнях яркости. Т.е. если $C_1 = (R_1, G_1, B_1)$, то $kC_1 = (kR_1, kG_1, kB_1)$

Модели цвета, основанные на выборе базовых цветов и их линейных комбинациях – линейные цветовые модели

Модель CIE RGB 1931

 $p_1 = 645.2 \text{ nm}$

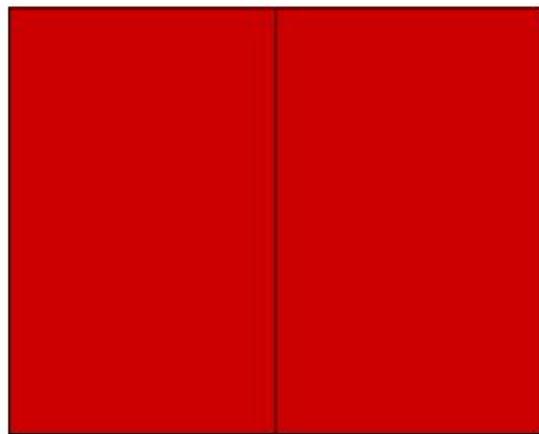
 $p_2 = 525.3 \text{ nm}$

 $p_3 = 444.4 \text{ nm}$

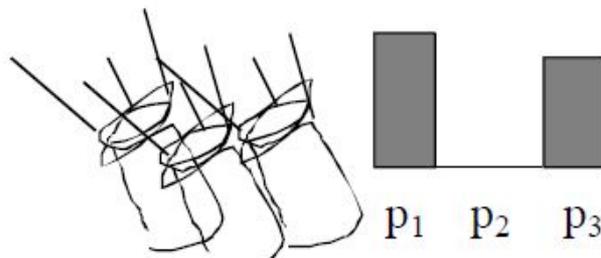
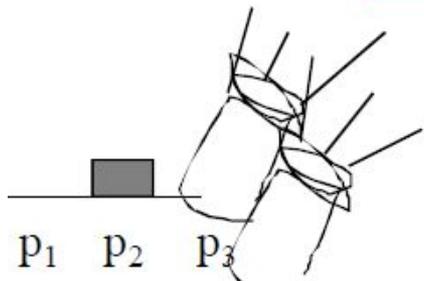
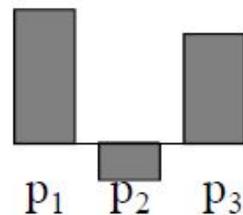
Подходят ли эти три цвета для описания всех видимых цветов?

Сопоставление цветов

Мы называем m «отрицательным» весом основного цвета, если цвет нужно добавлять к сопоставляемому свету.



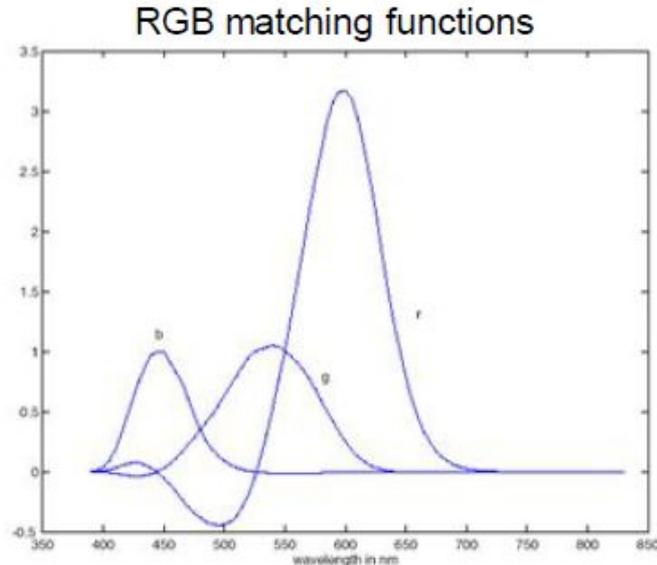
Веса основных цветов, необходимых для сопоставления:



Модель CIE RGB 1931

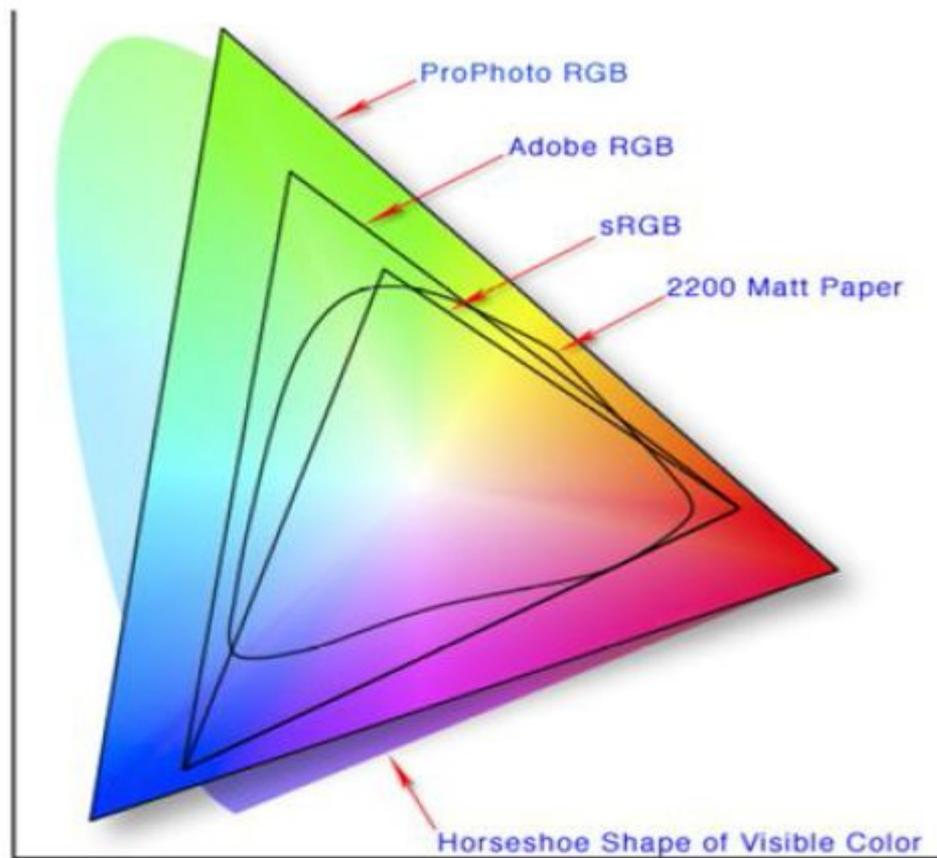
-  $p_1 = 645.2 \text{ nm}$
-  $p_2 = 525.3 \text{ nm}$
-  $p_3 = 444.4 \text{ nm}$

Модель CIE RGB 1931
не полностью
описывает всё
множество видимых
цветов



«Вычитание» необходимо
для соответствия некоторым
длинам волны

Различные модели RGB



Яркость и цветность

Визуально можно выделить две характеристики цвета:

- Яркость (brightness)
- Цветность (chromacity)
 - Тон (hue)
 - Насыщенность (saturation)

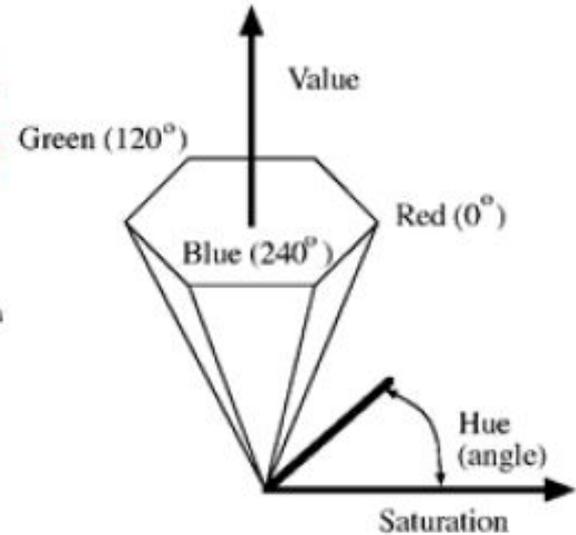
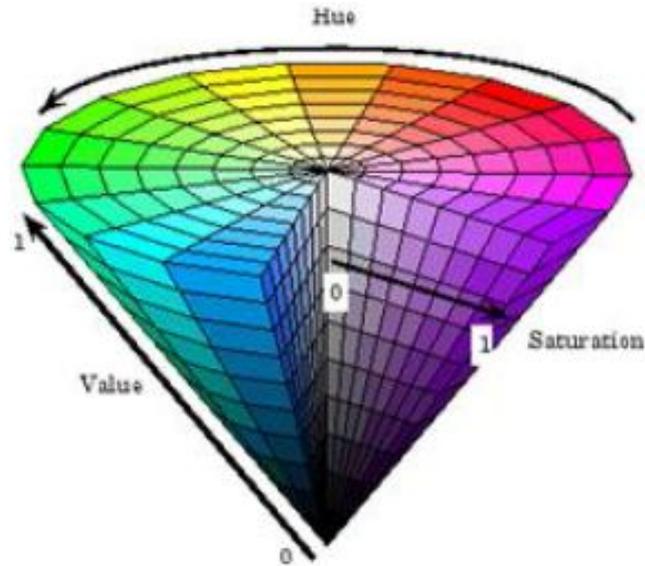
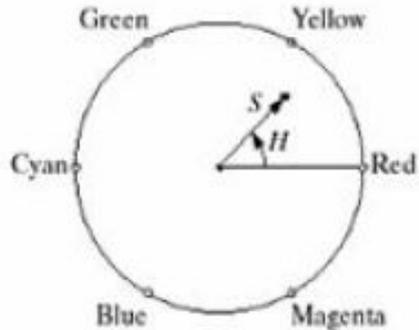
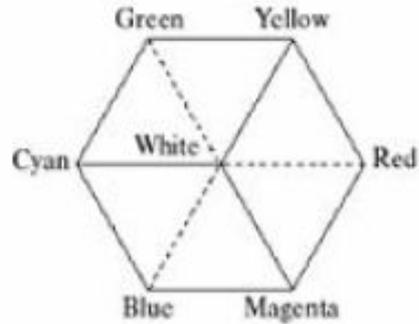
В RGB нет таких понятий, что бывает не всегда удобно

$$Y = .299R + .587G + .114B$$

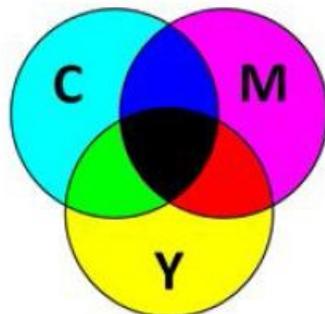
$$I = .596R - .275G - .321B$$

$$Q = .212R - .528G + .311B$$

Модель HSV



Модель CMYK

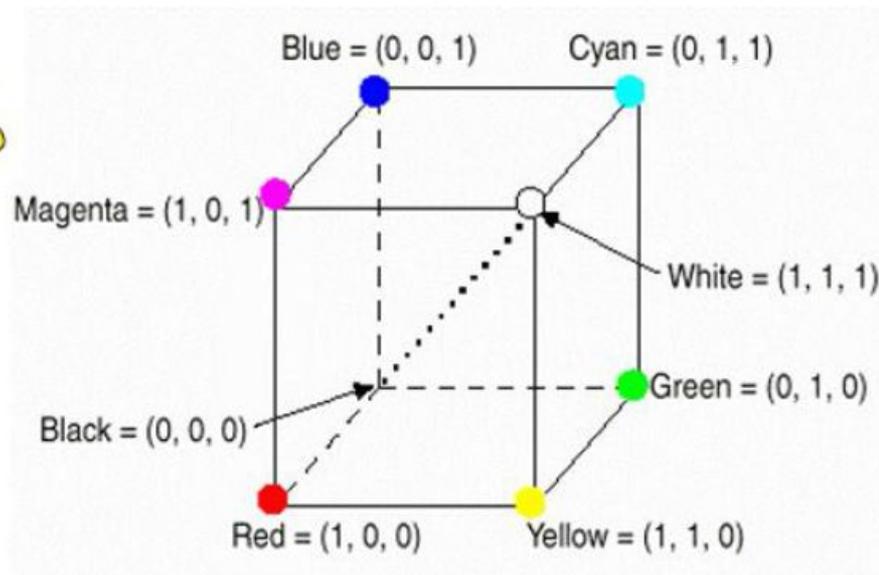


CMY

$$C = G + B = W - R$$

$$M = R + B = W - G$$

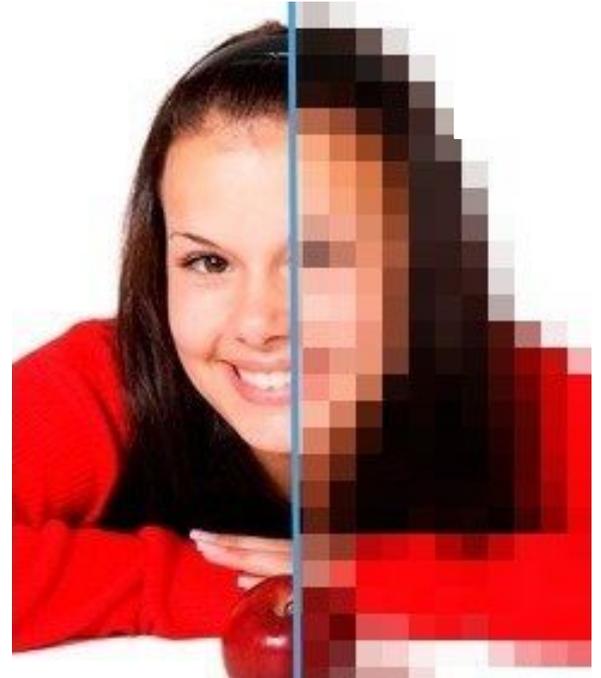
$$Y = R + G = W - B$$



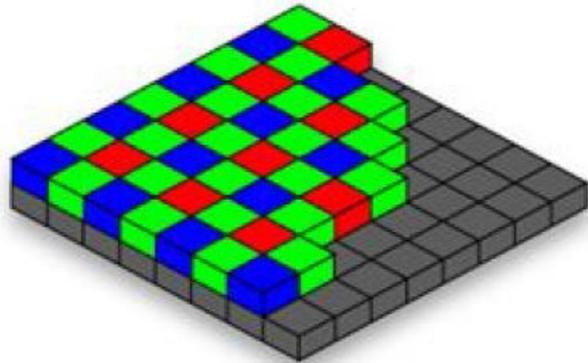
- Аддитивная система – RGB
- Субтрактивная система – CMY
- CMYK для повышения качества

Цифровое цветное изображение

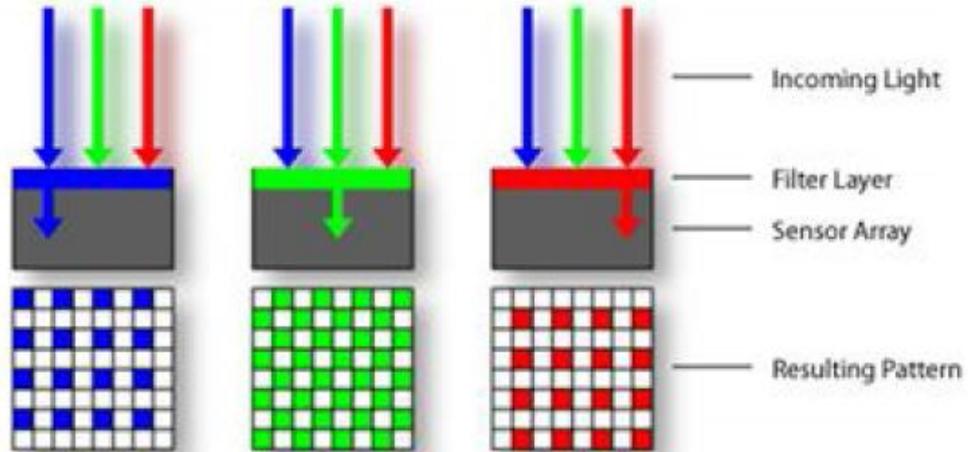
- Цифровое полутоновое изображение – матрица размера $N \times M \times C$, элементами которой являются значения одного из C каналов в каком-либо цветовом пространстве
- Размер одного значения обычно 1 байт, т.е. $\in [0, 255]$
- Сейчас иногда используется большая точность (10-16 бит)



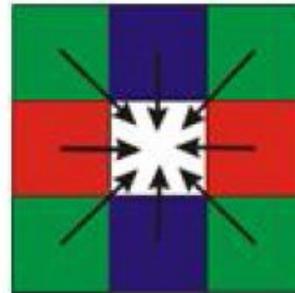
Мозаика и демозаикинг



Байеровский шаблон



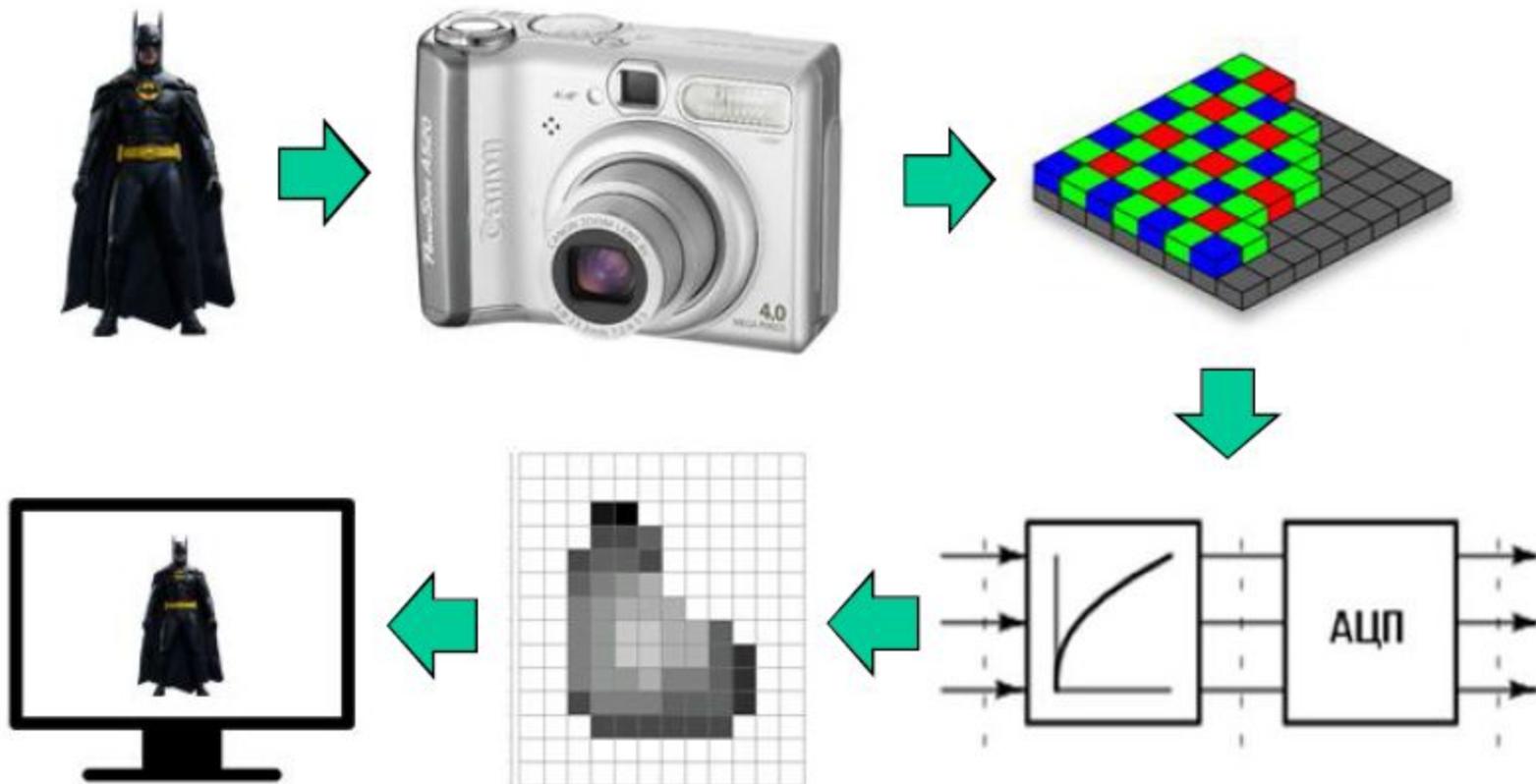
Демозаикинг - оценка пропущенных
цветовых значений



Обработка изображений

- Для улучшения восприятия человеком
- Для улучшения восприятия компьютером
- Для технических нужд
- Для развлечения

Источники ошибок в передаче



Традиционные дефекты



Темное или слабоконтрастное



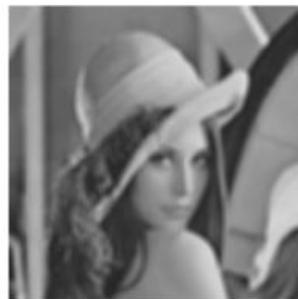
Неправильные цвета



Низкое разрешение



Шумное



Нерезкое



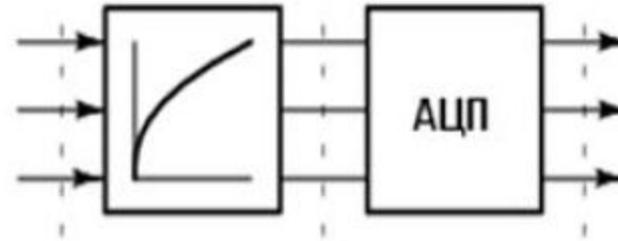
Неравномерно
освещённое



Поврежденное

Плохие яркость / контраст

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



Оценка качества передачи тонов



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

Точечные операторы

Операторы, которые определяют значение выходного пикселя лишь по значению входного. Т.е. все пиксели обрабатываются независимо друг от друга

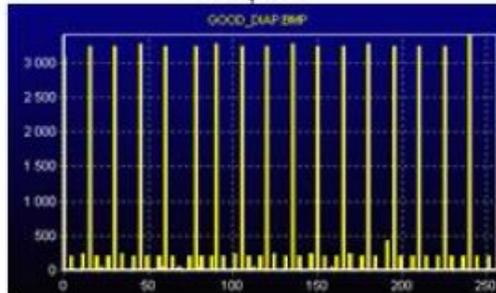
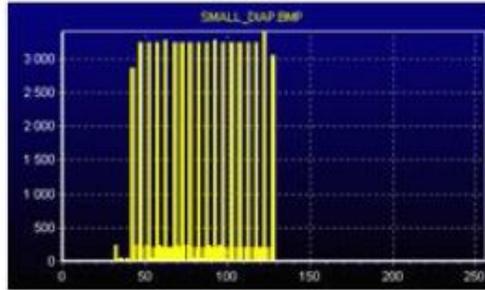
$$f^{-1}(y) = x$$

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

Пишем формулу в “обратном” виде, поскольку хотим восстановить исходное, “правильное” изображение

Линейная коррекция

Компенсация узкого диапазона яркости - линейное растяжение гистограммы
(Histogram equalization)



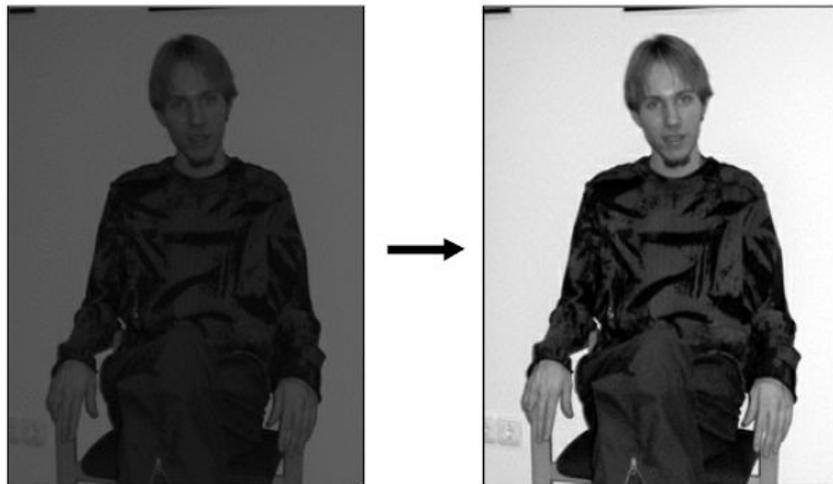
$$f^{-1}(y) = (y - y_{\min}) * \frac{(255 - 0)}{(y_{\max} - y_{\min})}$$

Линейная коррекция

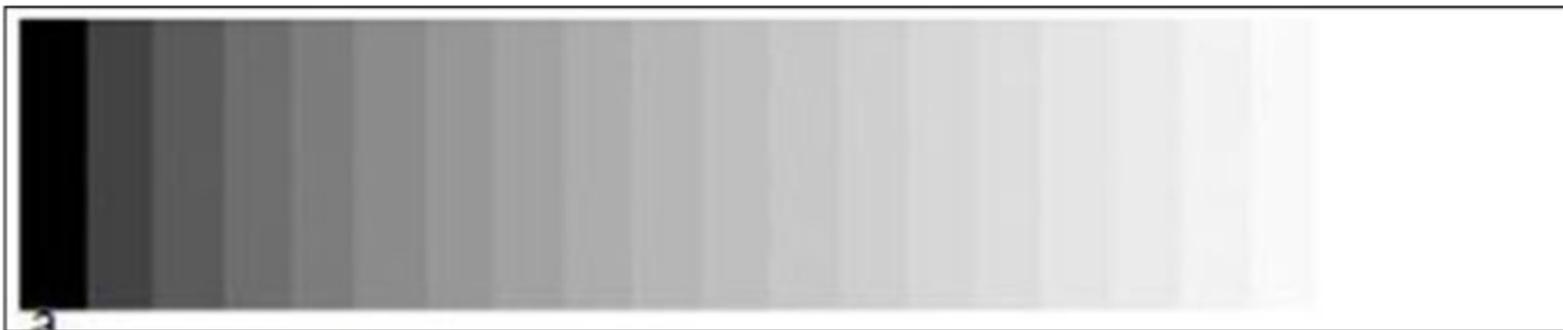
Неустойчива к “выбросам”:



Решение: использовать не максимальные и минимальные значения, а $n\%$ максимальных и $n\%$ минимальных в качестве границ приведения к 0 и 255



Нелинейность восприятия яркости



Физически равномерные яркости



Субъективно равномерные яркости

Нелинейная коррекция

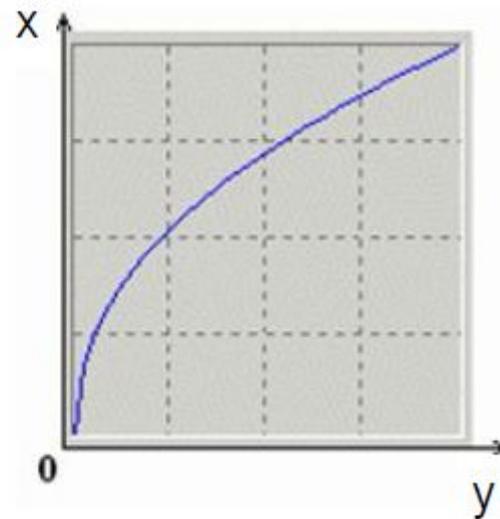
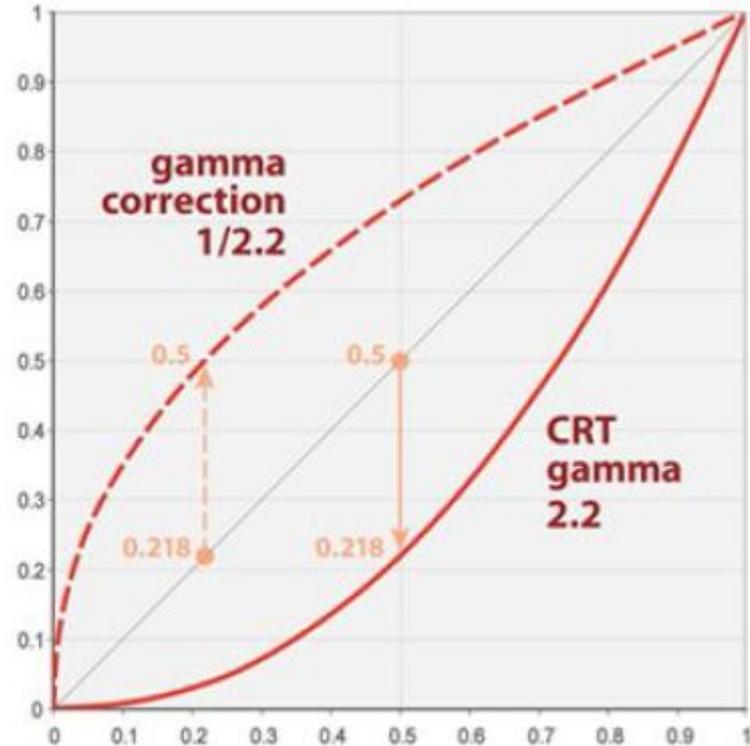


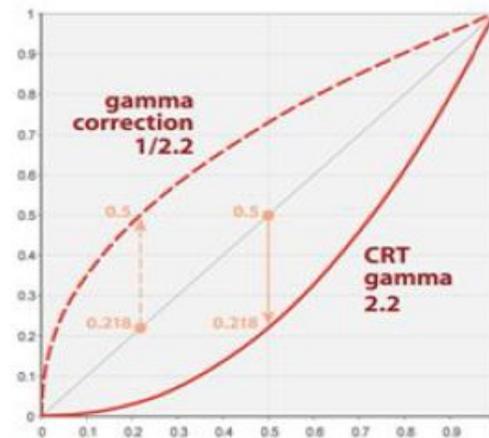
График функции $f^{-1}(y)$

Гамма-коррекция

- Преобразование вида $y = c \cdot x^\gamma$
- Изначальная цель – коррекция для правильного отображения на мониторе
 - на ЭЛТ-мониторах была нелинейная зависимость между напряжением и яркостью пиксела
- Сейчас важнее как раз лучшая контрастность изображения при ограничении диапазона (те же 8бит)



Гамма-коррекция



Нелинейная коррекция

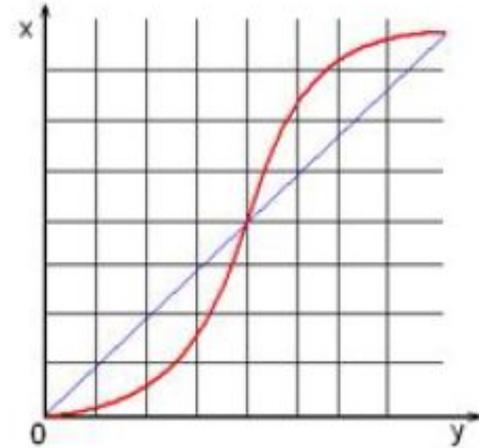
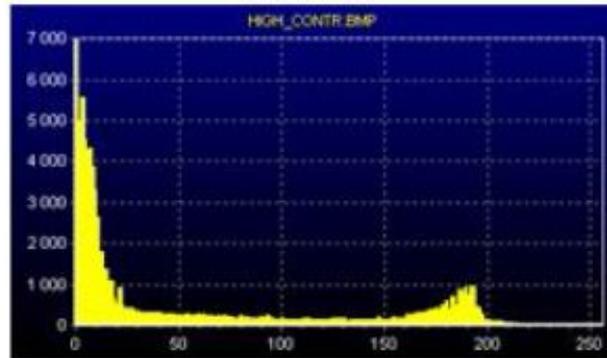
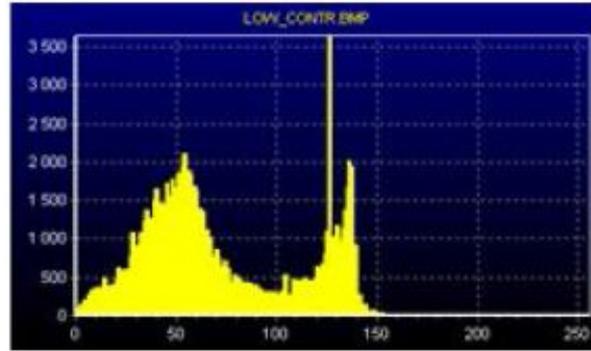


График функции $f^{-1}(y)$

Баланс белого

- Белый цвет при различном освещении может приобретать различные оттенки
- Глаз и мозг умеют подстраиваться под освещение, камера не умеет

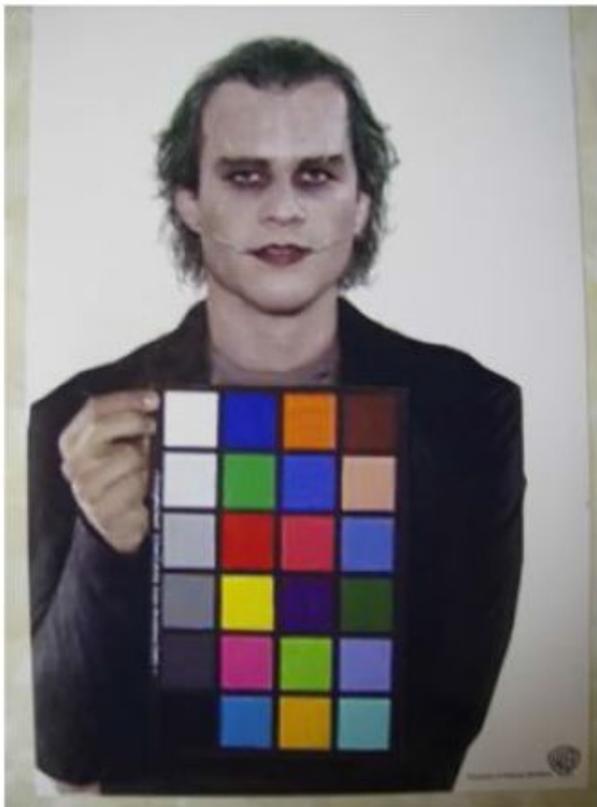
Неправильный баланс



Правильный баланс



Коррекция по шаблону



Используем шаблон с известными цветами

Зная правильные значения цветов,
находим искомое преобразование

Оценка параметров цветокоррекции

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



Другие варианты угадывания

- Модель блика:
 - Обычно блики на изображении имеют цвет источника цвета
 - Ищем самый яркий пиксель и берем веса для каналов, пропорциональные его цветам
- Линейное растяжение гистограммы для каждого канала
 - Наивный и плохой подход, чаще портит цвета, чем восстанавливает

Шумоподавление

Причины возникновения шума:

- Несовершенство сенсоров
- Внешнее электромагнитное воздействие на сенсор
- Пыль/грязь на камере
- Потери при сжатии и передаче



Виды шума



Original



Salt and pepper noise



Impulse noise



Gaussian noise

- **Потеря информации**
(*data drop-out noise*)

- **Соль и перец:**
случайные черные и белые пиксели
- **Импульсный:**
случайные белые пиксели

- **Гауссов:**
колебания яркости,
распределенные по
нормальному закону

- Аддитивный шум
- $\text{Image}(i,j) = \text{true}(i,j) + \text{noise}(i,j)$
- $\text{Noise}(i,j) \sim N(\mu, \sigma)$

Временная фильтрация



Зашумленные изображения



Усреднение по 10
изображениям

Пространственная фильтрация aka свертка

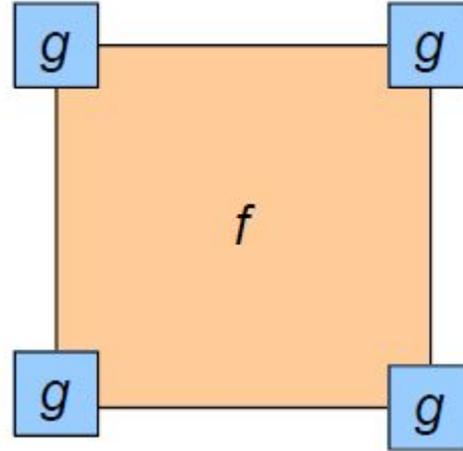
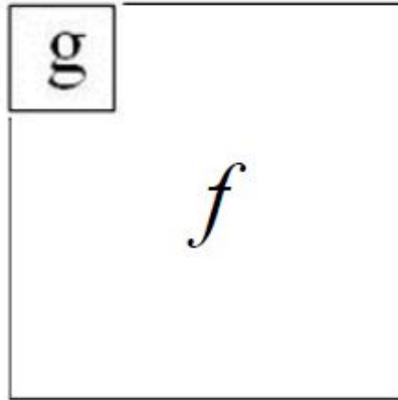
- Вычислим новое значение для пикселя как функцию от его окрестности
- В простейшем случае можем просто усреднить по окрестности
- Обозначим распределение весов по окрестности как ядро фильтра

1 — 9	1	1	1
	1	1	1
	1	1	1

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

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

Свертка и паддинг



Нужно экстраполировать изображение для фильтрации по краям (padding)

- Константным значением
- Повторением значений на краях
- Зеркальным отражением от края изображения
- Замостить изображение

Свойства свертки

- 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 сверток



Original

0	0	0
0	1	0
0	0	0



Filtered
(no change)



Original

0	0	0
0	0	1
0	0	0



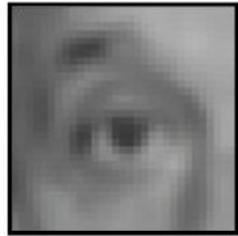
Shifted left
By 1 pixel



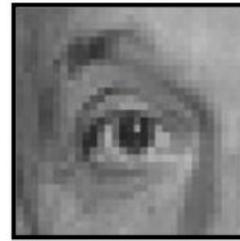
Исходное

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

1	1	1
1	1	1
1	1	1



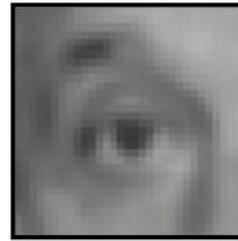
Результат



Исходное

$$\frac{1}{16}$$

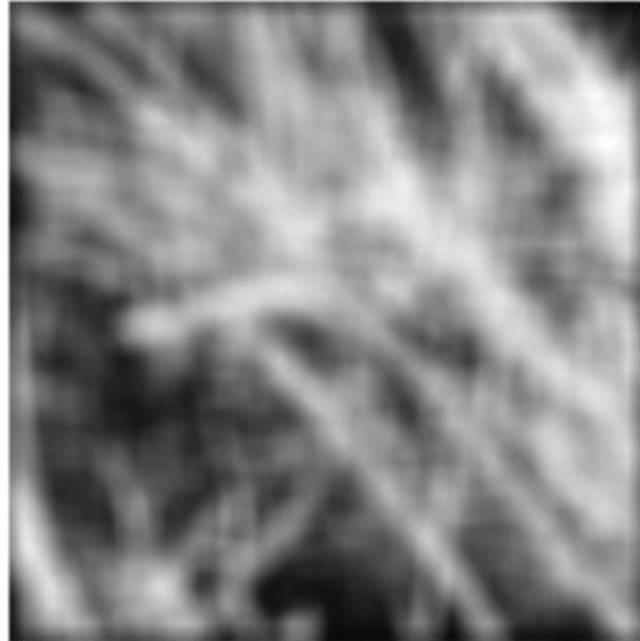
1	2	1
2	4	2
1	2	1



Результат

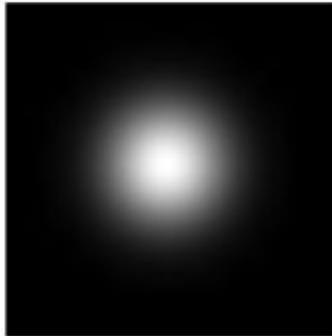
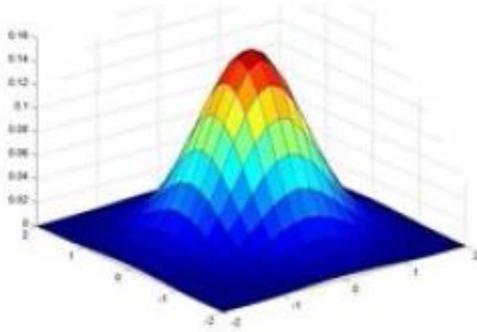
Проблемы вох-фильтра

При свертке Вох-фильтром на изображении появляются угловатые артефакты



Фильтр (ядро) Гаусса

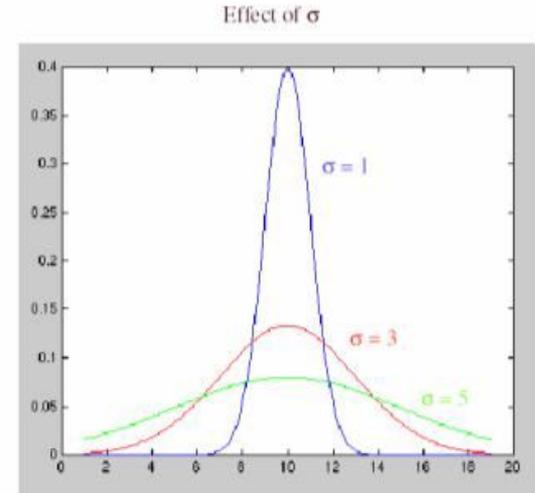
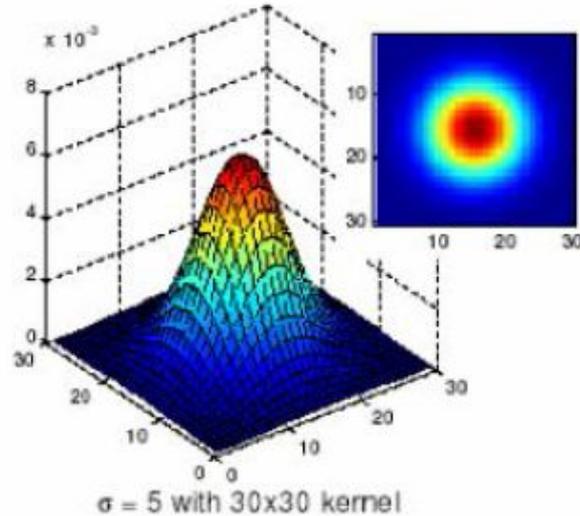
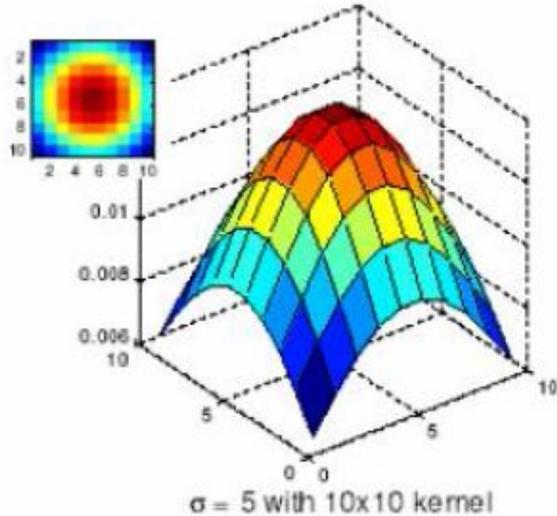
$$G_{\sigma} = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



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

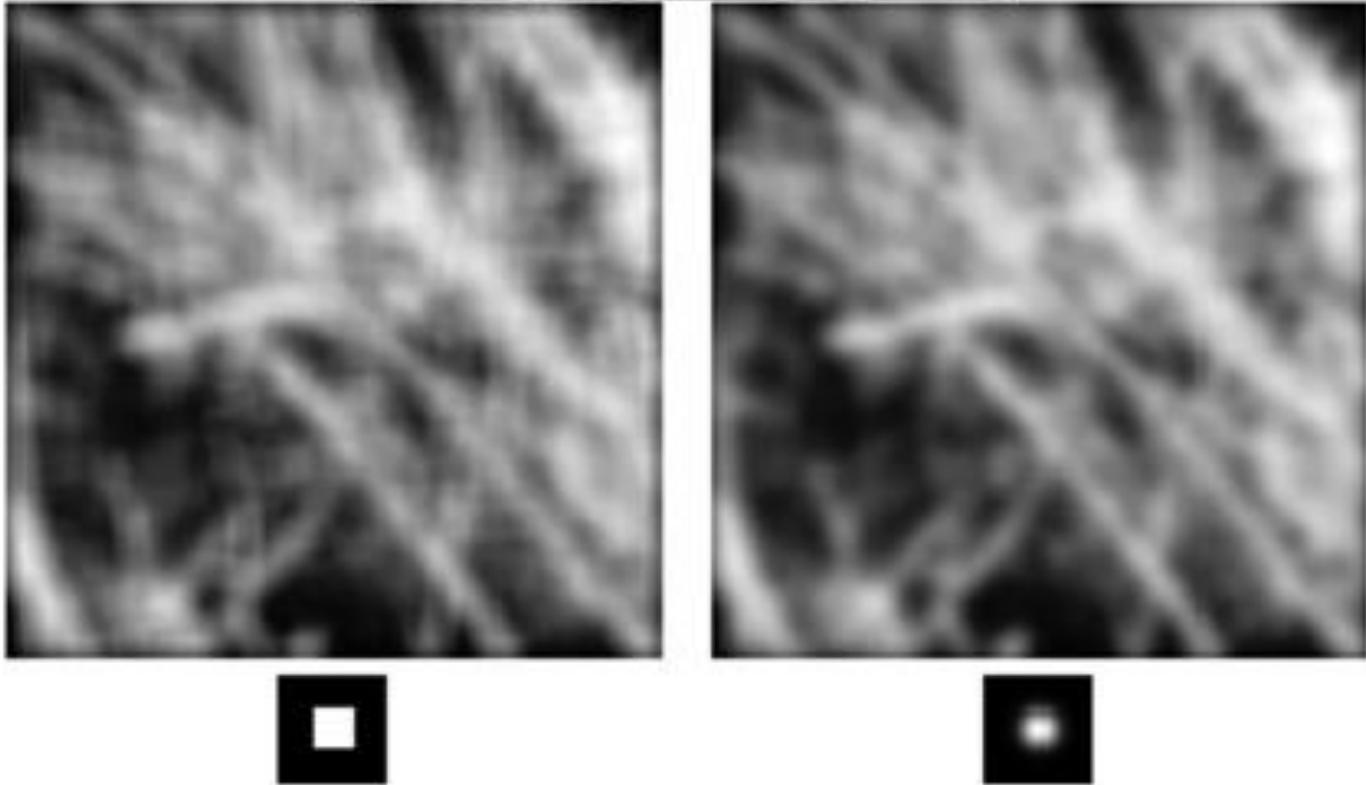
5 x 5, $\sigma = 1$

Выбор размера ядра



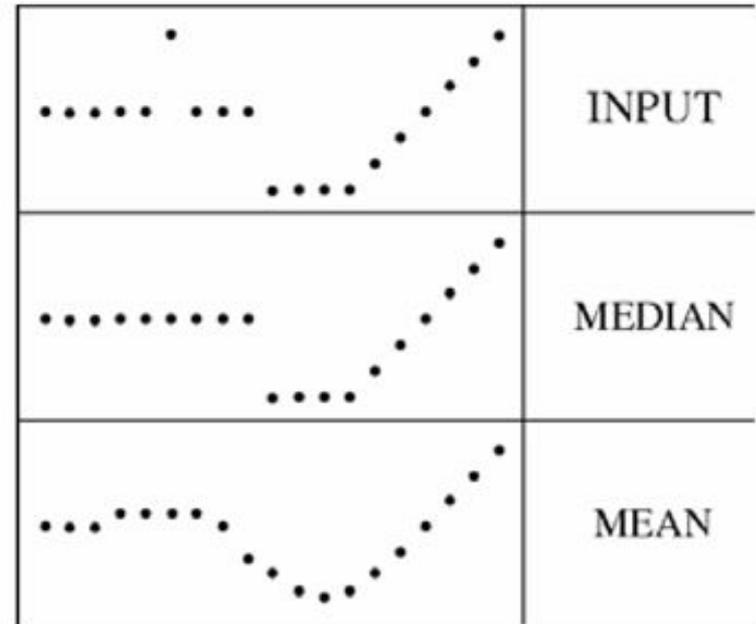
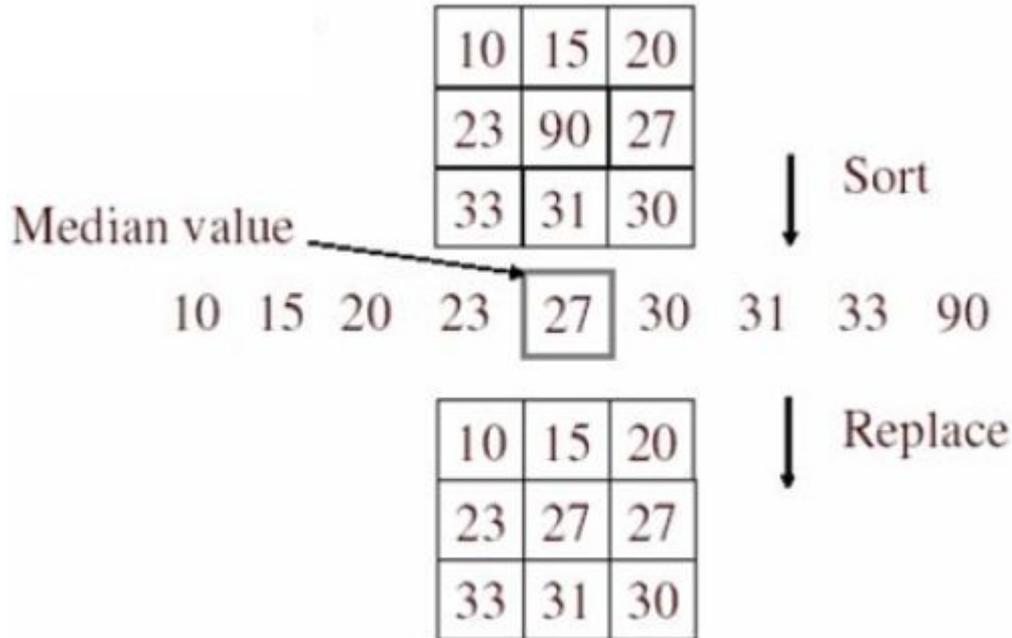
Размер ядра ограничен - обычно берем в качестве радиуса 3σ

Фильтр Гаусса - сравнение



Медианный фильтр

Выбираем медиану из выборки пикселей по окрестности данного -
устойчивость к выбросам!



Сравнение фильтров

3x3

5x5

7x7

Гауссов



Медианный



Повышение резкости

Что теряется при сглаживании?



Добавим дополнительно высокие частоты:



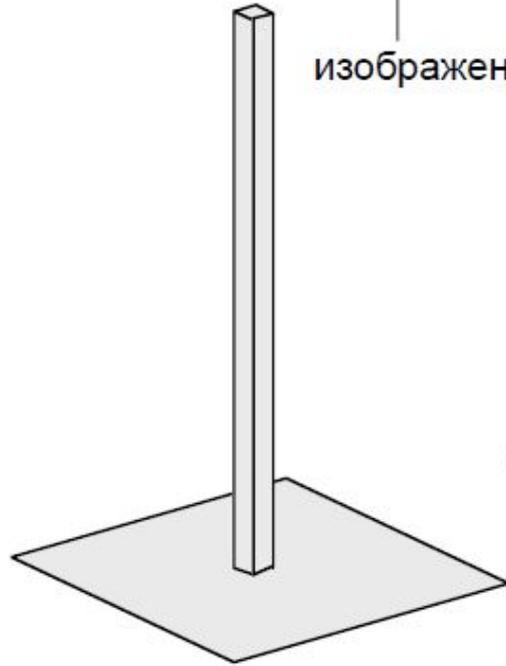
Повышение резкости

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

↑
изображение

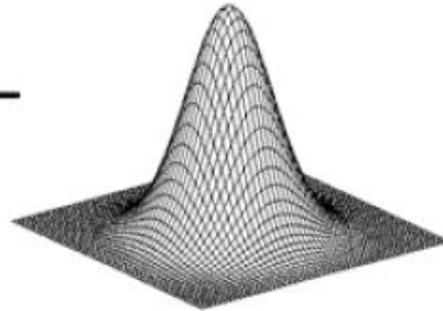
↑
сглаженное
изображение

↑
Единичный
Фильтр



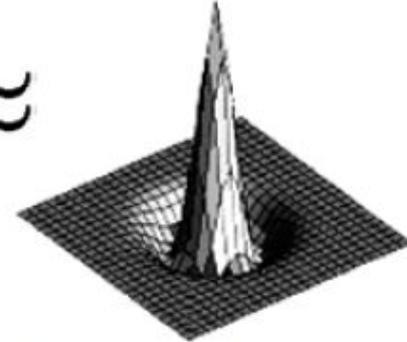
Единичный фильтр

—



Гауссов

≈



Лапласиан Гауссиана

Повышение резкости

Ядро
свертки

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



Выделение краев



Описание края

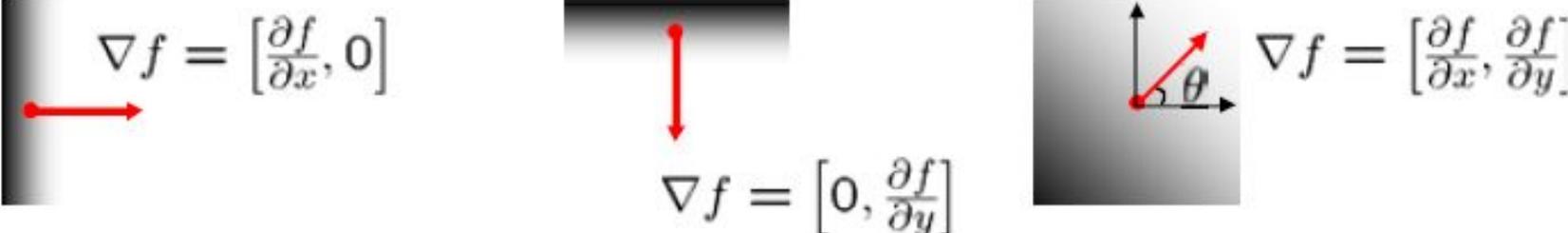
Край - место резкого изменения значения функции интенсивности изображения



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: $\theta = \tan^{-1} \left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$

Дифференцирование и свертка

- 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 линейна и инвариантна к сдвигу
- Мы можем записать ее как свертку

-1	1
----	---

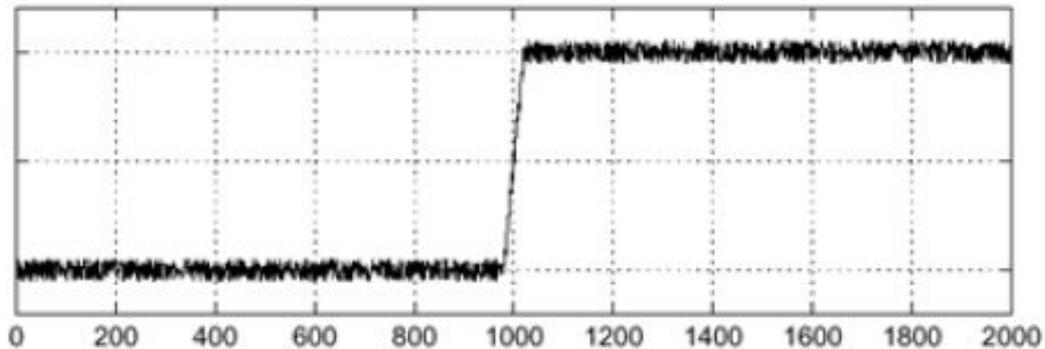
Простейший фильтр

Влияние шума

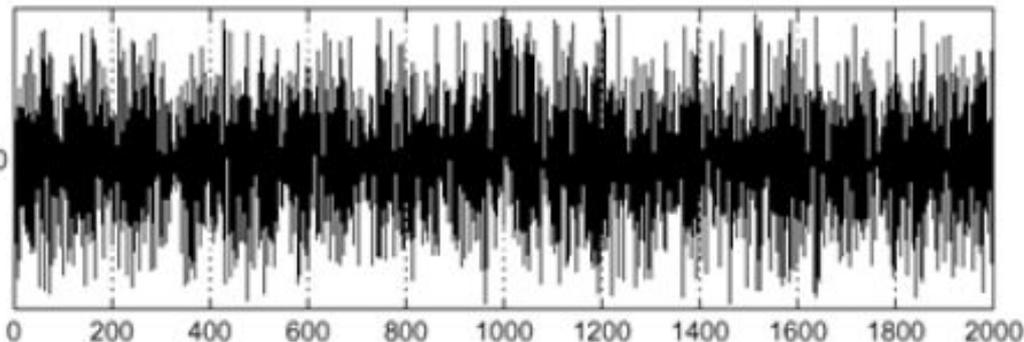
Разностные
производные
чувствительны к
шуму

Выход - перед
поиском краев
сглаживаем
изображение!

$f(x)$

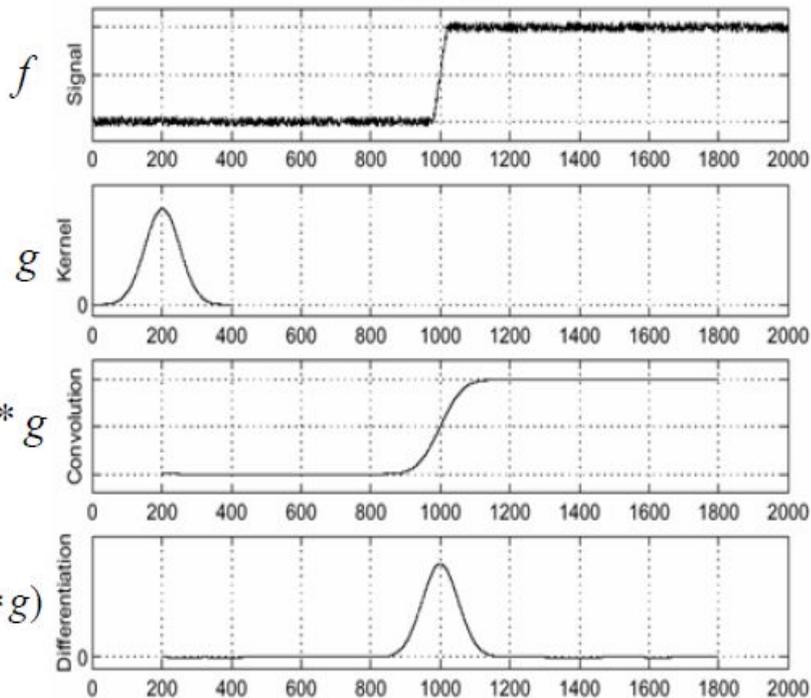


$\frac{d}{dx}f(x)$



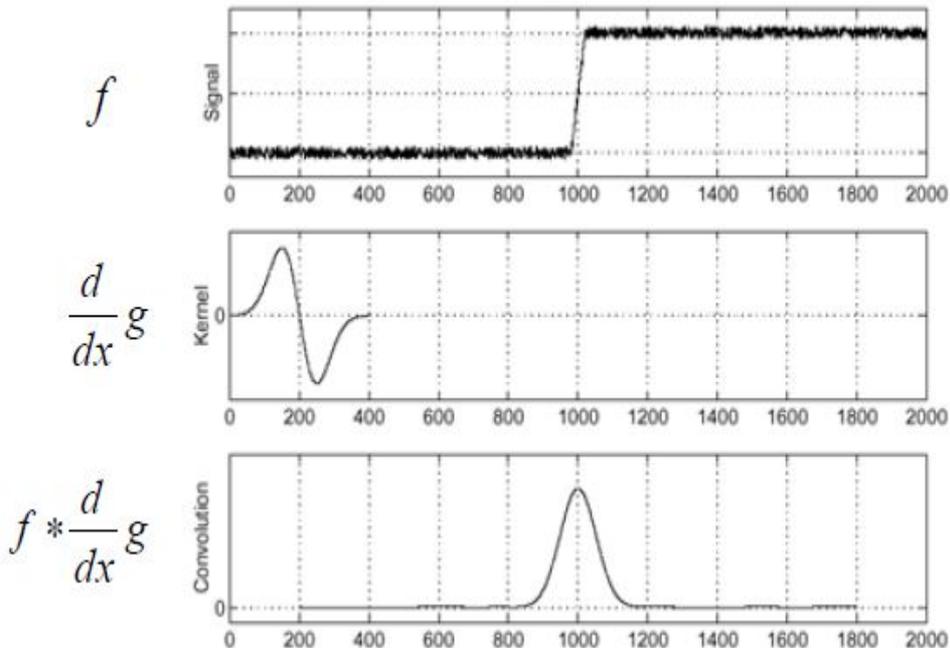
Нивелирование влияния шума

Sigma = 50



$$\frac{d}{dx}(f * g) = f * \frac{d}{dx}g$$

Sigma = 50



Популярные разностные фильтры

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

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

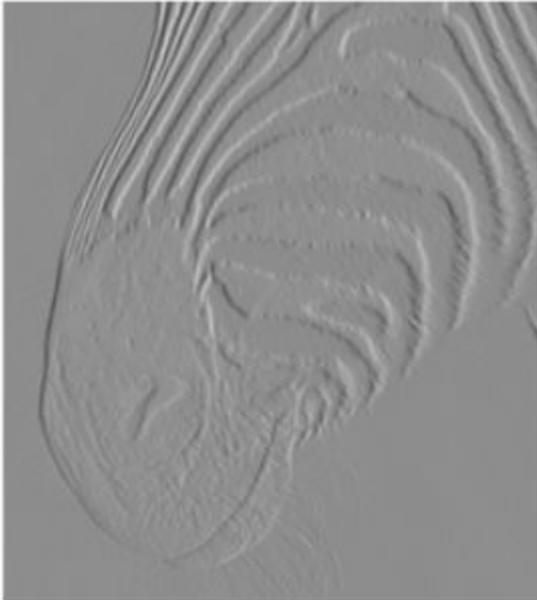
$$\begin{bmatrix} +3 & +10 & +3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix} \quad \begin{bmatrix} +3 & 0 & -3 \\ +10 & 0 & -10 \\ +3 & 0 & -3 \end{bmatrix}$$

Шарп

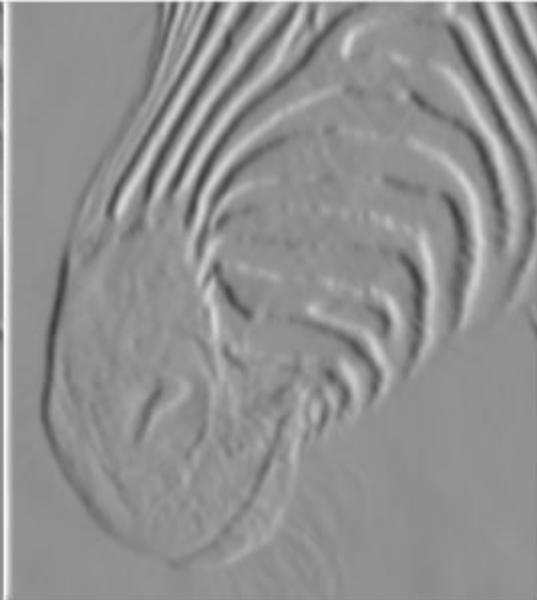
$$S_x = \begin{bmatrix} 1 & 2 & 0 & -2 & -1 \\ 4 & 8 & 0 & -8 & -4 \\ 6 & 12 & 0 & -12 & -6 \\ 4 & 8 & 0 & -8 & -4 \\ 1 & 2 & 0 & -2 & -1 \end{bmatrix} \quad S_y = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Собель 5x5

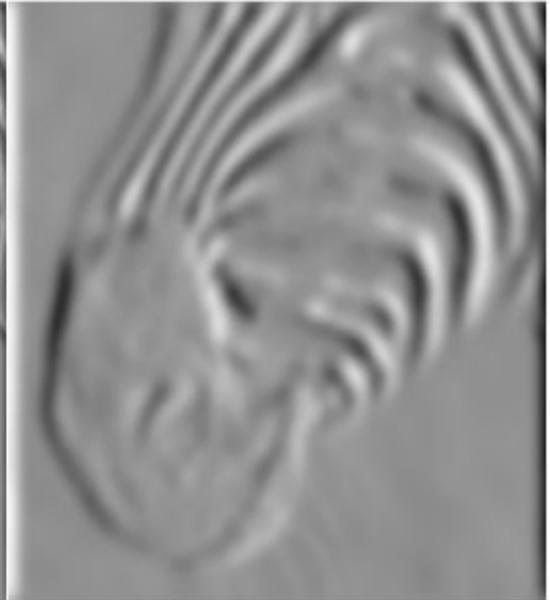
Разностные фильтры - примеры



1 pixel



3 pixels



7 pixels

Просто градиентов недостаточно!

- Края слишком толстые и размытые - недостаточно точности
- Края то появляются то пропадают - нет информации о связности



Алгоритм Кэнни

1. Свертка изображения ядром - производной фильтра Гаусса
2. Поиск направления и модуля (сила) градиента
3. Подавление локальных немаксимумов вдоль направлений градиента (позволяет истончить найденные края до 1 пикселя)
4. Отсечение по двойному порогу (гистерезису)
 - a. Края с силой выше верхнего порога считаем сильными и оставляем
 - b. Края с силой ниже нижнего порога отбрасываем
 - c. Края между порогами остаются потенциальными
5. Потенциальные края считаем итоговыми краями, если они прилегают к сильным краям

Алгоритм Кэнни



Резюме

- Восприятие человеком цвета линейно
 - Восприятие яркости нелинейно, но мы держимся
 - Целый ряд задач можно решать линейными методами, в особенности линейной фильтрацией, т.е. сверткой
-
- Все равно в итоге все используют нейросети...
 - ... но даже в нейросетях свертки играют важнейшую роль!