

Анализ изображений

В презентации используются материалы лекций Антона Конушина по предмету «Компьютерное зрение» на Факультете вычислительной математики и кибернетики МГУ

источник www.lektorium.tv



La Gare Montparnasse, 1895

Мы видим

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

Компьютер видит

Задача компьютерного зрения

- «To see means to know what is where by looking»
 - *David Marr, Vision, 1982*
- «Тест Тьюринга» - компьютер должен ответить на любой вопрос об изображении, на который может ответить человек
- Что это в действительности обозначает?
 - Зрение - источник семантической информации о мире
 - Зрение - источник метрической информации о трехмерном мире

Семантическая информация



Slide credit: Fei-Fei, Fergus & Torralba

Классификация сцены

- вне помещения
- город
- уличное движение
- Пекин, Китай
- Пл. Тяньаньмэнь



Выделение объектов



Выделение границ объектов



Признаки объектов



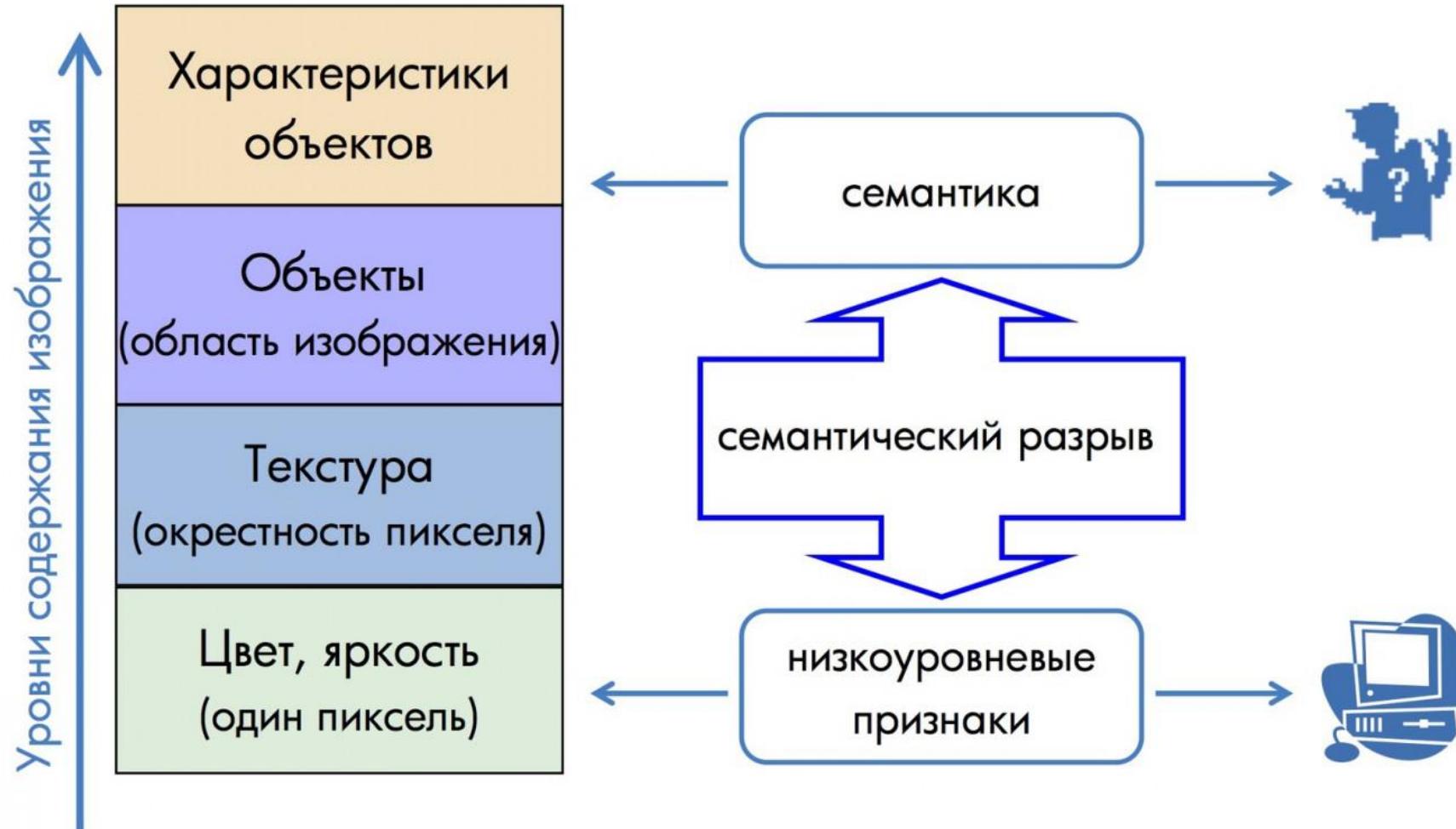
Почему зрение это сложно

- ✓ 25+% мозга человека отвечает за зрение
- ✓ Неоднозначность интерпретации
- ✓ «ИИ-полная» задача – решение задачи зрения на уровне человека равносильно решению задачи искусственного интеллекта



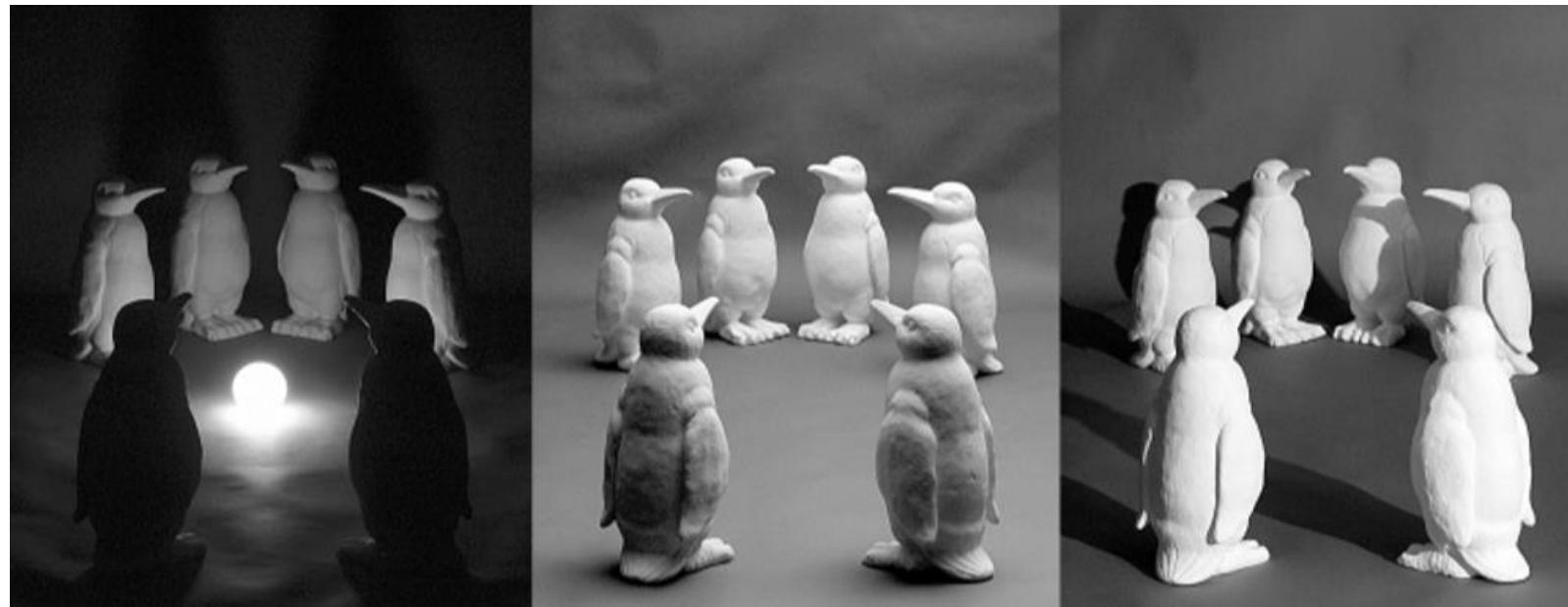
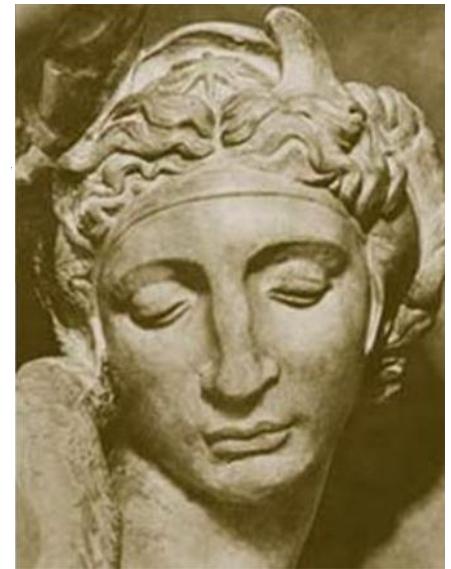
Наш мозг склонен видеть то, что мы хотим видеть, а как научить такому же компьютер — открытый вопрос.

Семантический разрыв



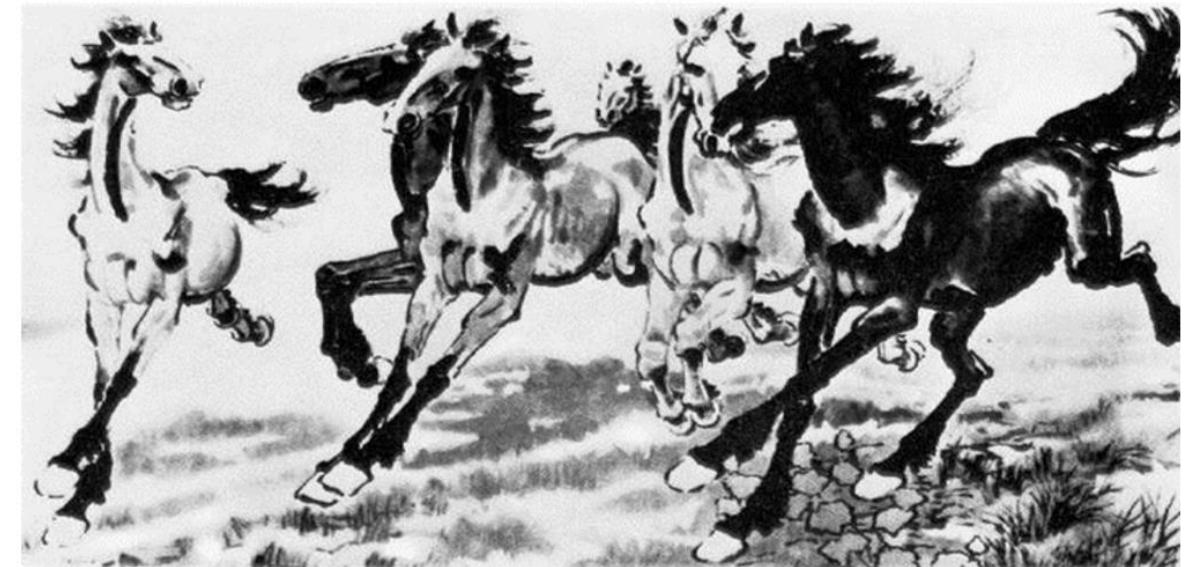
Сложности компьютерного зрения:

- один и тот же предмет кажется разным с разных точек наблюдения



- Цвет видимых объектов зависит от освещения

- Восприятие размера зависит от контекста



Xu, Beihong 1943

- Живые объекты могут менять форму

- При перекрытии теряется часть информации об объекте
- Маскировка природных объектов



Внутриклассовая изменчивость





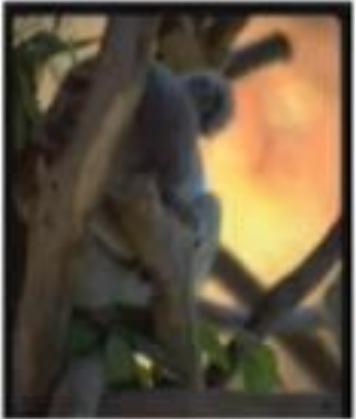
Illumination



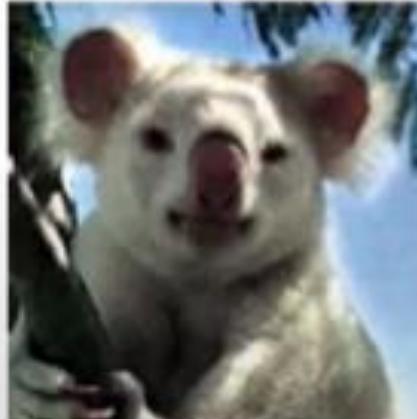
Object pose



Clutter



Occlusions



Intra-class
appearance



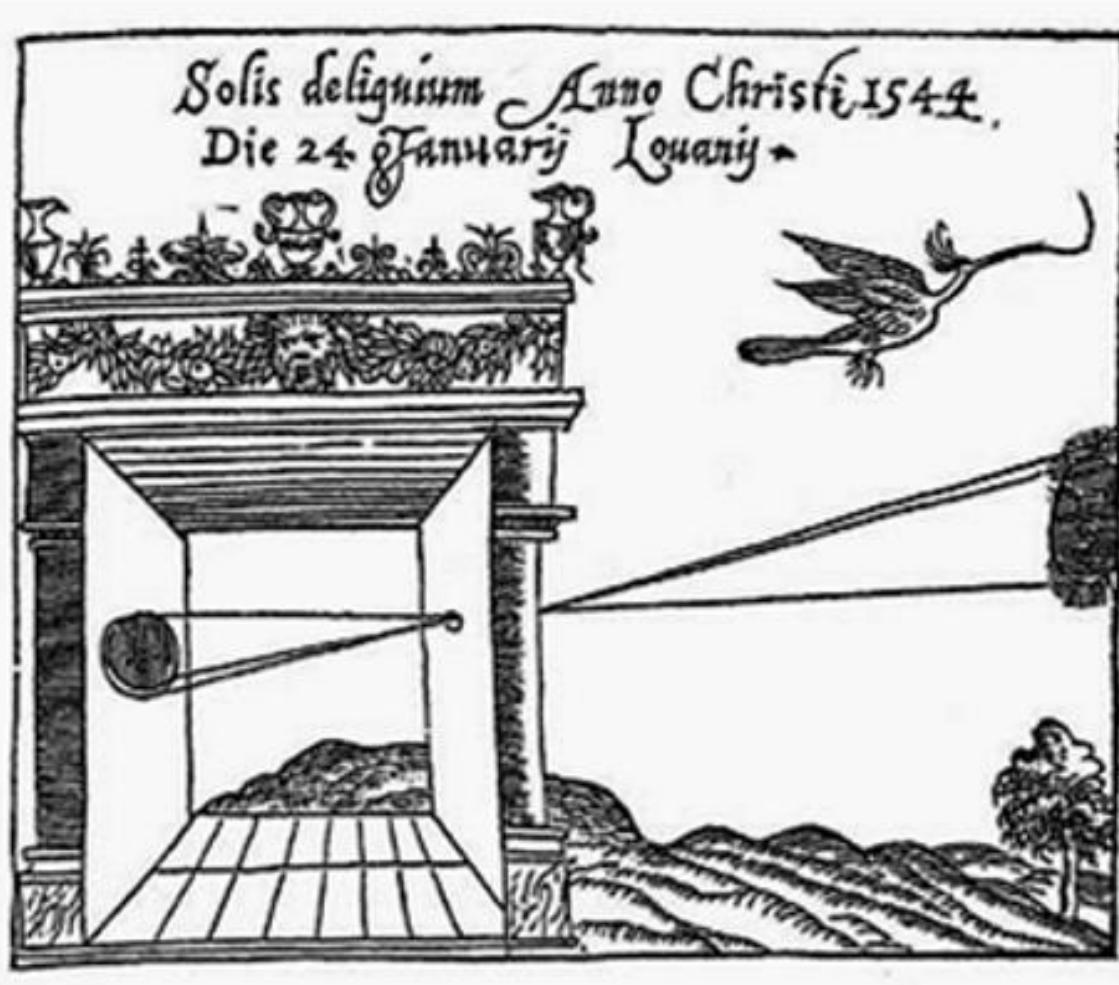
Viewpoint

Выводы: зрение изначально нечеткая задача

- Разные 3D сцены дают одно и то же 2D изображение
- Необходимы априорные знания о структуре и свойствах мира
- Нам нужно сопоставлять наблюдения и априорные знания



История: Камера-обскура



- Принцип был известен еще Аристотелю (384-322 до Н.Э.)
- Помогала художникам: описана Леонардо да Винчи (1452-1519)

История: 1525



Figure 3. Albrecht Dürer

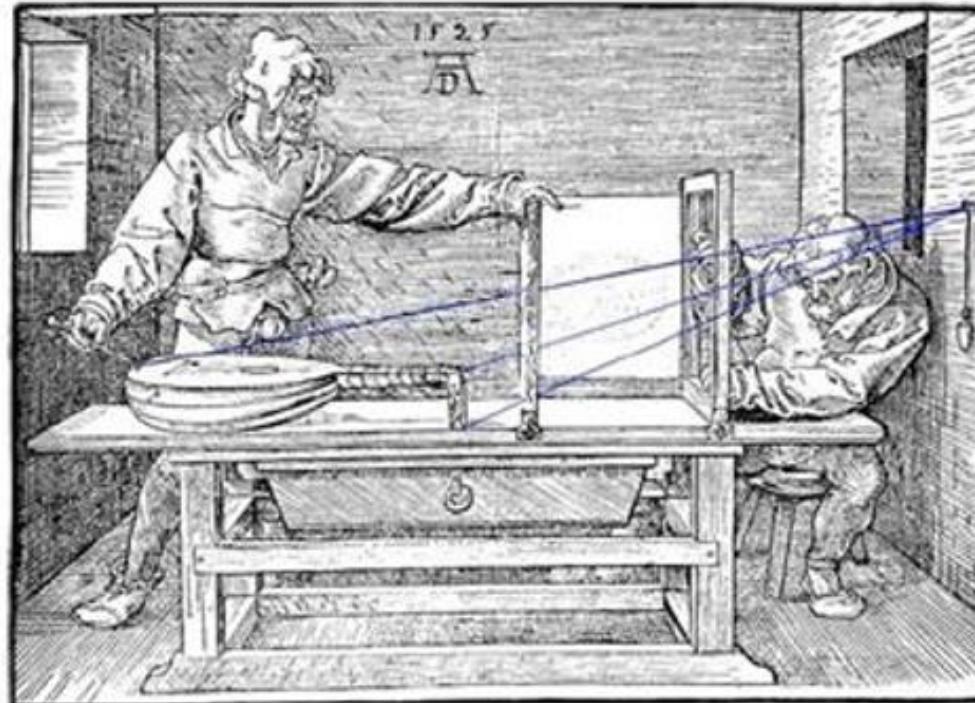
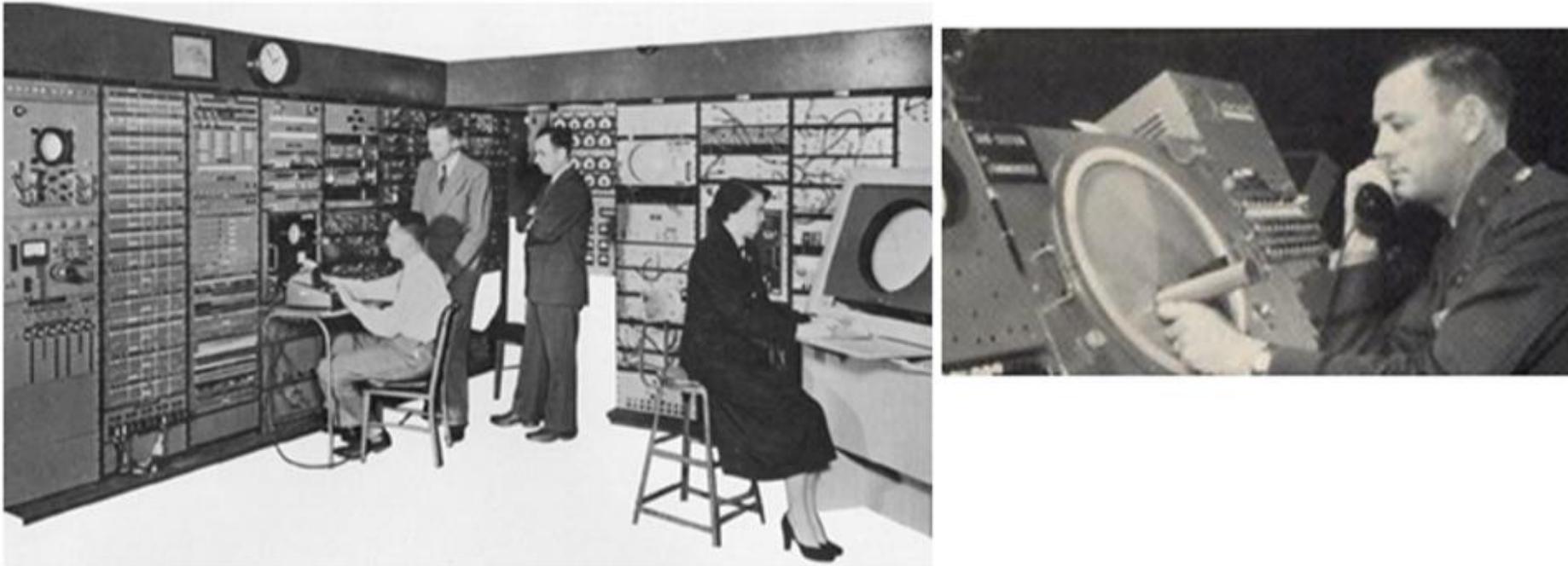


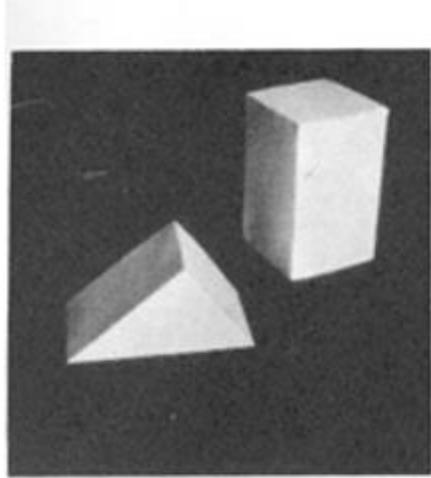
Figure 4. Woodcut by Dürer *The Draughtsman And The Lute* from his book: '*Treatise on Mensuration with the Compass and Ruler in Lines, Planes, and Whole Bodies*' (1525)

История: Whirlwind, MIT, 1951

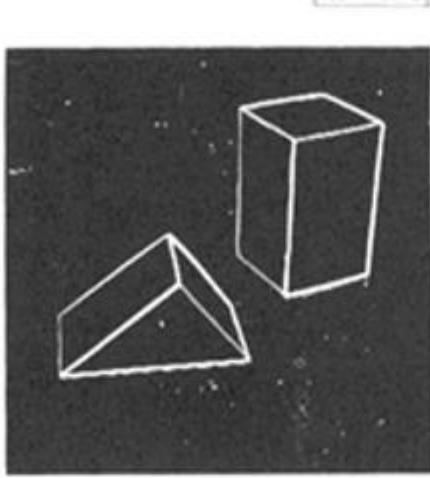


- Первый компьютер, отображающий текст и графику в реальном времени на мониторе
- Точками карту, значком самолёт.
- «Световое перо» для взаимодействия с экраном (запрос информации об объекте)

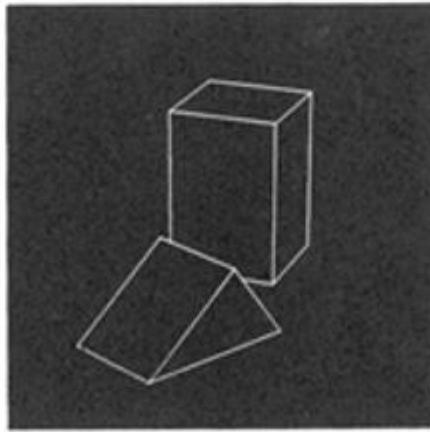
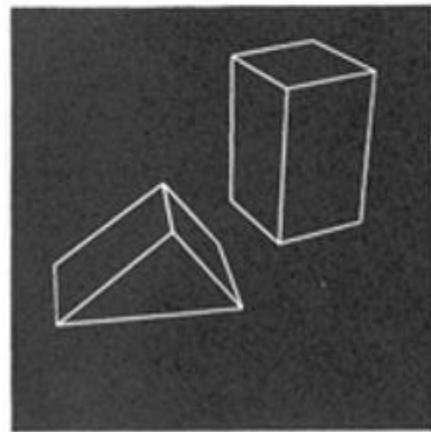
История: Положено начало компьютерному зрению



(a) Original picture.



(b) Differentiated picture.



L. G. Roberts, *Machine Perception of Three Dimensional Solids*,
Ph.D. thesis, MIT Department of
Electrical Engineering, 1960

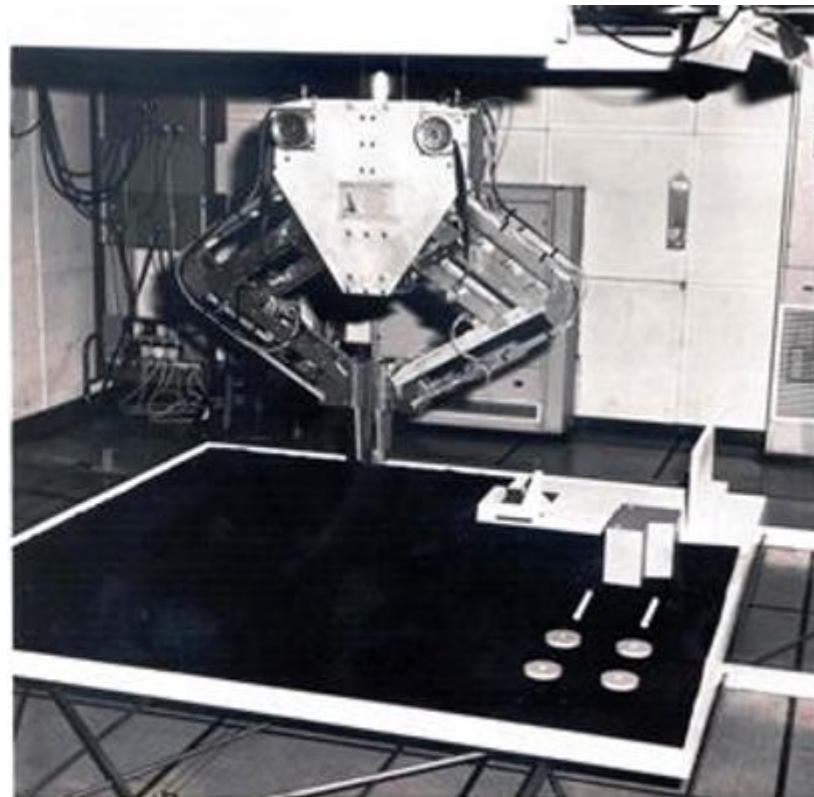
История: Virtual Reality, Harvard, 1968



- Ivan Sutherland перешел в Гарвард, где разработал первый Head Mounted Display (HMD)
- Виртуальная комната (wireframe), в которую можно войти

История: Freddy II, 1973

- Университет Эдинбурга
- Один из первых роботов с системой машинного зрения
- 5 степеней свободы
- Умеет собирать машинки из кубиков, разбросанных по столу
- 384КБ RAM в управляемом компьютере



Области применения компьютерного зрения

- **Распознавание и обработка документов**
- **Обработка фотографий** – улучшение качества, ретушь, изменение размера и формы, композиция
- **Интернет** – поиск, аннотация, поиск дубликатов, распознавание объектов
- **Системы безопасности** – видеонаблюдение, отслеживание, распознавание объектов, распознавание жестов и событий
- **Дистанционное зондирование и ГИС** – карты, анализ спутниковых данных
- **Неразрушающий контроль** - диагностика, контроль качества
- **Медицинские системы** – анализ томограмм
- **Спецэффекты в кино** – композиция, монтаж фонов, захват движения

Детектор лиц Viola-Jones (2001)



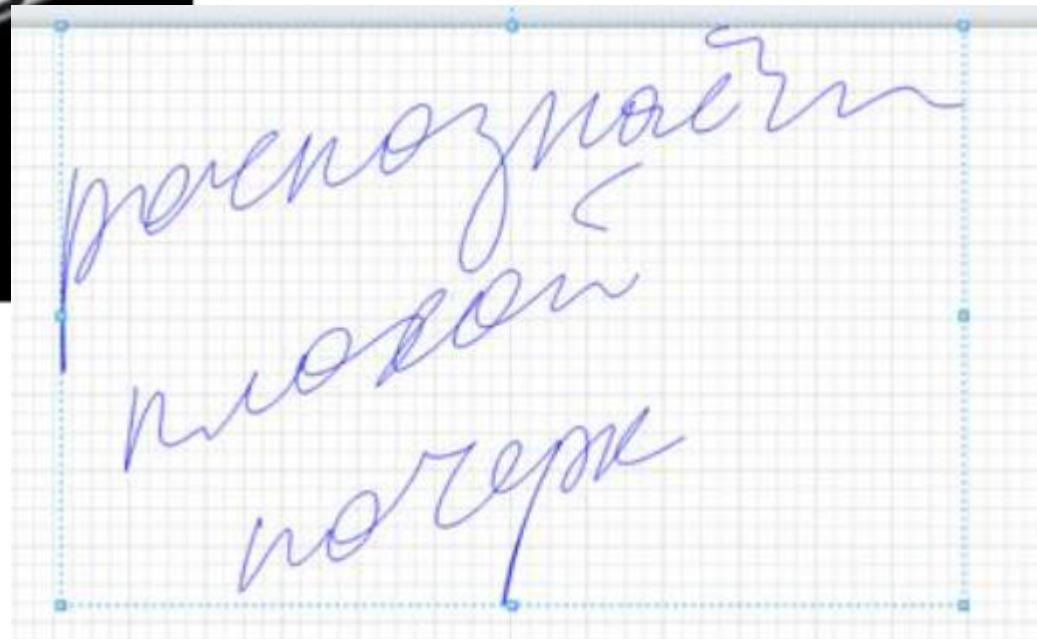
Алгоритм Viola-Jones – первый быстрый и надежный алгоритм поиска лиц. Демонстрация силы машинного обучения.

Kinect (2010)

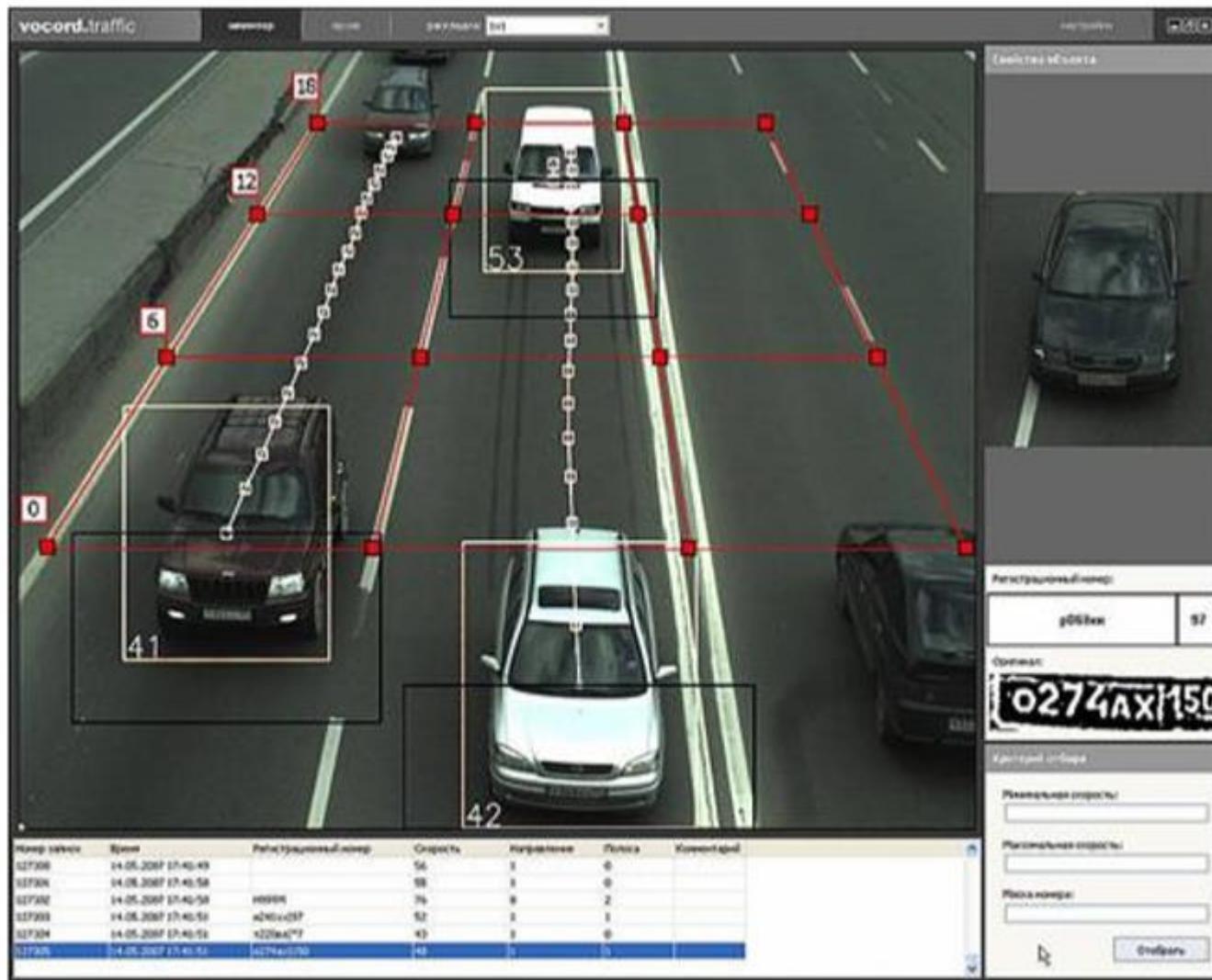


Первая потребительская система взаимодействия с
компьютером с помощью жестов

Распознавание текста



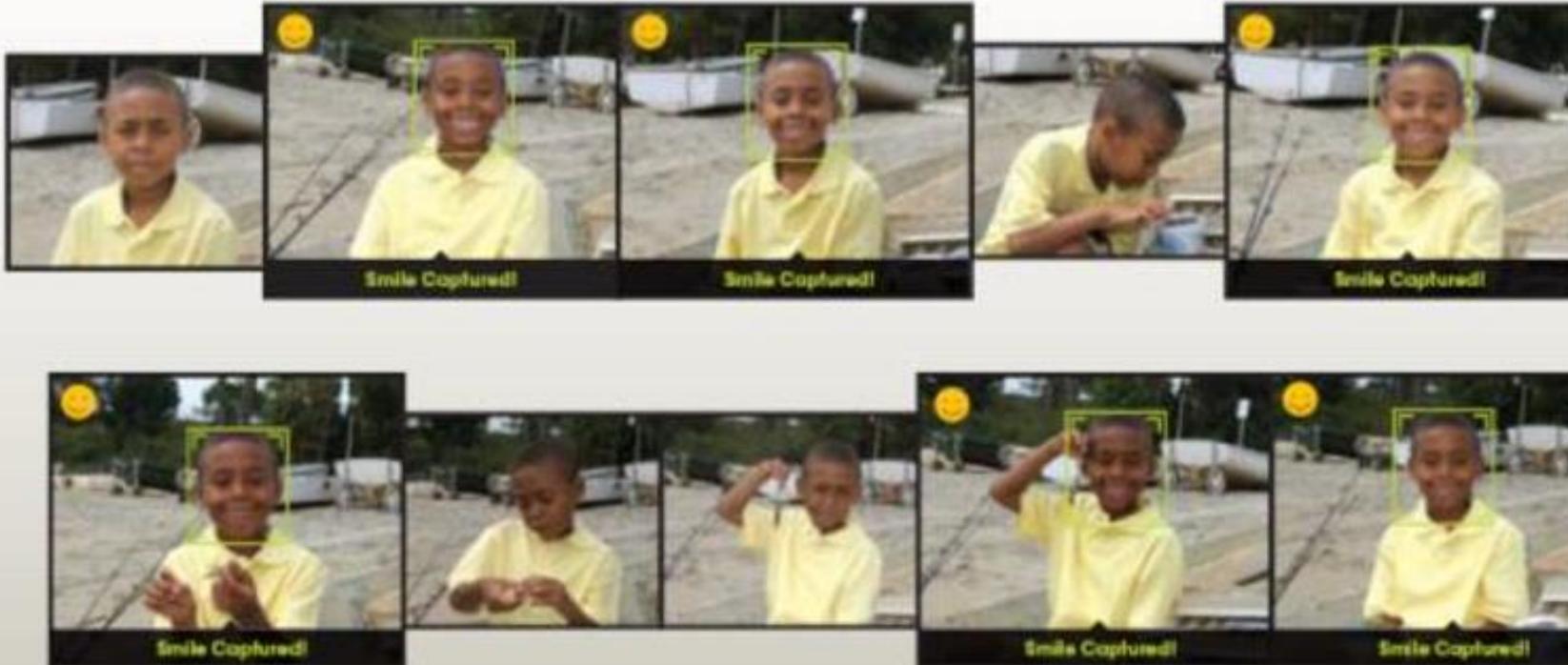
Видеонаблюдение



Обнаружение лица и улыбки

The Smile Shutter flow

Imagine a camera smart enough to catch every smile! In Smile Shutter Mode, your Cyber-shot® camera can automatically trip the shutter at just the right instant to catch the perfect expression.



Биометрия

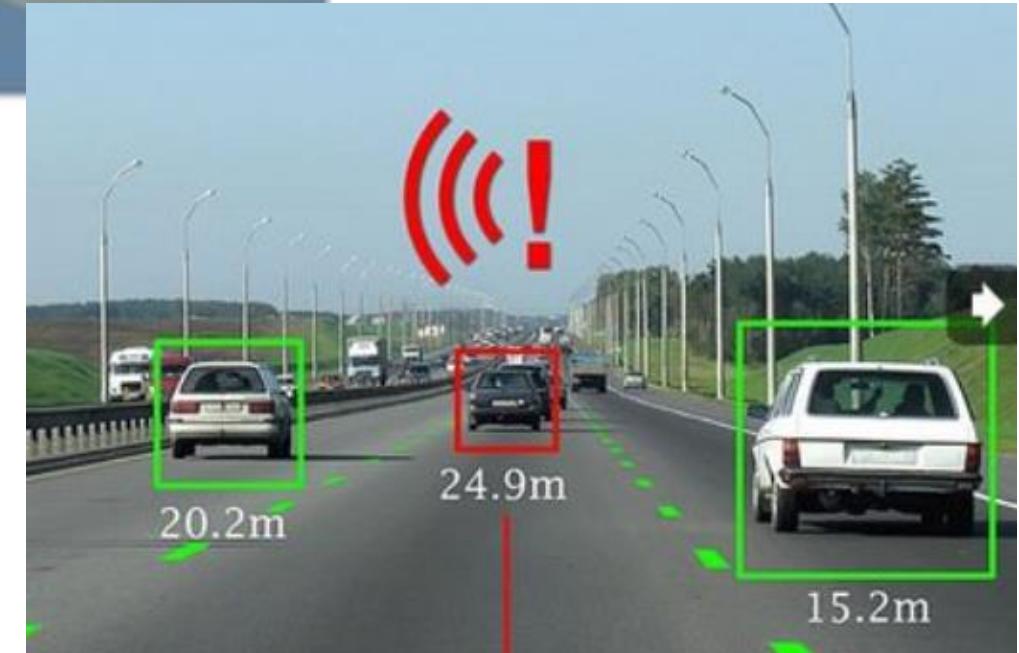
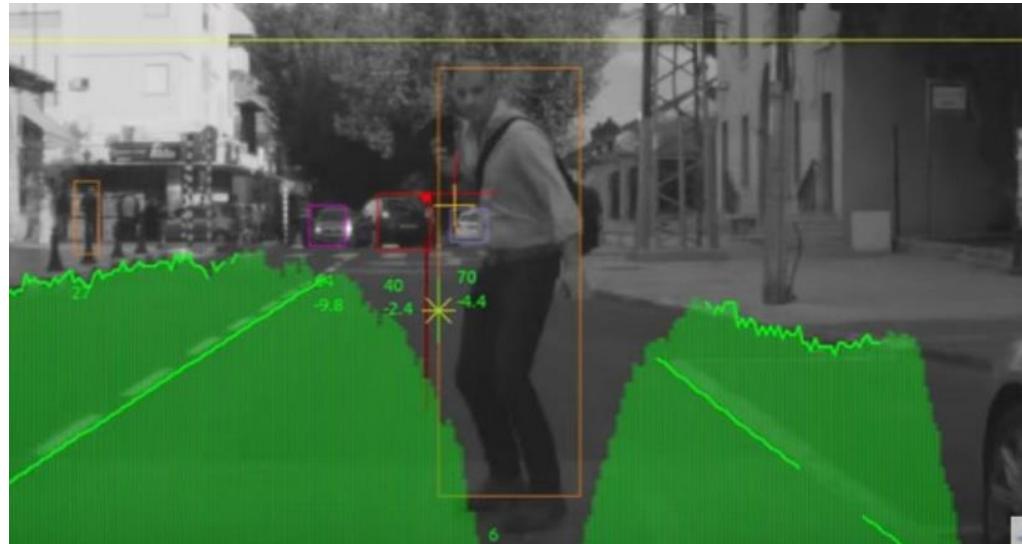


Biolink, Россия
<http://www.biolink.ru>

Сканеры отпечатков пальцев для контроля доступа

Умные машины

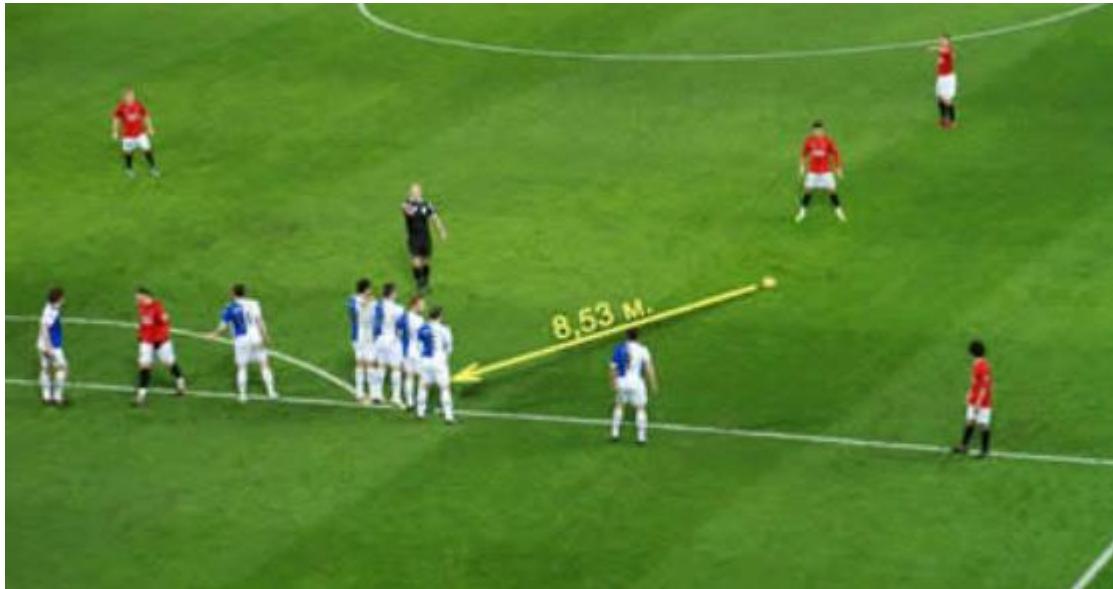
К 2010: 70% производителей машин



3D модели и захват движения



Спортивные соревнования



Зрение в космосе

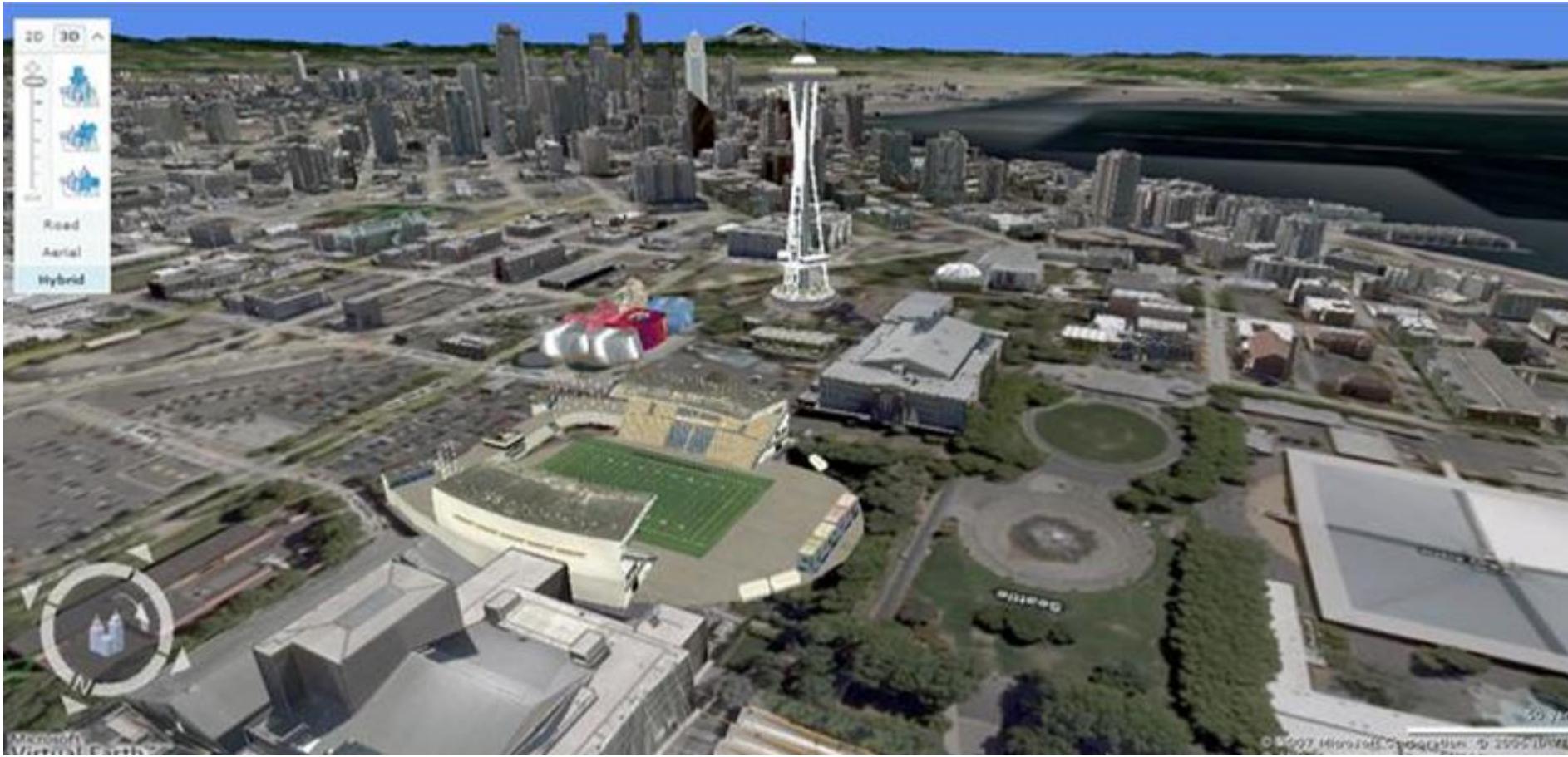


[NASA'S Mars Exploration Rover Spirit.](#)

Системы зрения использовались для:

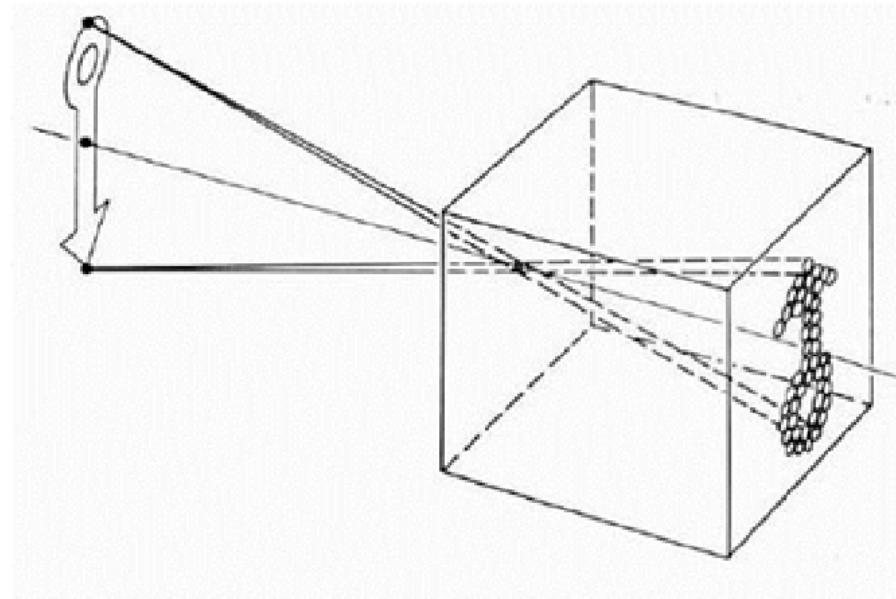
- Склейка панорам
- 3D моделирование местности
- Поиск препятствий, определение местоположения

Трехмерные карты



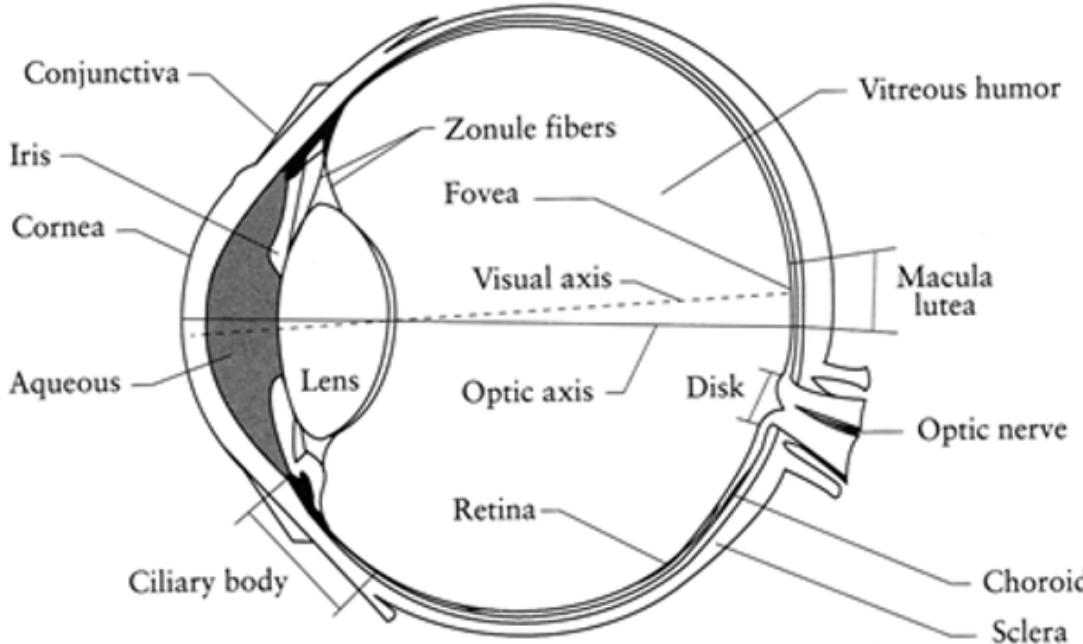
Как же устроено наше зрение

Вспомним модель камеры-обскура



Камера-обскура:

- Захватывает пучок лучей, проходящих через одну точку
- Точечное отверстие называется «Центр проекции»
(фокальная точка / **focal point**)
- Изображение формируется на картинной плоскости (**Image plane**)



Глаз как камера!

- **Радужка** – цветная пленка с радиальными мышцами
- **Зрачок** - отверстие (апертура), диаметр управляетяется радужкой
- **Хрусталик** – «линза», меняющая форму под действием мышц
- Где матрица?
 - Клетки-фоторецепторы на сетчатке

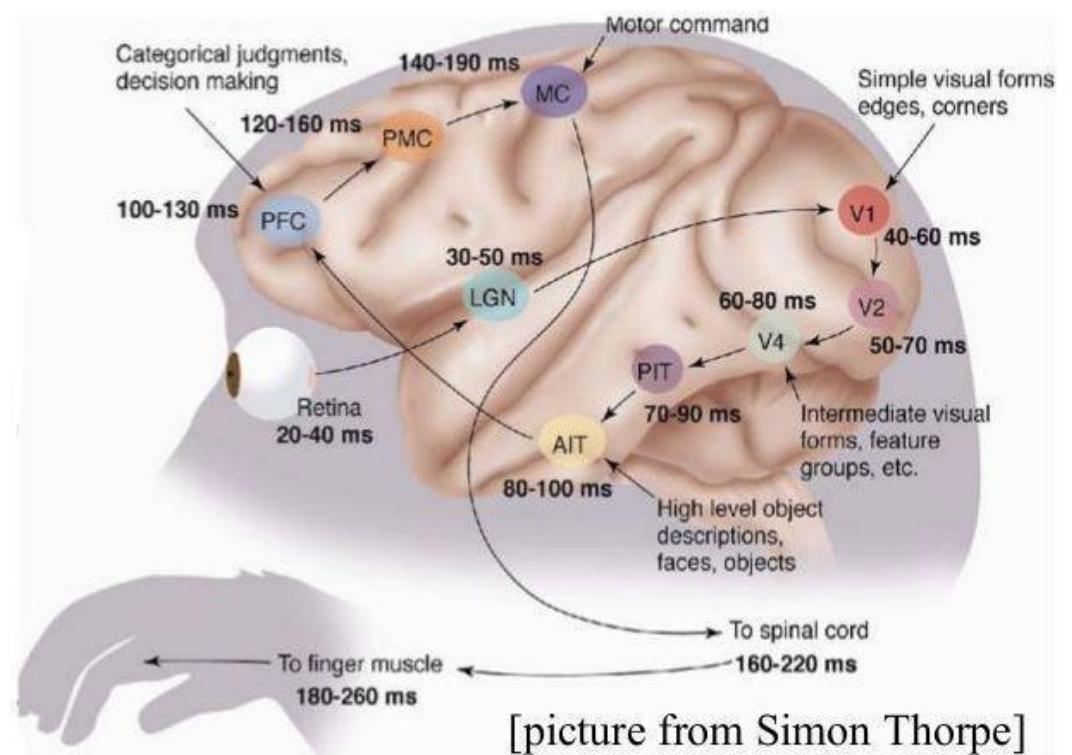
Дэвид Марр (1970e)

Дэвид Кортни Марр — британский нейробиолог и психолог. Марр работал на стыке психологии, искусственного интеллекта и нейрофизиологии, разрабатывая новые вычислительные модели обработки визуальной информации в мозге.

Исследуя компьютерное моделирование зрения, пришел к выводу о необходимости допустить этап "первоначального наброска" (primal sketch). На этом этапе световые шаблоны, воздействуя на рецепторы, преобразуются в кодированные описания линий, точек или граней вместе с их расположением, ориентацией и цветом.

Марр выделяет трехступенчатый процесс восприятия трехмерных объектов.

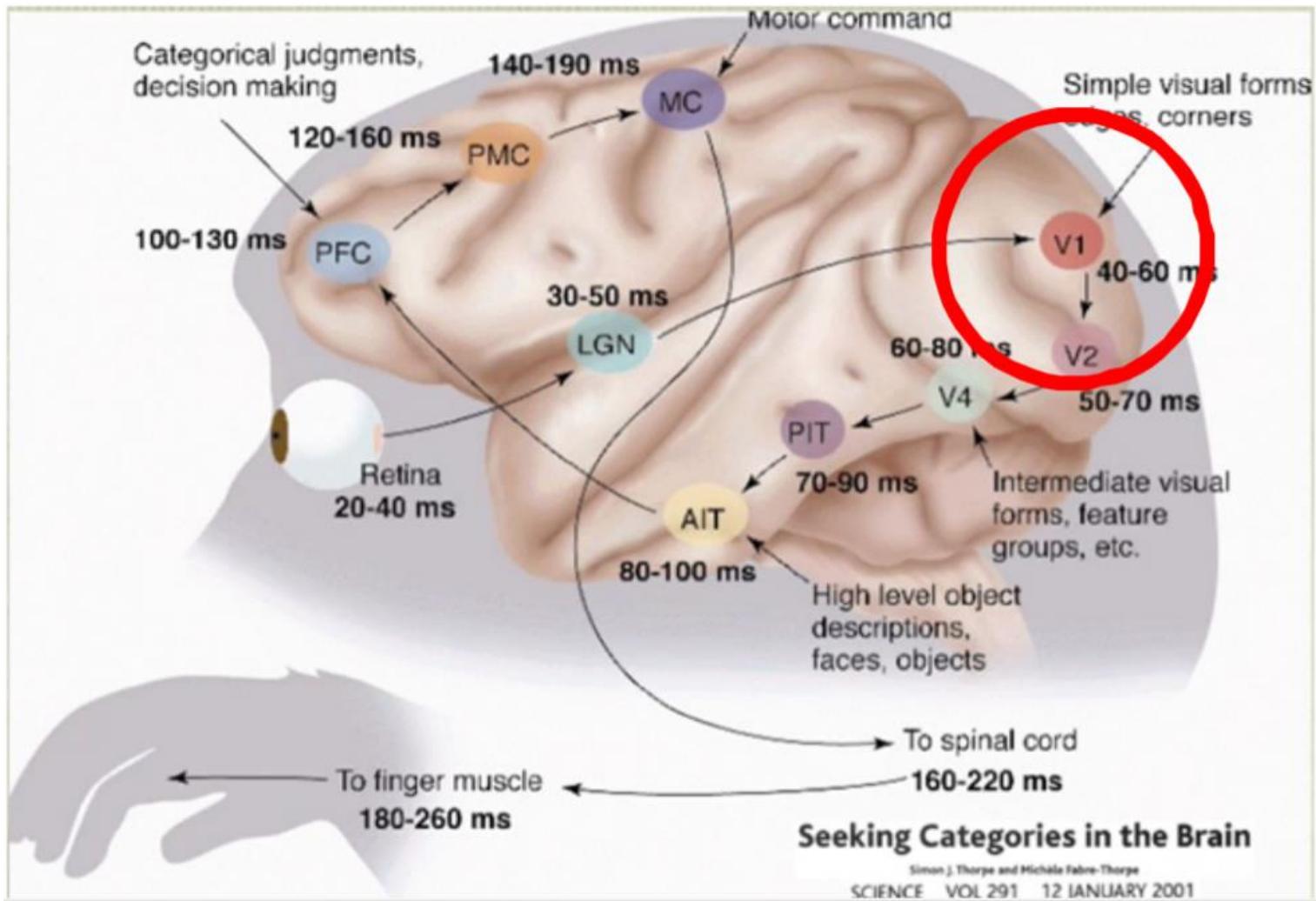
- **На первом этапе** формируется двухмерный первоначальный набросок сенсорной информации, поступающей на сетчатку глаза (Primal sketch).
- **На втором этапе** создается псевдотрехмерная картина объектов на основе признаков глубины и др (2.5D sketch). На втором этапе отражаются не все характеристики трехмерности.
- **На третьем этапе** создается полноценная трехмерная картина ориентации объектов и их взаимного расположения (3D model).



На иллюстрации показано, как во времени происходит процесс обработки зрительного сигнала. Свет попадает на сетчатку, там он преобразуется в электрические импульсы и через 20–40 миллисекунд “изображение” передается дальше. **Первый уровень** обработки занимает 10–20 миллисекунд (на картинке время показано нарастающим итогом, т.е., например, до выдачи команды моторной системе проходит всего 140-190 мс). **На втором уровне** обработки, ещё через 20–30 миллисекунд, сигнал приходит на нейроны, которые распознают простые локальные формы.

Потом еще один уровень, ещё один уровень, и вот на **четвёртом уровне** мы уже видим промежуточные формы: можно найти нейроны, которые “зажигаются” при виде квадратов, градиентов цветов или других подобных объектов.

Потом ещё пара уровней нейронов, и вот, через 100 миллисекунд от начала процесса, у нас уже зажигаются нейроны, отвечающие высокоуровневым объектам. Например, если вы познакомились с человеком, у вас в мозге появляется нейрон, отвечающий за его лицо; более того, скорее всего появляется и нейрон или группа нейронов, “отвечающая” за этого человека в целом и срабатывающая при любом контакте с ним, в том числе письменном, когда лица не видно. Если вы увидите лицо снова (а нейрон за это время не разобучится обратно, «забыв» прежнюю информацию), то через ~100 миллисекунд в вашей голове активируется тот самый нейрон.

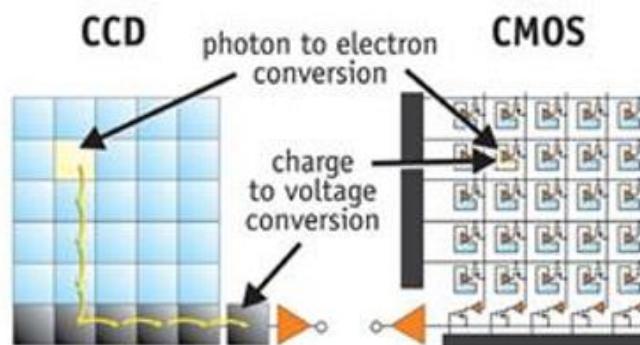


Современная камера



Тот же самый принцип, но с пленкой,
объективом и т.д.

Цифровая камера - дискретизация



CCDs move photogenerated charge from pixel to pixel and convert it to voltage at an output node. CMOS imagers convert charge to voltage inside each pixel.

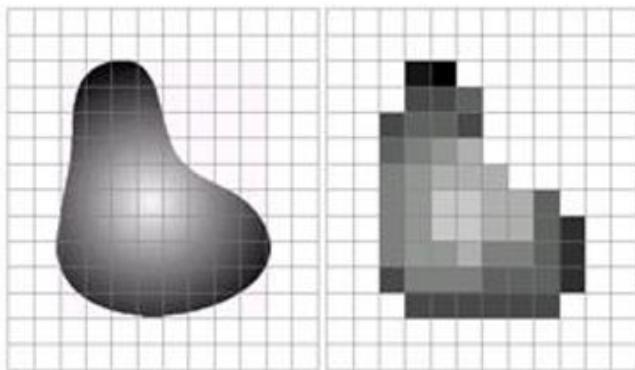
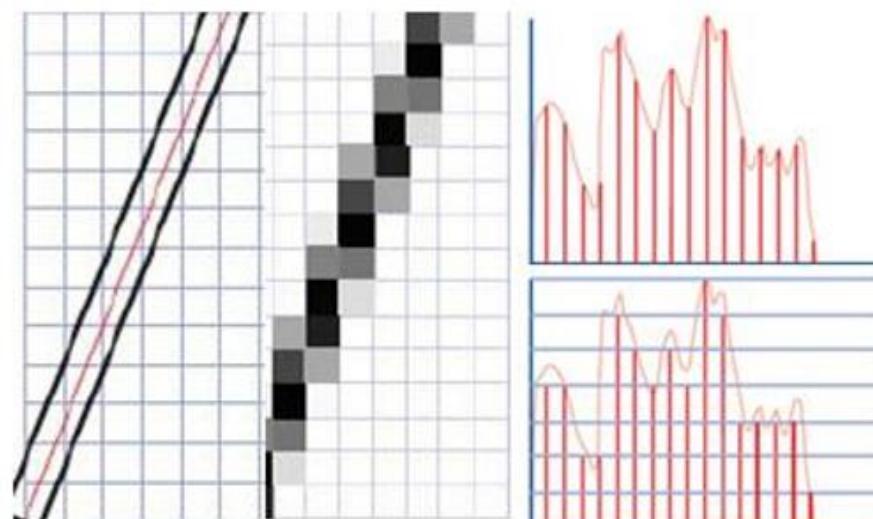


FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

По пространству



По яркости

Изображение

Изображение оптическое – картина, получаемая в результате прохождения через оптическую систему лучей, распространяющихся от объекта, и воспроизводящая его контуры и детали.

Физический энциклопедический словарь.

Функция интенсивности (яркости) канала,
заданная на 2x мерной сетке (матрице)

$$I = g(x, y), \{x \in [x_0, x_1], y \in [y_0, y_1]\}$$

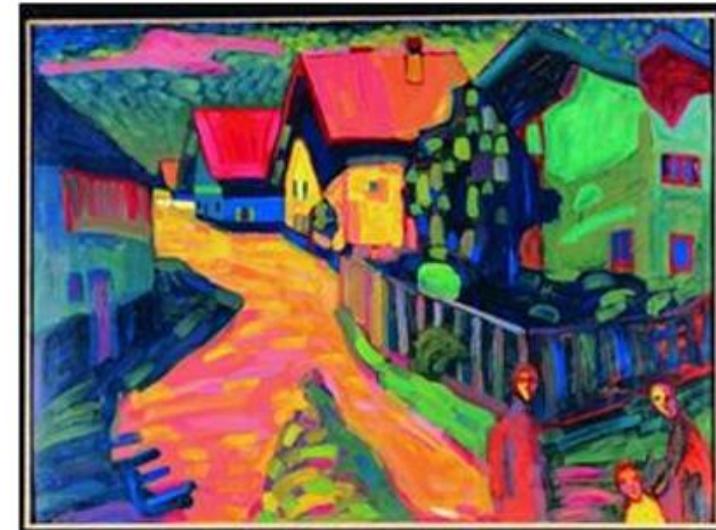
Используется дискретное представление

$$I = g(i, j), \{i = \overline{1, n}, j \in \overline{1, m}\}$$

Что такое цвет?

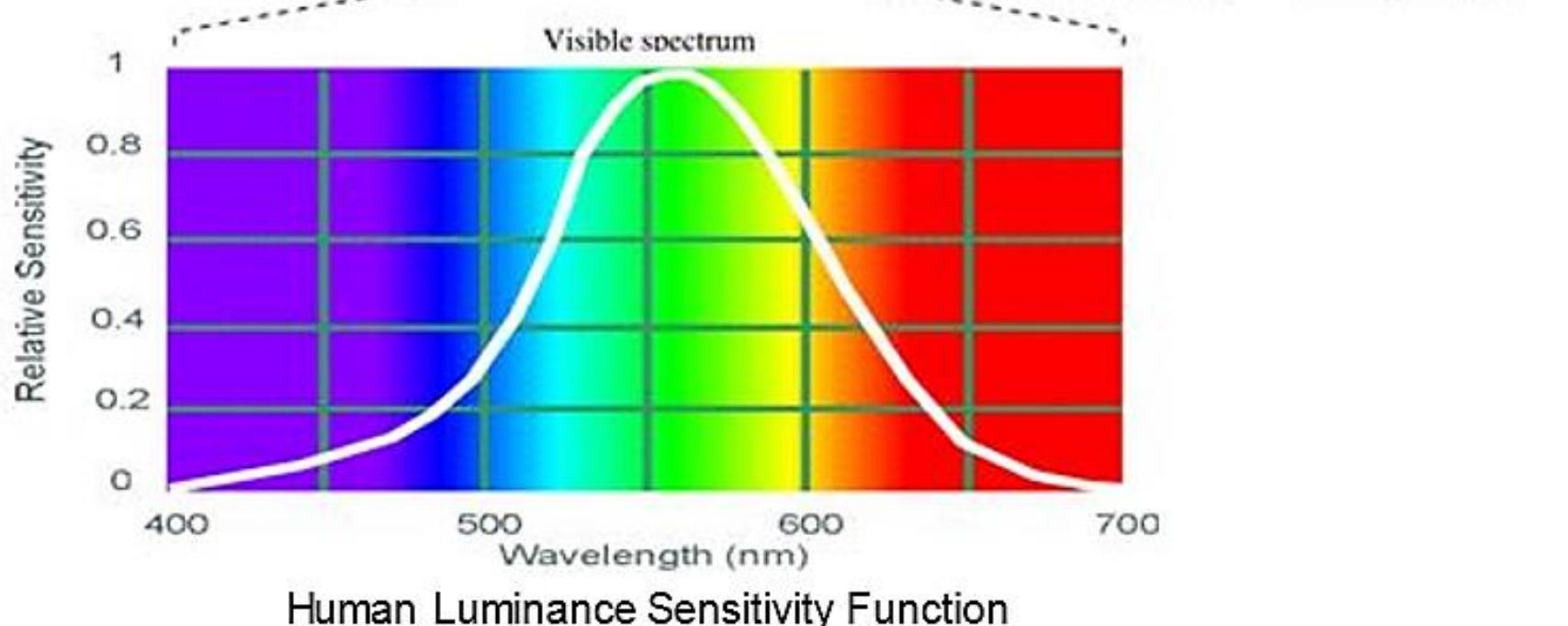
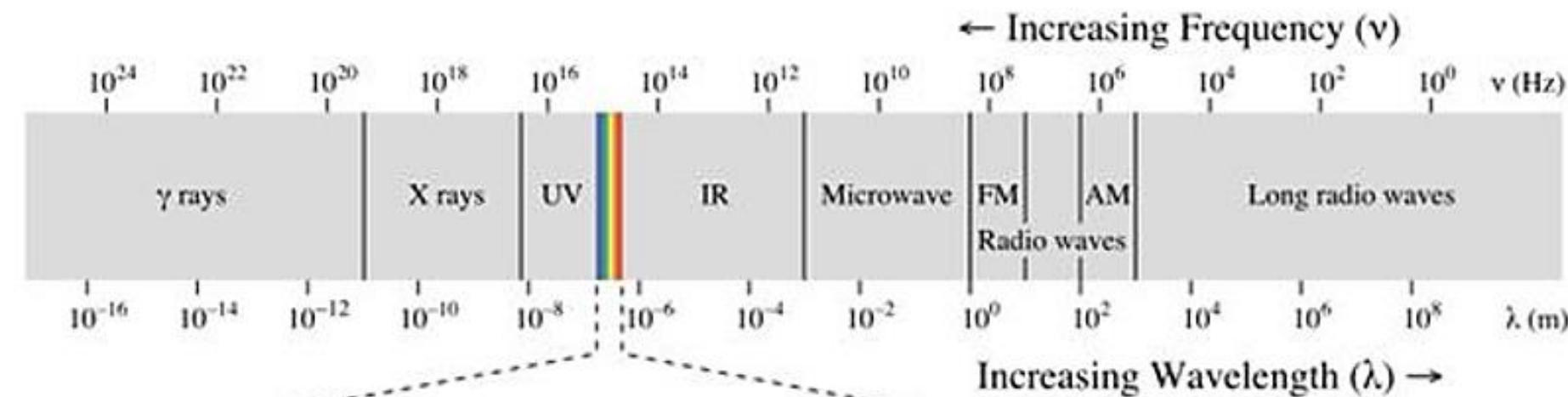
Цвет – это психологическое свойство нашего зрения, возникающее при наблюдении объектов и света, а не физические свойства объектов и света
(S. Palmer, *Vision Science: Photons to Phenomenology*)

Цвет – это результат взаимодействия света, сцены и нашей зрительной системы



Wassily Kandinsky (1866-1944), Murnau Street with Women, 1908

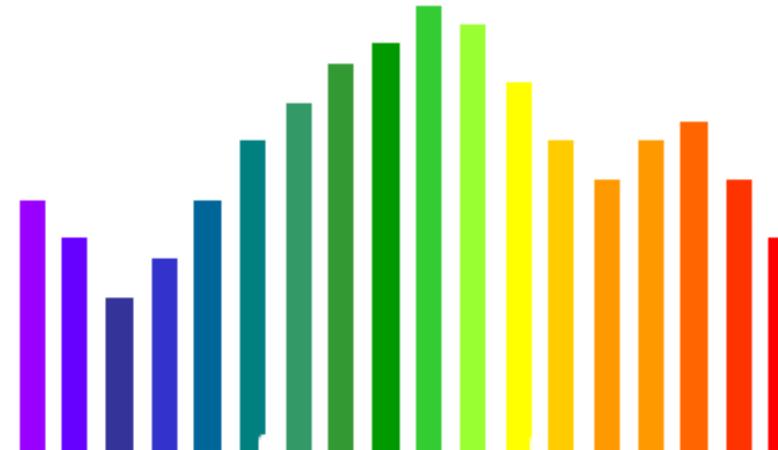
Электромагнитный спектр



Физика света

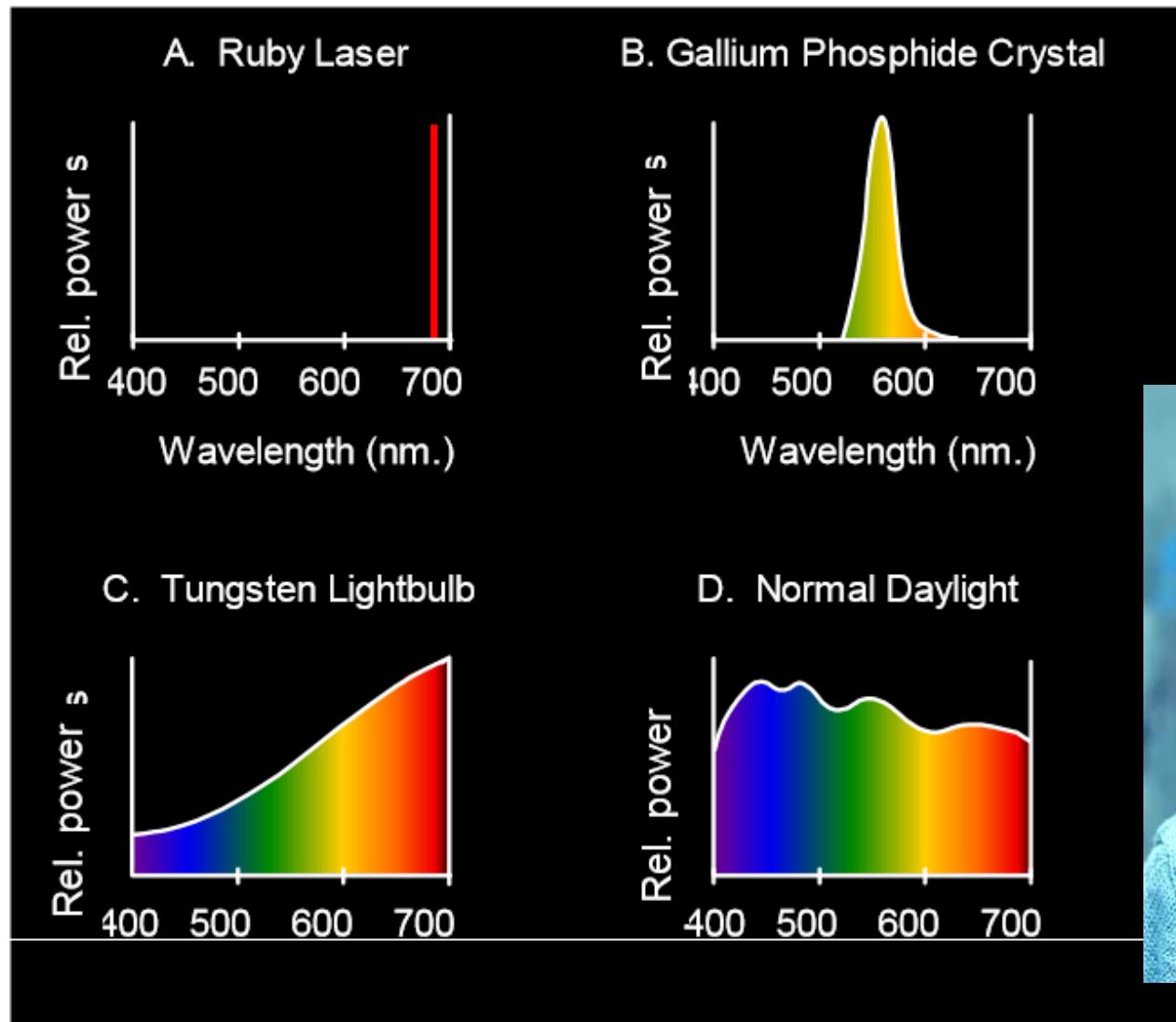
Любой источник света можно полностью описать спектром: количество излученной энергии в единицу времени для каждой длины волны в интервале 400 - 700 нм.

Relative
spectral
power



Физика света

Примеры спектров разных источников света



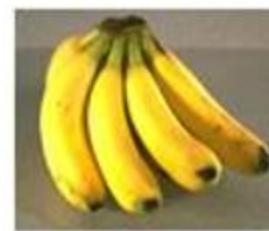
В спектре дневного света преобладает синий, поэтому в действительности при дневном освещении все окрашено в голубой, но наш мозг адаптирует цвета.

Хроматическая адаптация

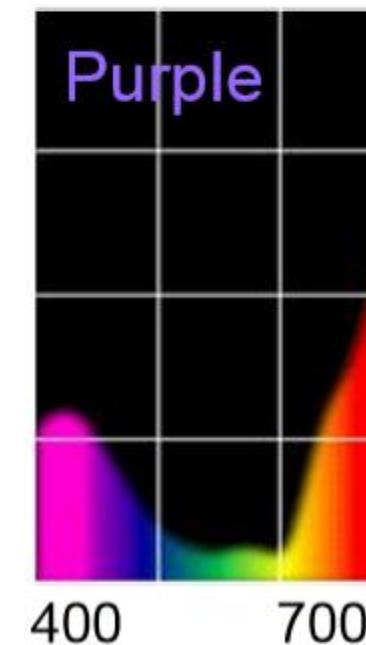
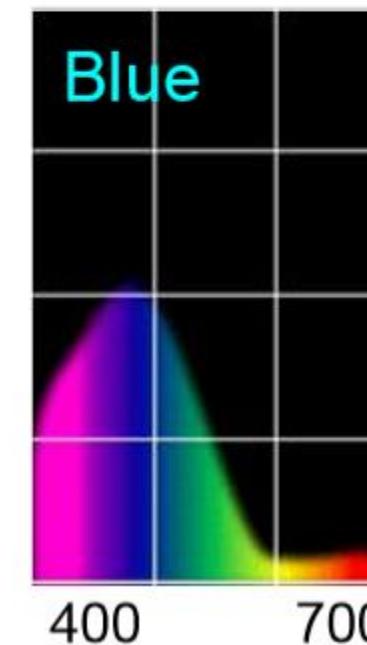
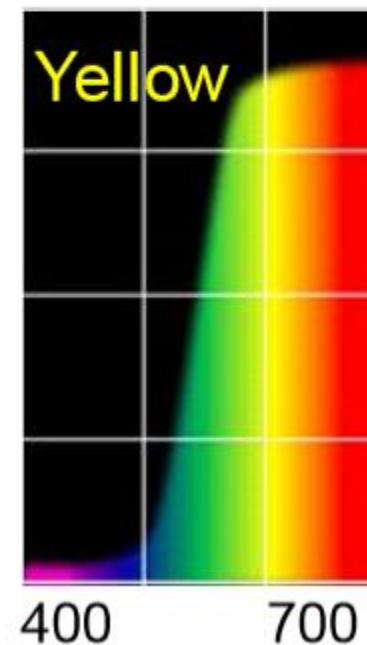
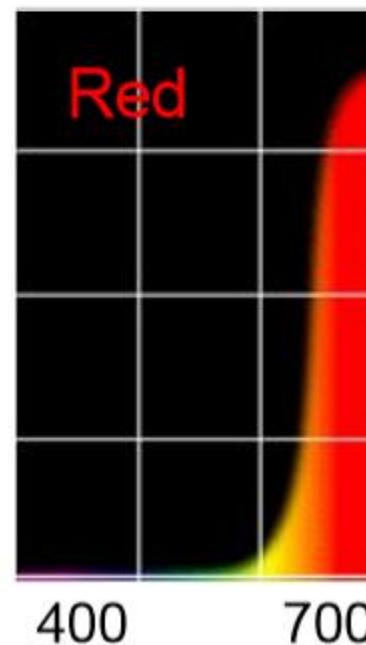


Физика света

Примеры спектров отраженного света от предметов



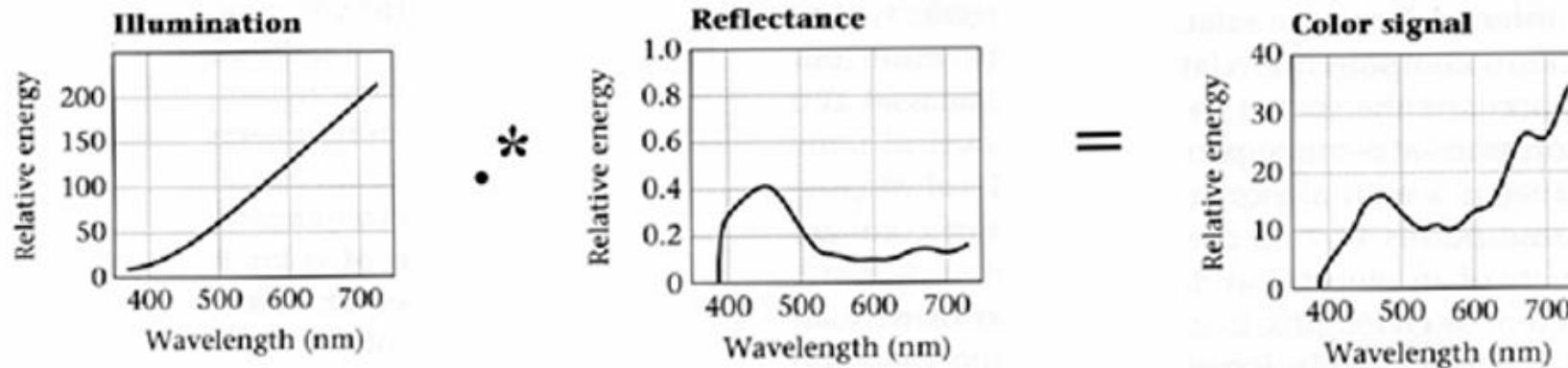
% Отраженного света



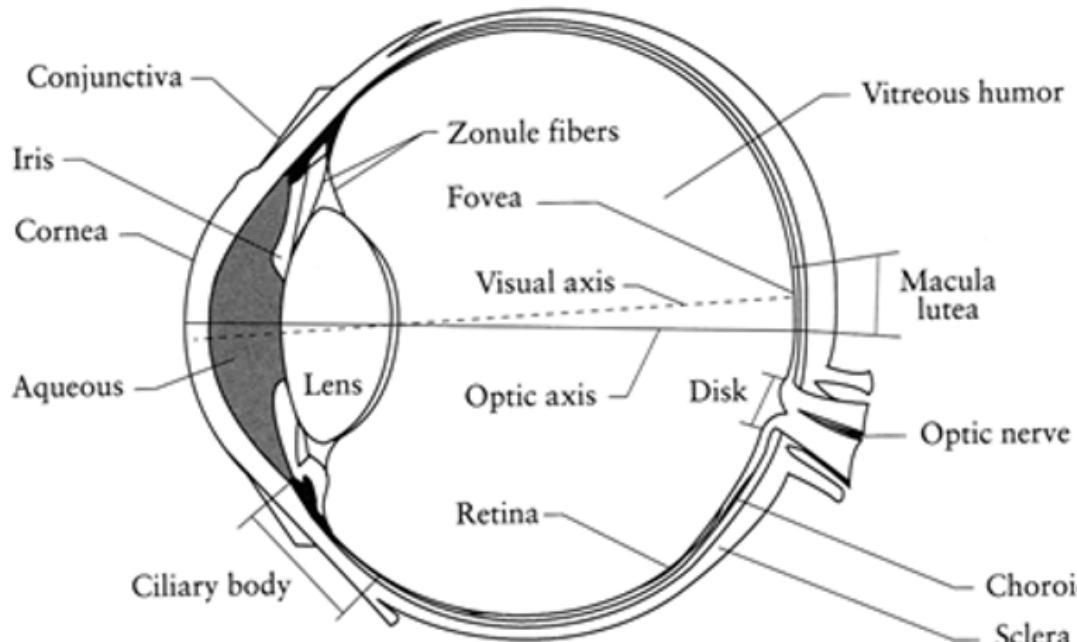
Взаимодействие света и объектов



Видимый цвет это
результат взаимодействия
спектра излучаемого света
и поверхности

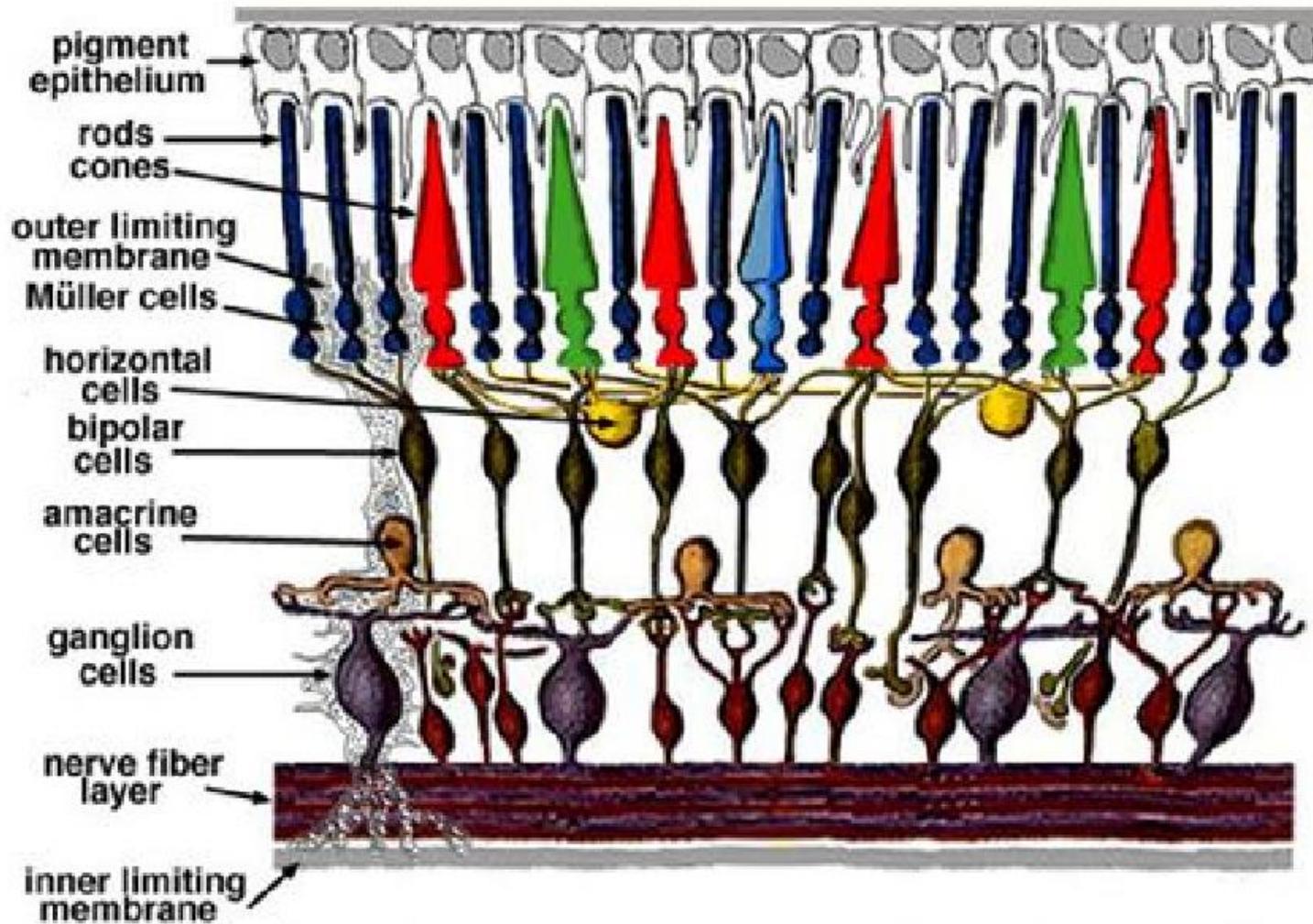


Человеческий глаз



Глаз как камера!

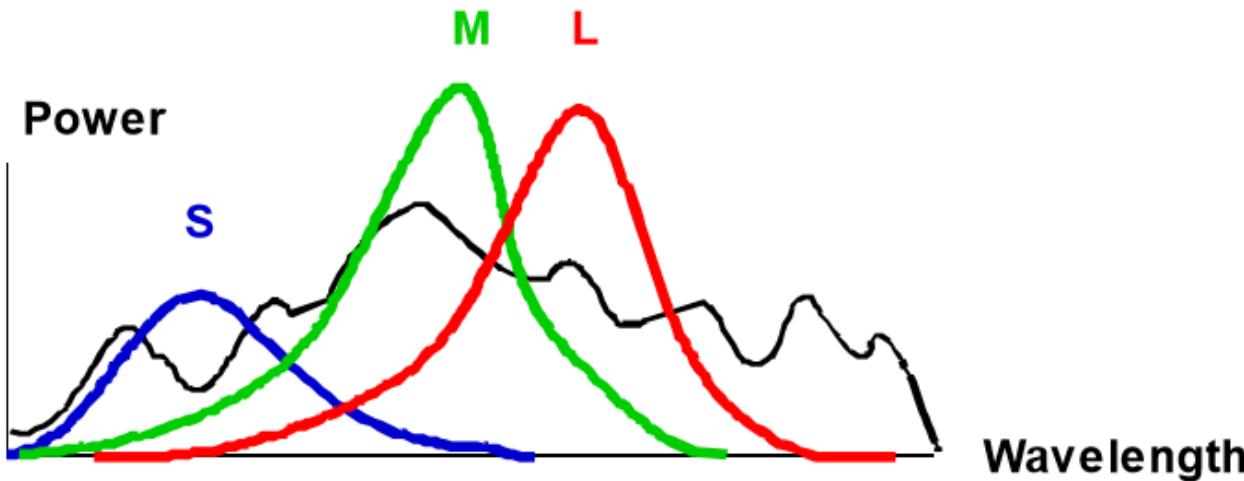
- **Радужка** – цветная пленка с радиальными мышцами
- **Зрачок** - отверстие (апертура), диаметр управляетяется радужкой
- **Хрусталик** – «линза», меняющая форму под действием мышц
- Где матрица?
 - Клетки-фоторецепторы на сетчатке



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

Свет

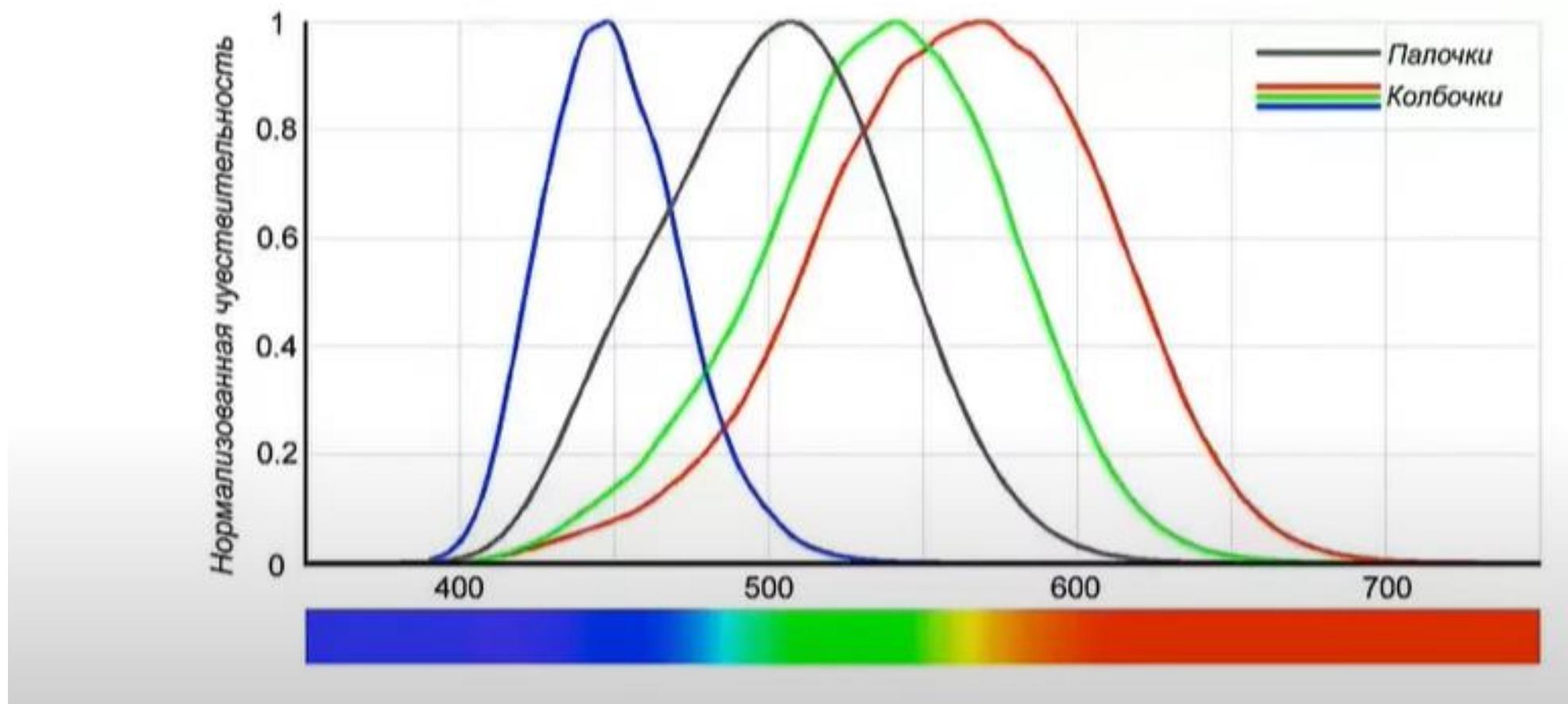
Восприятие цвета



Палочки и колбочки – фильтры спектра

- Спектр умножается на кривую отклика, производится интегрирование по всем длинам волн
 - Каждый тип колбочек даёт 1 число
- В: Как же мы можем описать весь спектр Змя числами?
- О: Мы и не можем! Большая часть информации теряется.
 - Два разных спектра могут быть неотличимы
 - » Такие спектры называются **метамеры**

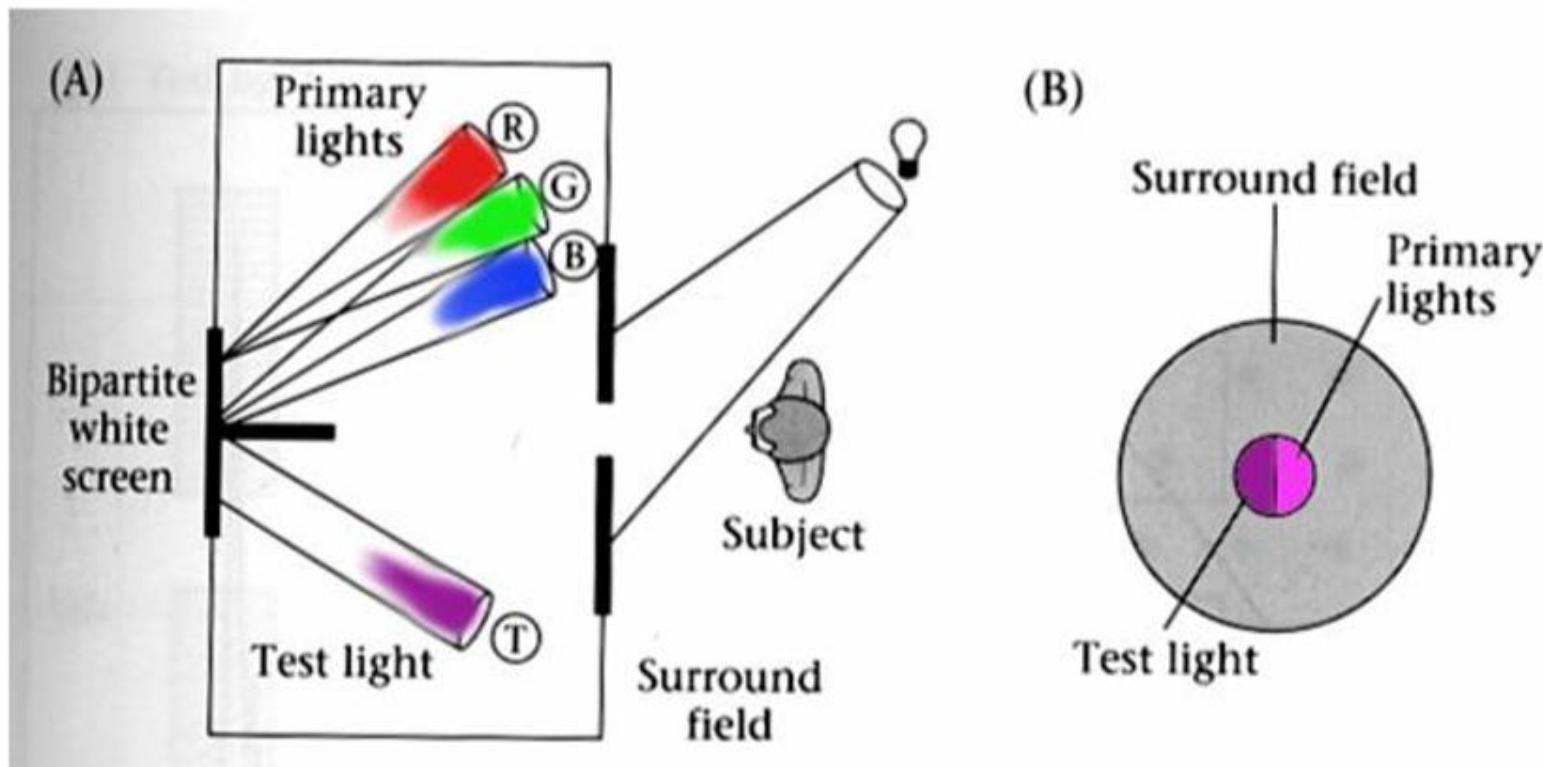
Палочки и колбочки



Стандартизация восприятия цвета

Мы хотим понять, какие спектры света вызывают
одинаковые цветовые ощущения у людей

Эксперименты по сопоставлению цвета



Трихроматическая теория цвета

В экспериментах по сопоставлению цвета большинству людей достаточно 3х основных цветов, чтобы сопоставить любой цвет

- Основные цвета должны быть независимы

Для одного и того же спектра, и одних и тех же основных цветов, люди выбирают одинаковые веса

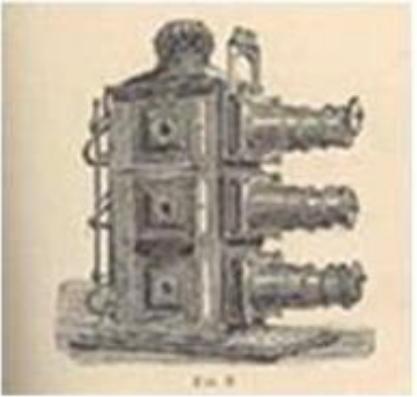
- Исключения: цветовая слепота

Трихроматическая теория

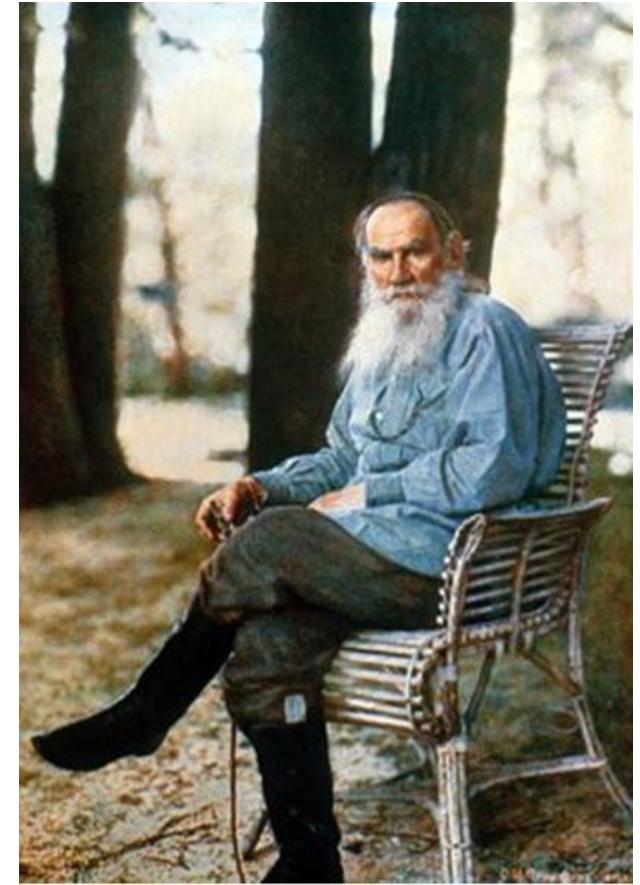
- Трех чисел оказывается достаточно, чтобы описать цвет
- История восходит к 18^у веку (Томас Юнг)

Цветные фотографии

Сергей Прокудин-Горский (1863-1944)
Фотографии Российской империи(19.09.1916)



Lantern
projector



Обработка изображений

Обработка изображений

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

Цели

1. Улучшение изображения **для восприятия человеком**
 - чтобы стало «лучше» с субъективной точки зрения человека
2. Улучшение изображения **для восприятия компьютером**
 - упрощение последующего распознавания
3. **Развлечение** (спецэффекты)
 - получить эстетическое удовольствие от красивого эффекта

Обработка изображений

Можно разделить на:

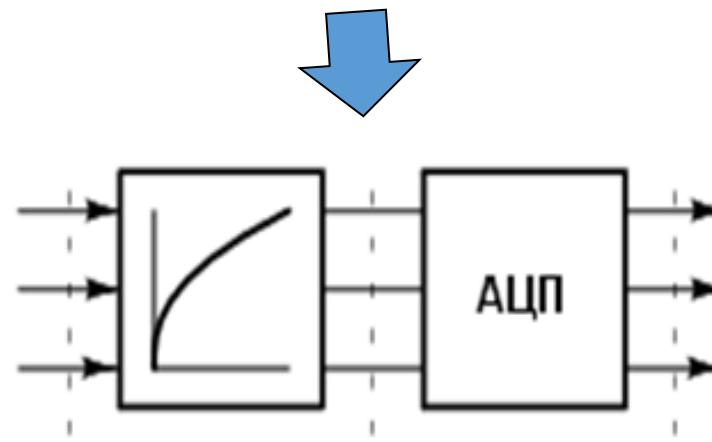
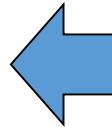
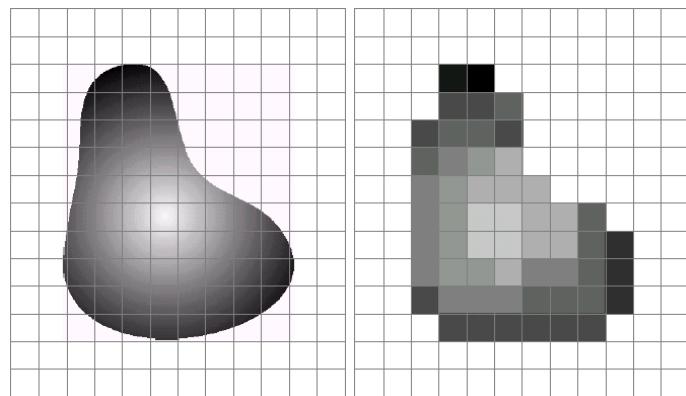
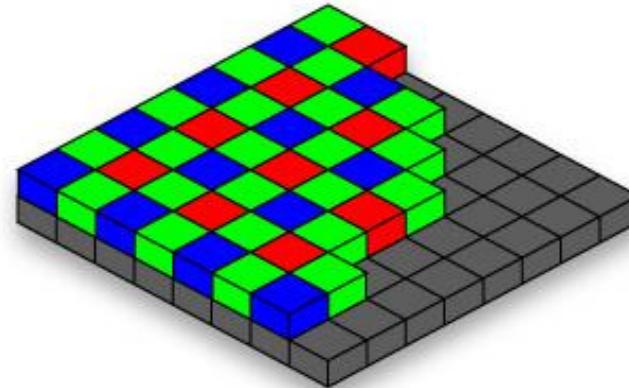
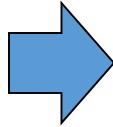
- **задачи фильтрации** (удаление шумов, повышение чёткости, осветление тёмных участков);
- **задачи анализа** (расчёт градиентов, поиск особых точек и границ объектов).

В процессе создания систем компьютерного зрения **фильтрация** изображения обычно предшествует извлечению более высокого уровня информации о сцене.

Назначение фильтрации

- Устранение шума в изображениях
- Улучшение качества изображения
- Усиление полезной и подавления нежелательной информации

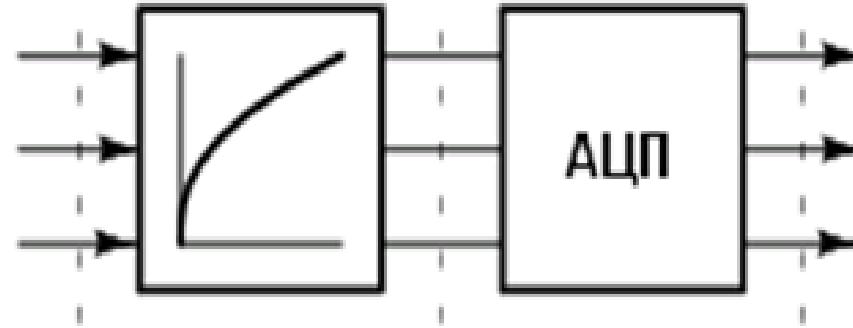
Цифровое изображение



Вспоминаем процесс получения цифрового изображения...

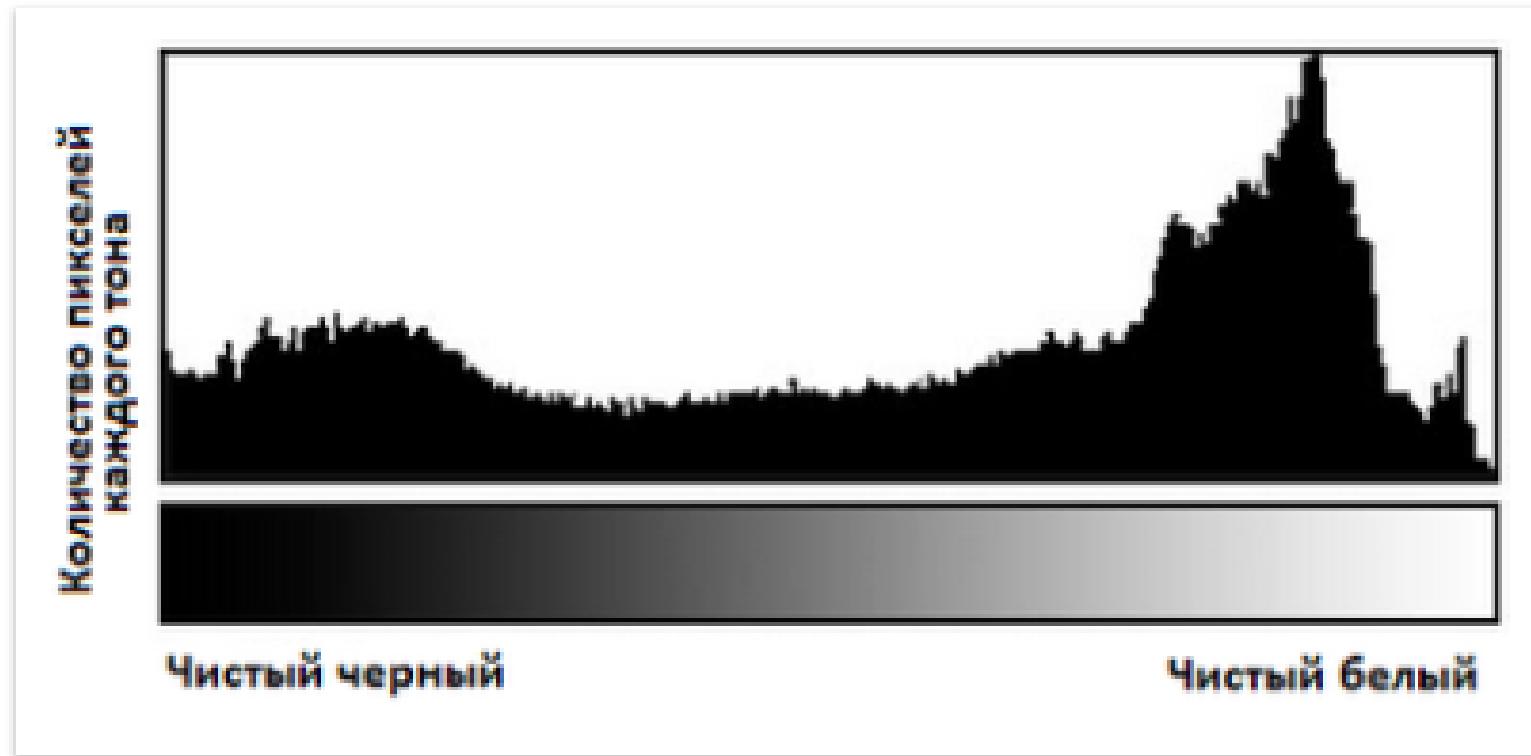
Почему оно может получиться плохо?

- Ограниченный диапазона чувствительности датчика
- “Плохие” функции передачи датчика



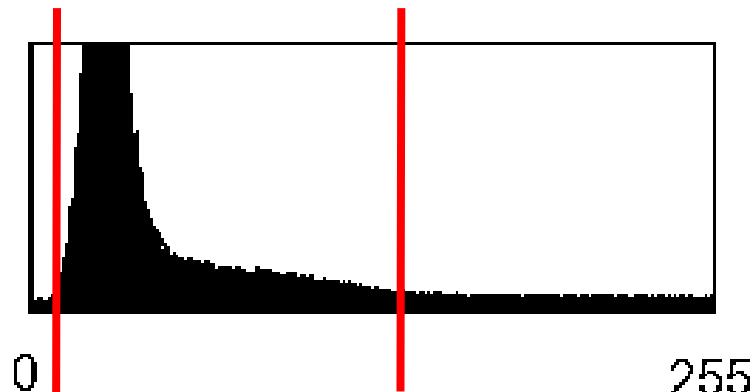
Что такое гистограмма изображения?

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

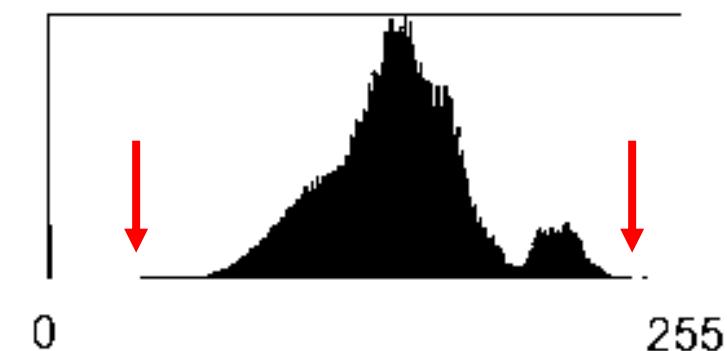


Что может не устраивать в изображении:

- Узкий или смещенный диапазон яркостей пикселей (тусклое или «пересвеченное» изображение)



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



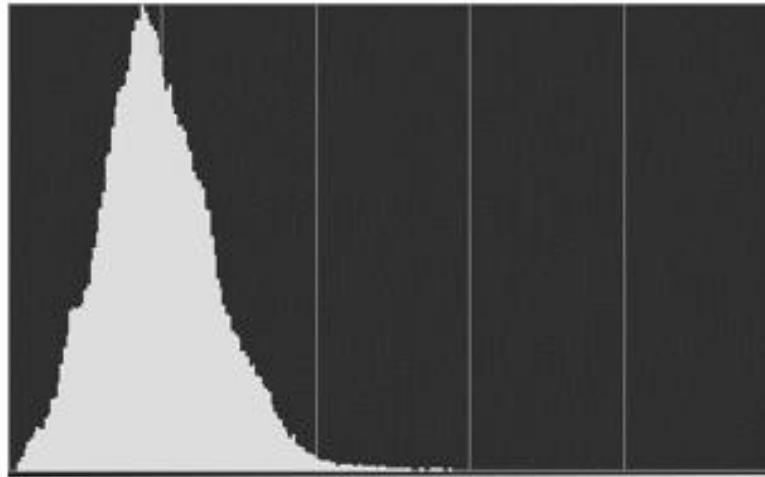
При коррекции к изображению применяется преобразование яркостей, компенсирующий нежелательный эффект:

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

Затемненное изображение (а) и соответствующая ему гистограмма (б);
засвеченное изображение (в) и соответствующая ему гистограмма (г).



а



б



в



г

Изменение контраста изображения

Существует три основных метода повышения контраста изображения:

1. **Линейная растяжка гистограммы** (линейное контрастирование),
2. **Нормализация гистограммы**,
3. **Выравнивание** (линеаризация или эквалайзация, equalization) гистограммы.

1. Линейная коррекция

Компенсация узкого диапазона яркостей – линейное растяжение:

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

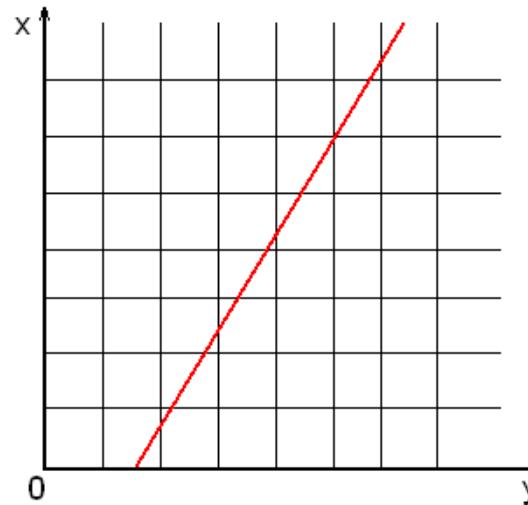
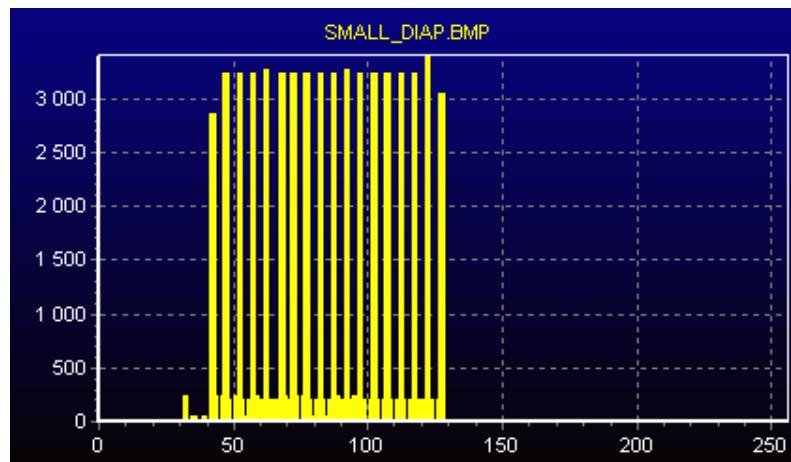
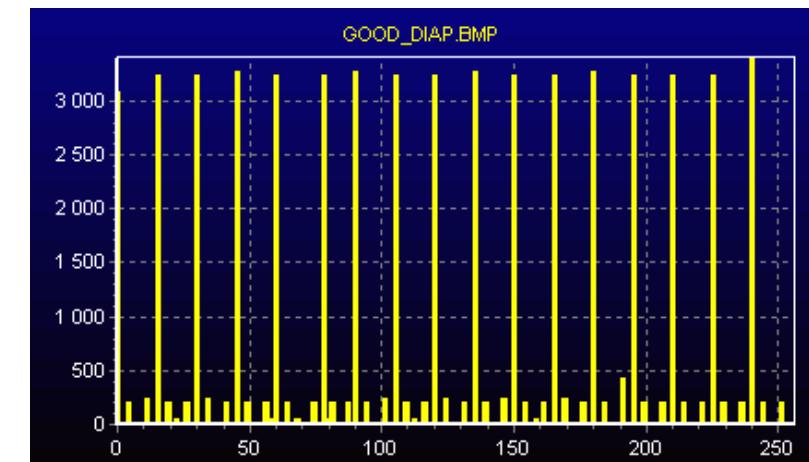


График функции $f^{-1}(y)$



Линейная коррекция

Линейное растяжение – «как AutoContrast в Photoshop»



НО линейная коррекция помогает не всегда!



Почему?

Нелинейная коррекция

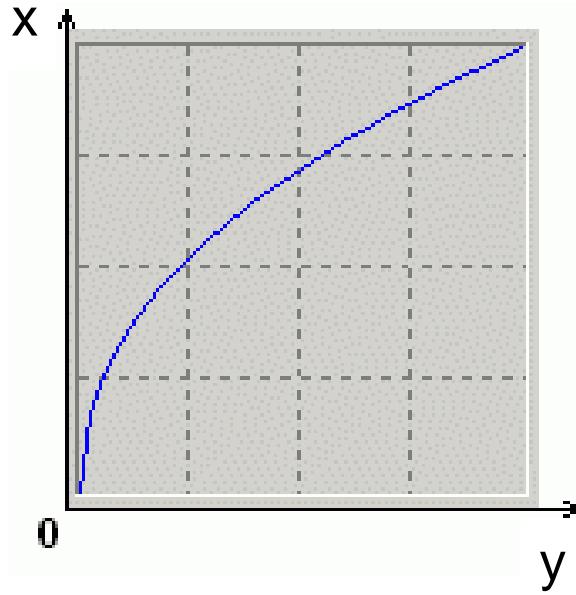


График функции $f^{-1}(y)$

Нелинейная коррекция

Нелинейная компенсация недостаточной контрастности

Часто применяемые функции:

- **Гамма-коррекция**

Изначальная цель – коррекция для правильного отображения на мониторе.

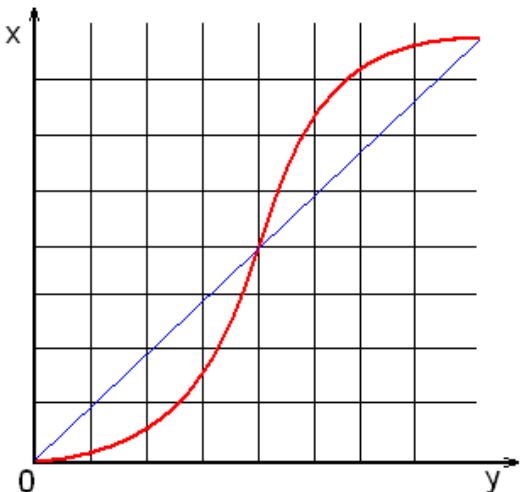
$$y = c \cdot x^\gamma$$

- **Логарифмическая**

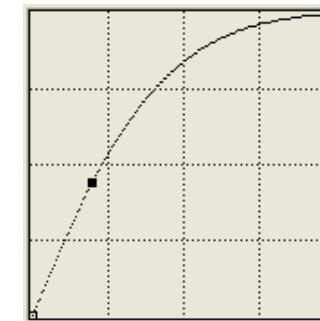
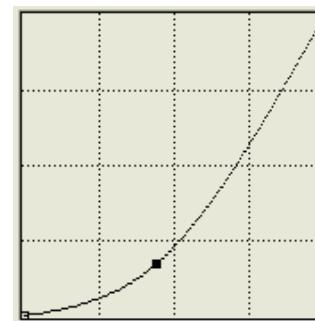
Цель – сжатие динамического диапазона при визуализации данных

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

Гамма-коррекция



Инструмент «Кривые» в Photoshop

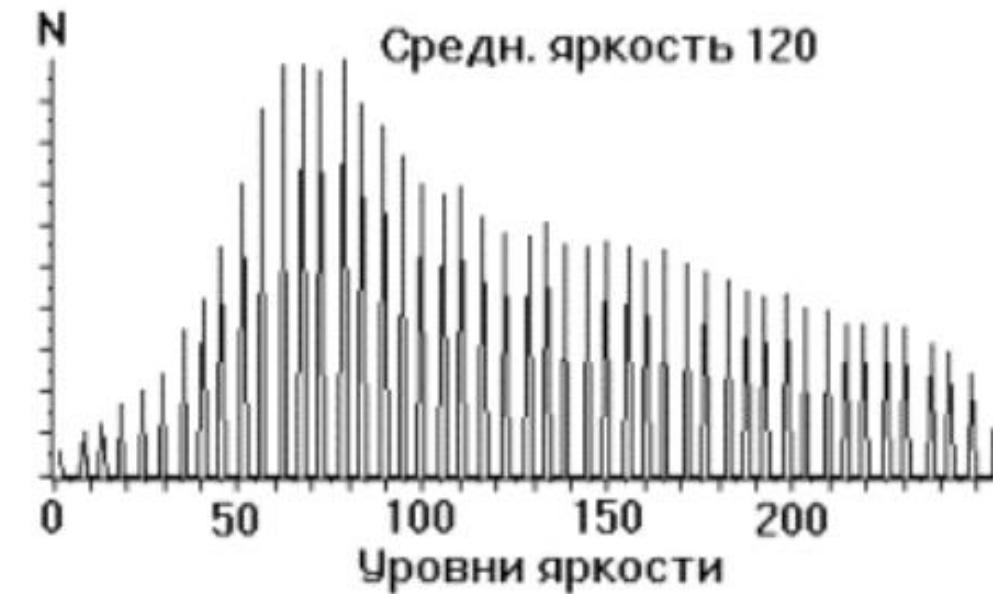
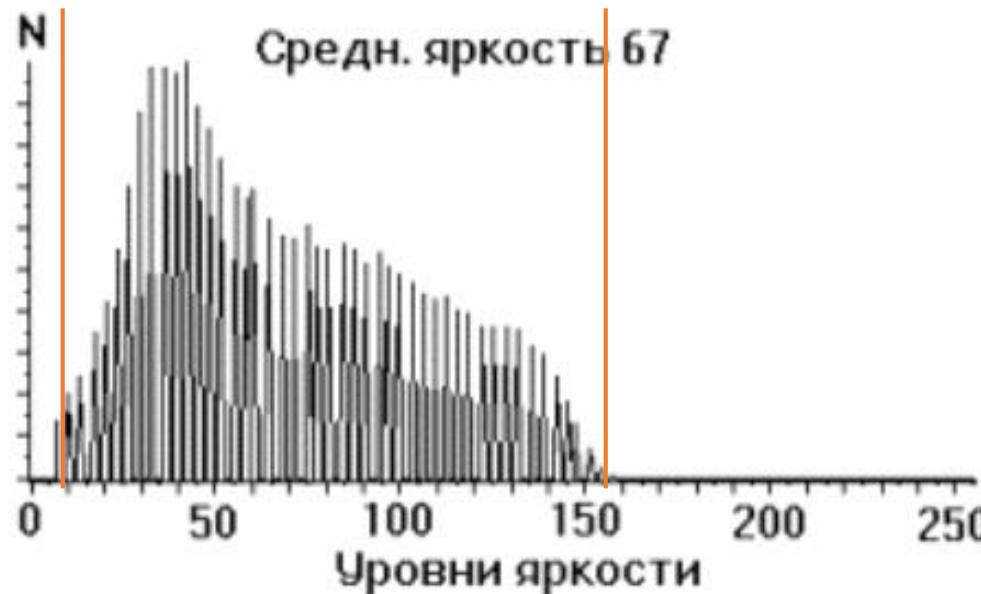
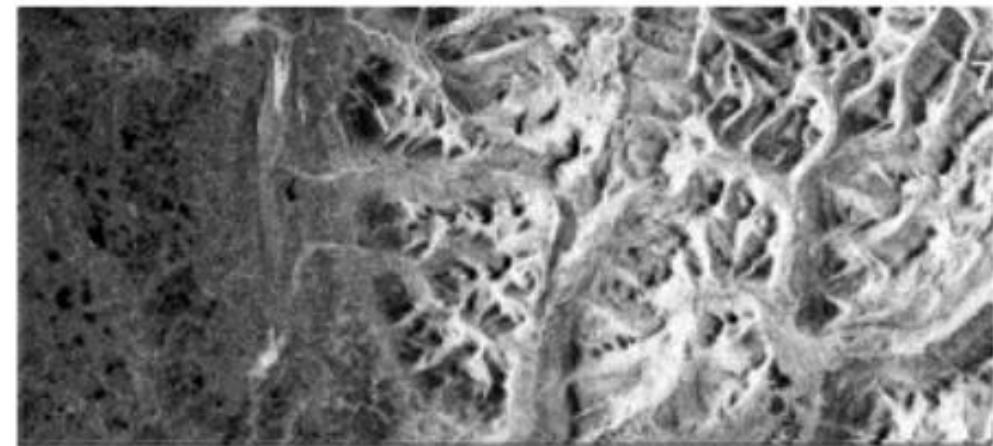
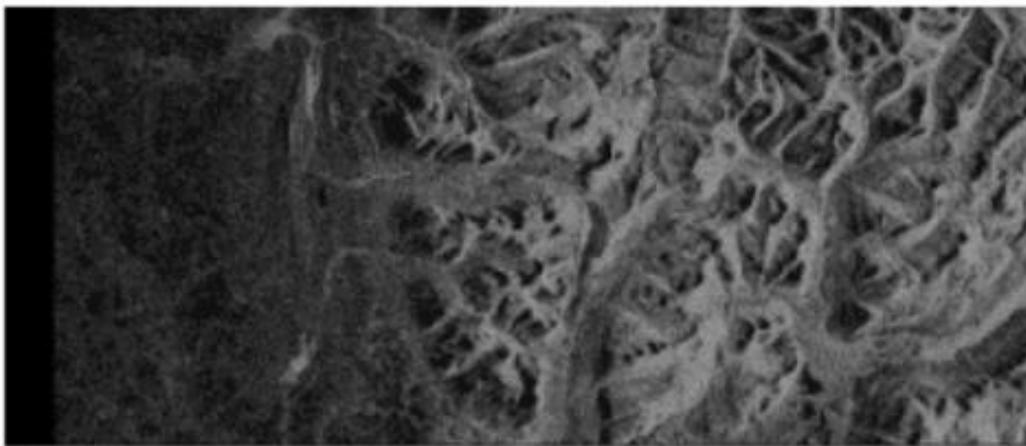


Графики функции $f^{-1}(y)$

2. Нормализация гистограммы

Нормализация гистограммы в отличие от линейной коррекции обеспечивает растяжку **не всего диапазона изменения интенсивностей, а только его наиболее информативной части.** Под информативной частью понимается набор пиков гистограммы, т.е. интенсивности, которые чаще остальных встречаются на изображении. Бины, соответствующие редко встречающимся интенсивностям, в процессе нормализации отбрасываются, далее выполняется обычная линейная растяжка получившейся гистограммы.

2. Нормализация гистограммы



3. Выравнивание гистограммы (эквалазия)

Цель выравнивания состоит в том, чтобы все уровни яркости имели бы одинаковую частоту, а гистограмма соответствовала равномерному закону распределения.

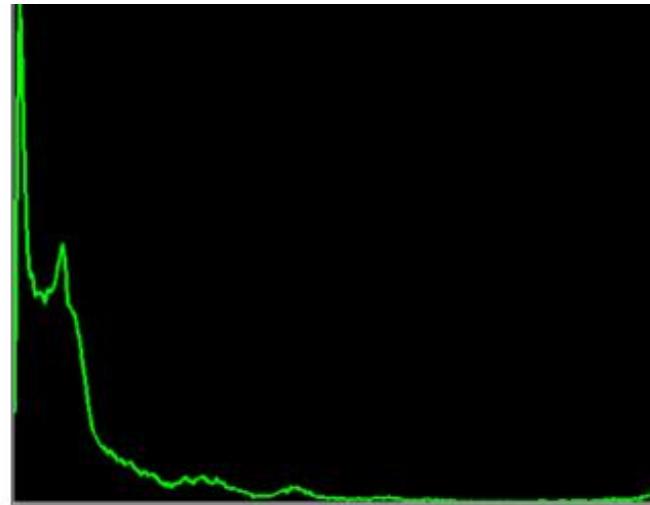
```
void equalizeHist(const Mat& src, Mat& dst)
```



Пример выравнивание гистограммы



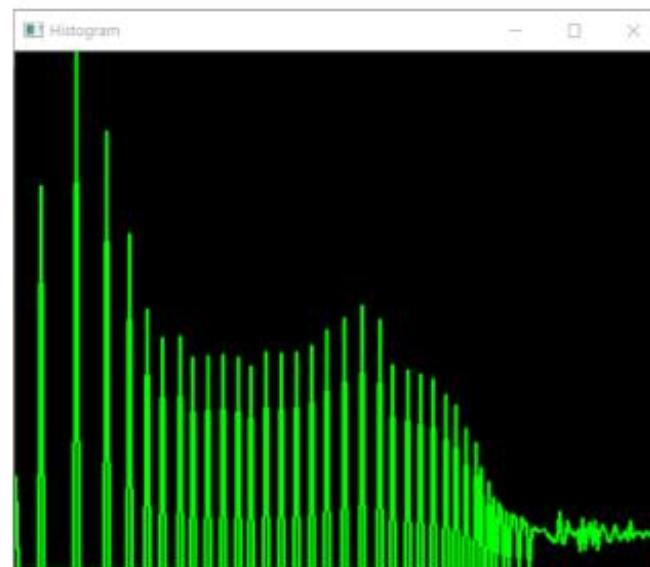
а



б



в



г

Цветовая коррекция

Изменение цветового баланса для компенсации:

- Неверного цветовосприятия камеры
- Цветного освещения



Цветовая коррекция изображений

Растяжение контрастности (“autolevels”)

Идея – растянуть интенсивности по каждому из каналов на весь диапазон;

Реализация:

- 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



Исходное изображение



Эквализация RGB

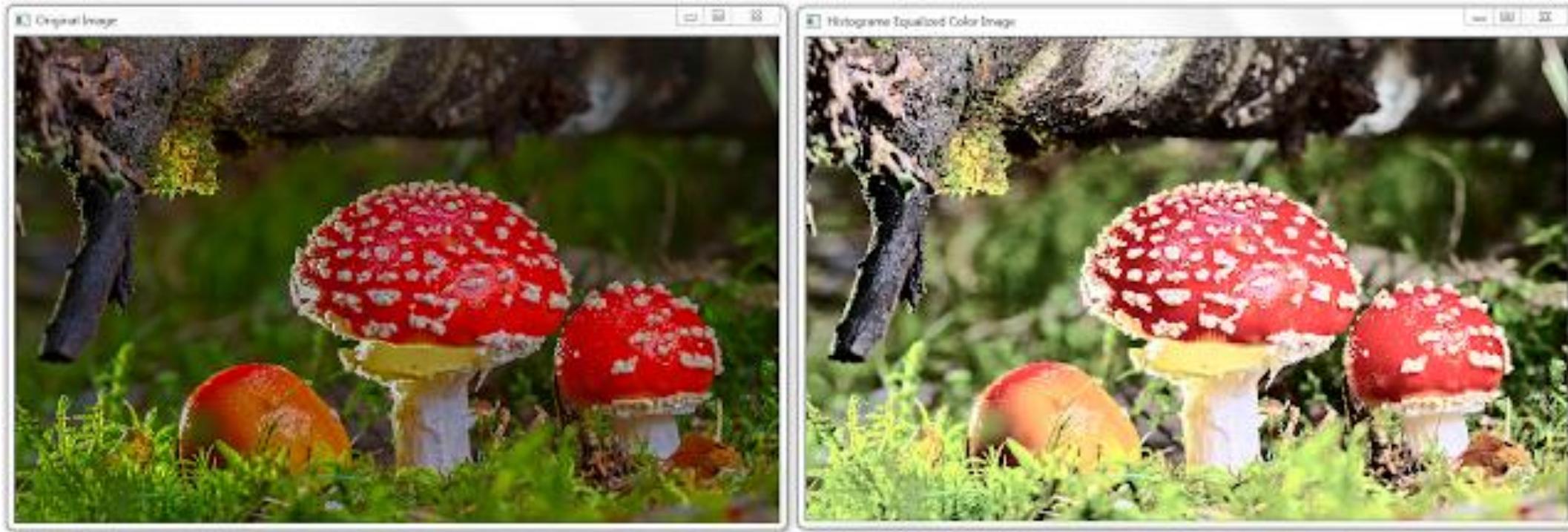


Эквализация интенсивности



Применение эквалазии к трем каналам **RGB** как правило приводит к не лучшим результатам. Цвета получаются холодными и неестественными. Эквалазия по интенсивности в модели **HSL** приводит к лучшим результатам.

HSI или HSL (от англ. hue, saturation, lightness (intensity)) — цветовая модель, в которой цветовыми координатами являются тон, насыщенность и светлота(итенсивность).



OpenCV загружает цветные изображения в цветовом пространстве BGR. В этом цветовом пространстве невозможно выровнять гистограмму, не затрагивая информацию о цвете, поскольку все 3 канала содержат информацию о цвете. Поэтому вам необходимо преобразовать изображение BGR в цветовое пространство, такое как **YCrCb**. В цветовом пространстве YCrCb канал Y изображения содержит только информацию об интенсивности, тогда как каналы Cr и Cb содержат всю информацию о цвете изображения. Поэтому для получения изображения с гистограммой, выровненной по гистограмме, следует обрабатывать только канал Y без изменения какой-либо информации о цвете. После обработки изображение YCrCb должно быть преобразовано обратно в цветовое пространство BGR перед вызовом функции `imshow()`.

Шумоподавление

Причины возникновения шума:

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



Шум фотоаппарата



Сильное сжатие JPEG



Подавление шума

Пусть дана камера и статичная сцена, требуется подавить шум. Шум случайная величина.



Простейший вариант: усреднить несколько кадров

Усреднение

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

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

“box filter”

Пример самого простого фильтра для шумоподавления

Определение свертки

Будем считать, что задано исходное полутоновое изображение I , и обозначим интенсивности его пикселей $I(x, y)$. Линейный фильтр определяется вещественнозначной функцией F , заданной на растре. Данная функция называется ядром фильтра, а сама фильтрация производится при помощи операции дискретной свертки (взвешенного суммирования) по следующей формуле:

$$I'(x, y) = \sum_i \sum_j F(i, j) \cdot I(x + i, y + j)$$

Свёртка — это операция **вычисления нового значения** заданного пикселя, при которой **учитываются значения окружающих его соседних пикселей**.

Главным элементом свёртки является т.н. **ядро свёртки** — это матрица (произвольного размера и отношения сторон; чаще всего используется квадратная матрица (по-умолчанию, 3x3)).

Определение свертки

Ядро свёртки – это числовая матрица с подобранными коэффициентами. Эти коэффициенты называют степенями влияния или “ценностями”, определяющими вес пикселей из рассматриваемой окрестности.

Входное изображение

12	14	41
43	84	24
2	12	12

×

Ядро фильтра

0,5	0,75	0,5
0,75	1,0	0,75
0,5	0,75	0,5

=

Результат

$$= \left[\begin{array}{l} 12 \cdot 0,5 + 14 \cdot 0,75 + 41 \cdot 0,5 + \\ + 43 \cdot 0,75 + 84 \cdot 1,0 + 24 \cdot 0,75 + \\ + 2 \cdot 0,5 + 12 \cdot 0,75 + 12 \cdot 0,5 \end{array} \right] \times \frac{1}{\text{div}} = 31,2$$

Коэффициент нормирования $\text{div} = 6$

Источник [Матричные фильтры обработки изображений / Хабр \(habr.com\)](#)

Хорошая визуализация процессов свертки [Визуальное объяснение ядер изображений \(setosa.io\)](#)

Для вычисления сверток в библиотеке OpenCV присутствует функция **filter2D()**:

```
void filter2D(const Mat& src, Mat& dst, int ddepth,  
             const Mat& kernel, Point anchor=Point(-1, -1),  
             double delta=0, int borderType=BORDER_DEFAULT)
```

```
const float kernelData[] = { -1.0f, -1.0f, -1.0f,  
                            -1.0f, 9.0f, -1.0f,  
                            -1.0f, -1.0f, -1.0f };  
const Mat kernel(3, 3, CV_32FC1, (float*)kernelData);  
filter2D(src, dst, -1, kernel);
```

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

Какое изображение
получим в итоге
применения фильтра с
таким ядром

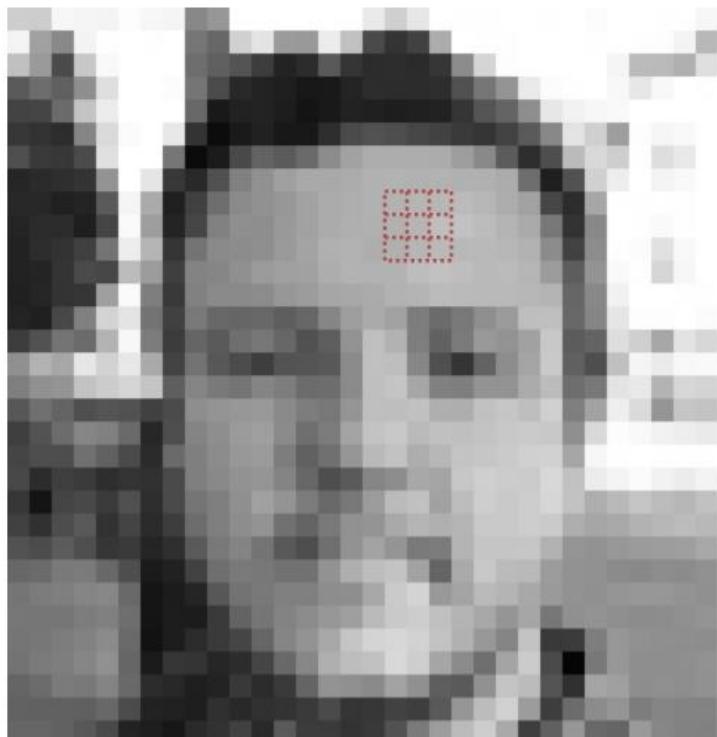
?

$$\begin{matrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{matrix}$$

Какое изображение
получим в итоге
применения фильтра с
таким ядром

?

повышения контраста



input image

$$\left(\begin{array}{ccc} 179 & + & 182 & + & 184 \\ \times 0 & & \times -1 & & \times 0 \\ \\ + & 178 & + & 182 & + & 191 \\ \times -1 & & \times 5 & & \times -1 \\ \\ + & 177 & + & 183 & + & 186 \\ \times 0 & & \times -1 & & \times 0 \end{array} \right) = 176$$

kernel:
sharpen



output image

Пример фильтра улучшения чёткости

$$\text{kernel} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



-2	-1	0
-1	1	1
0	1	2

Ядро тиснения создает иллюзию глубины, подчеркивая различия пикселей в заданном направлении. В данном случае в направлении вдоль линии от левого верхнего угла к правому нижнему.



Фильтры. Сглаживание изображений

Сглаживание или размытие – это одна из самых простых и часто используемых операций обработки изображений. Как правило, размытие применяется, чтобы уменьшить шумы или артефакты.

В библиотеке OpenCV реализовано несколько функций размытия изображения, к примеру:

- `blur()`,
- `GaussianBlur()`
- `medianBlur()`

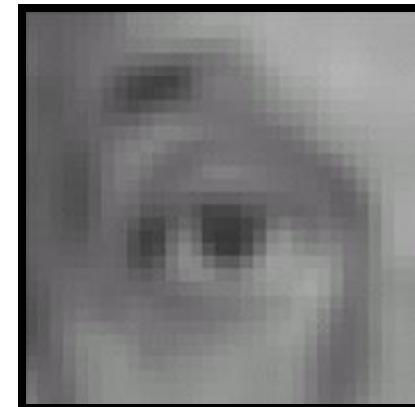
```
void blur(const Mat& src, Mat& dst, Size ksize,  
          Point anchor=Point(-1, -1),  
          int borderType=BORDER_DEFAULT)  
  
void GaussianBlur(const Mat& src, Mat& dst, Size ksize,  
                     double sigmaX, double sigmaY=0,  
                     int borderType=BORDER_DEFAULT)  
  
void medianBlur(const Mat& src, Mat& dst, int ksize)
```

Простейшие фильтры. Blur



Original

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

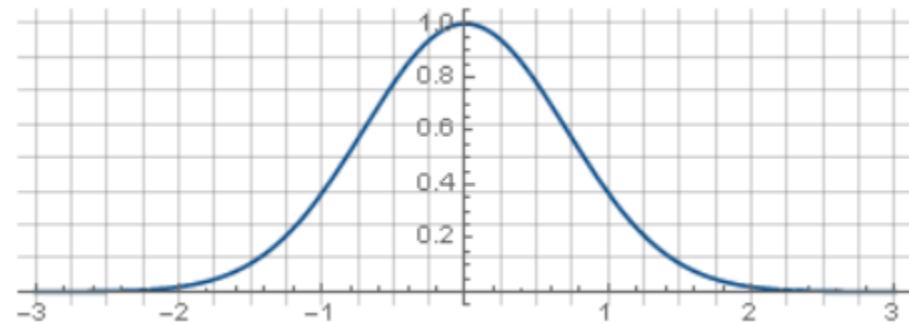


Blur (with a
box filter)

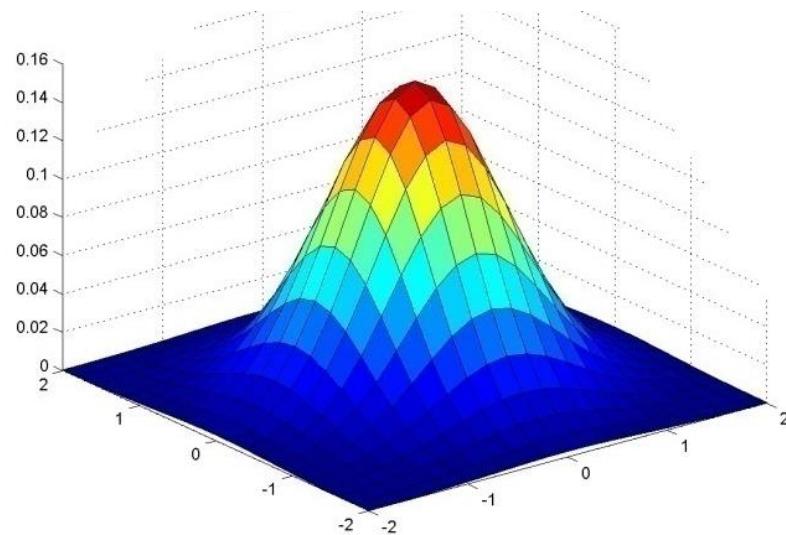
$$K = \frac{1}{\text{ksize.width} * \text{ksize.height}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 \\ \dots \\ 1 & 1 & 1 & \dots & 1 & 1 \end{bmatrix}$$

Фильтр Гаусса (Gaussian Blur)

Фильтр называется Gaussian, потому что он строится из функции, известной как гауссиана,



$$e^{-x^2}$$

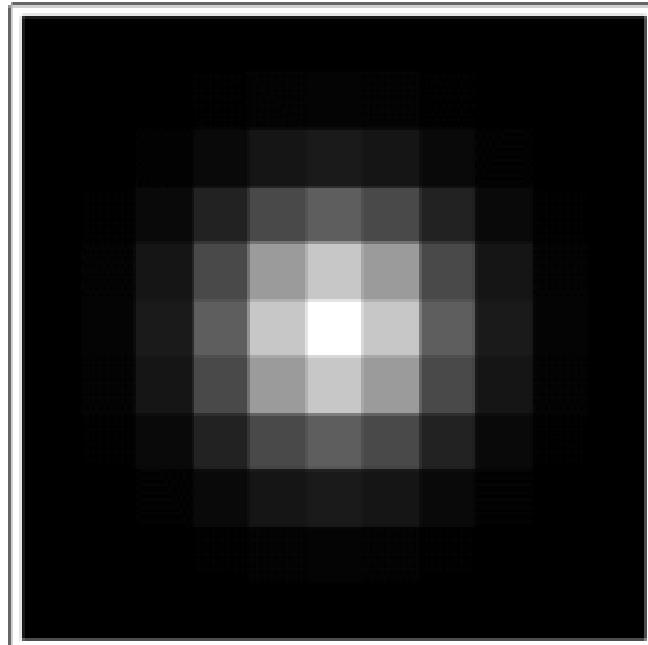


$$e^{-(x^2+y^2)}$$

Фильтр Гаусса (Gaussian Blur)

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

Для нормализации по координатам введена **сигма** (среднеквадратическое отклонение), которая определяет степень сжатия гауссианы



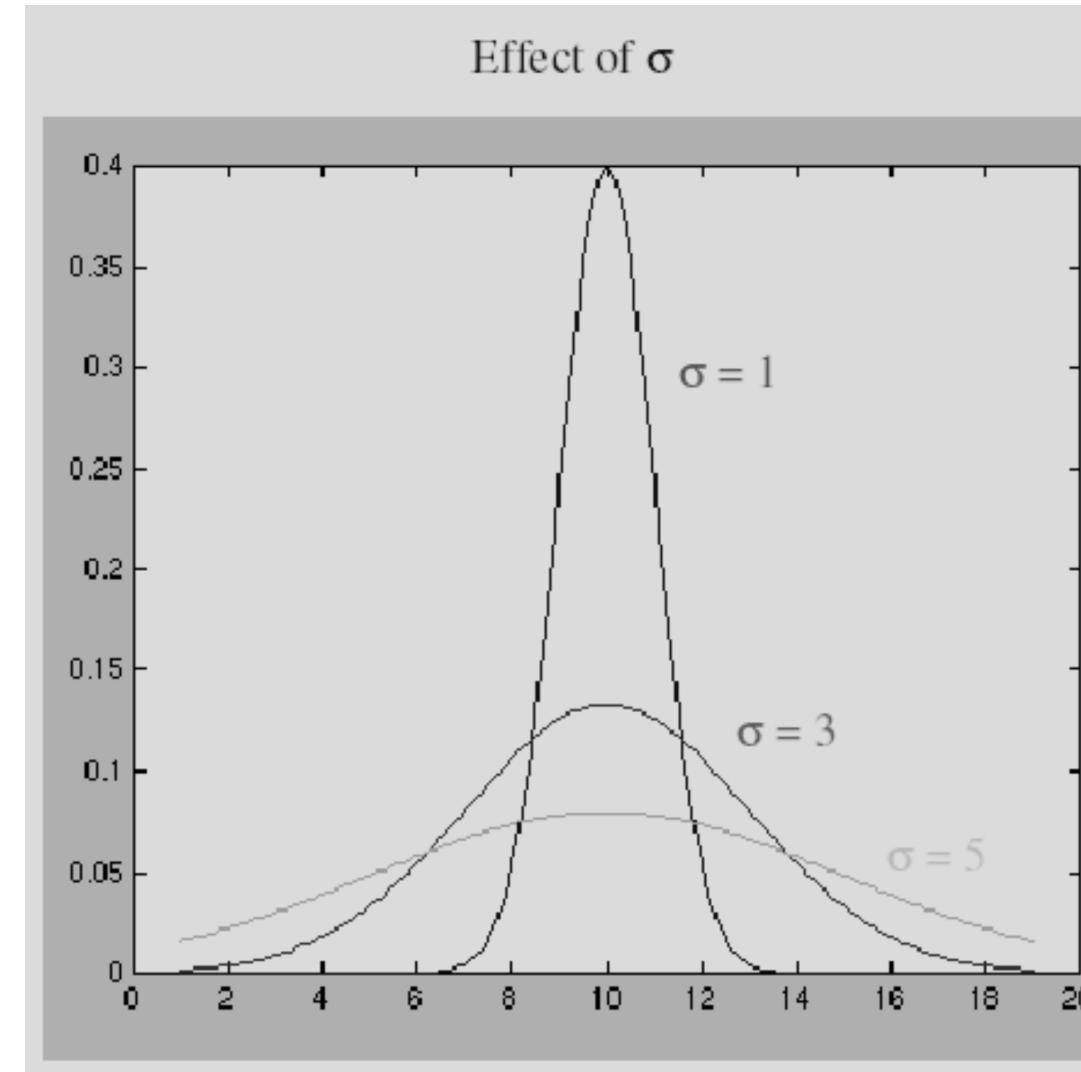
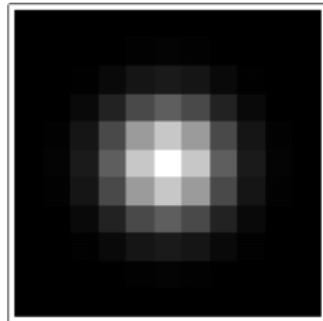
Ядро фильтра Гаусса

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

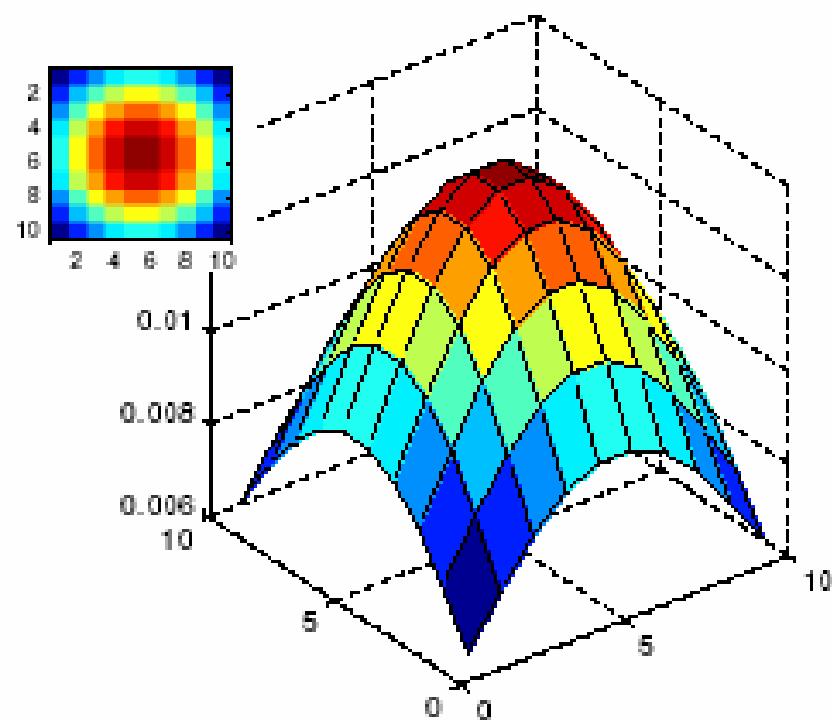
$5 \times 5, \sigma = 1$

```
void GaussianBlur(const Mat& src, Mat& dst, Size ksize,  
                  double sigmaX, double sigmaY=0,  
                  int borderType=BORDER_DEFAULT)
```

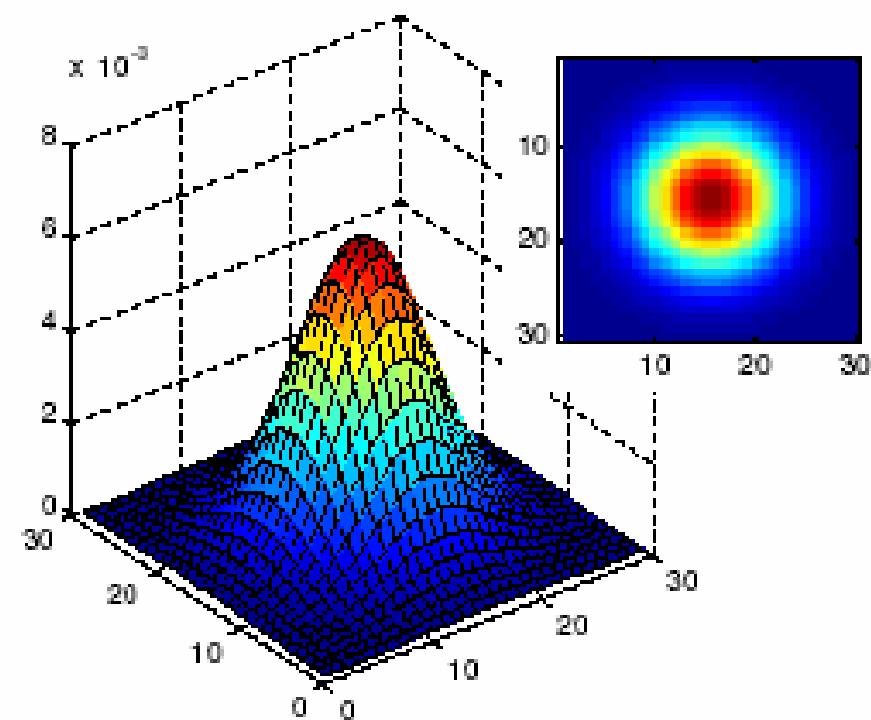
зависимость размера
сглаживающей области от
сигма



Выбор размера ядра



$\sigma = 5$ kernel 10x10



$\sigma = 5$ kernel 30x10



Original



Salt and pepper noise



Impulse noise

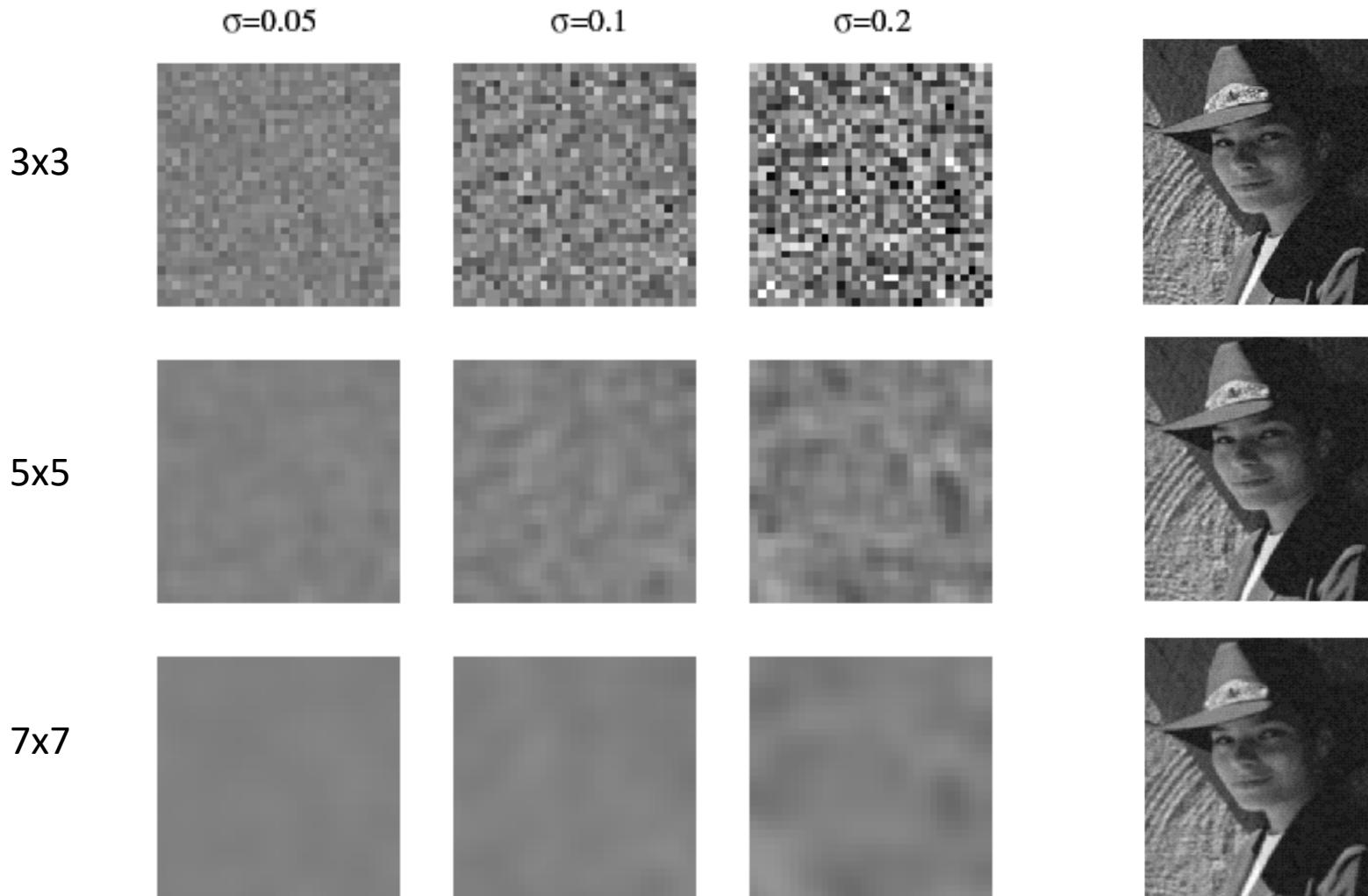


Gaussian noise

Виды шума

- **Соль и перец:** случайные черные и белые пиксели
- **Импульсный:** случайные белые пиксели
- **Гауссов:** колебания яркости, распределенные по нормальному закону

Подавление гауссова шума



Сглаживание фильтрами большого радиуса подавляет шум, но размывает изображение

Подавление шума «соль и перец»

3x3



5x5



7x7

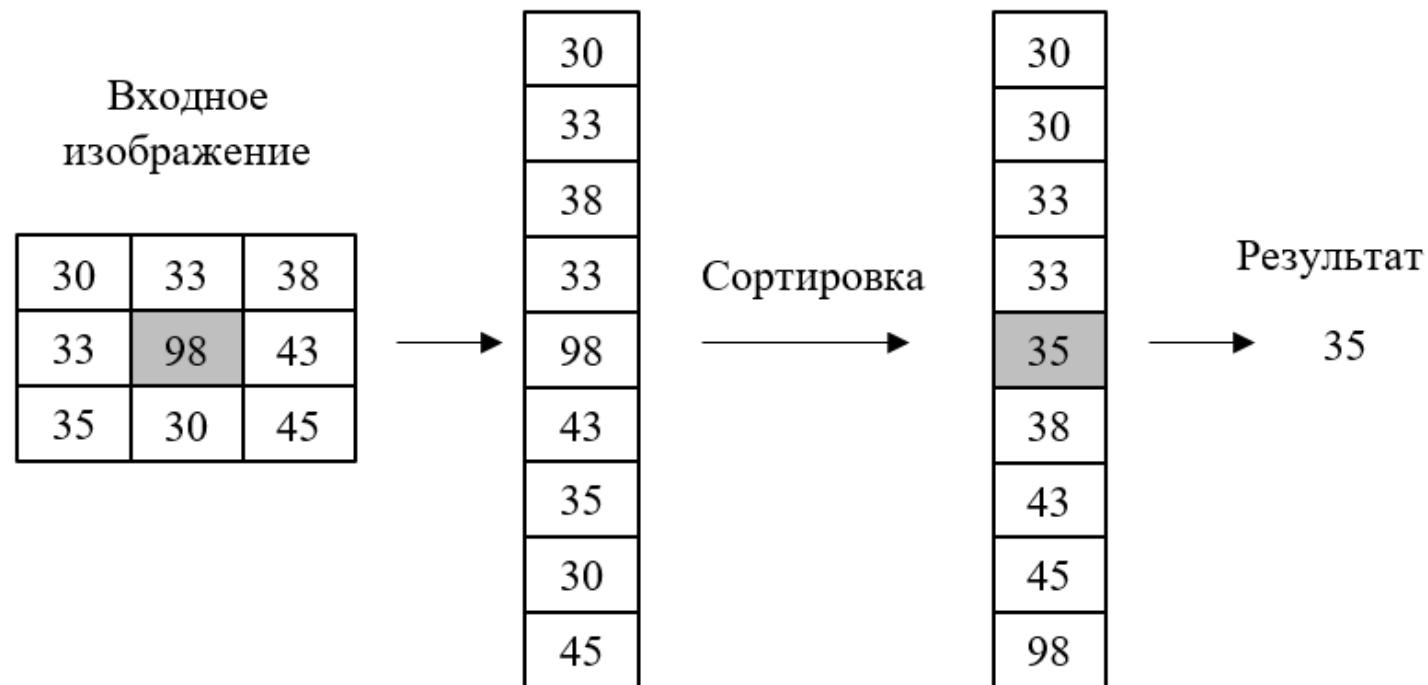


Чем плох результат?

Медианный фильтр

Фильтр работает с матрицами различного размера, но в отличие от матрицы свёртки, размер матрицы влияет только на количество рассматриваемых пикселей.

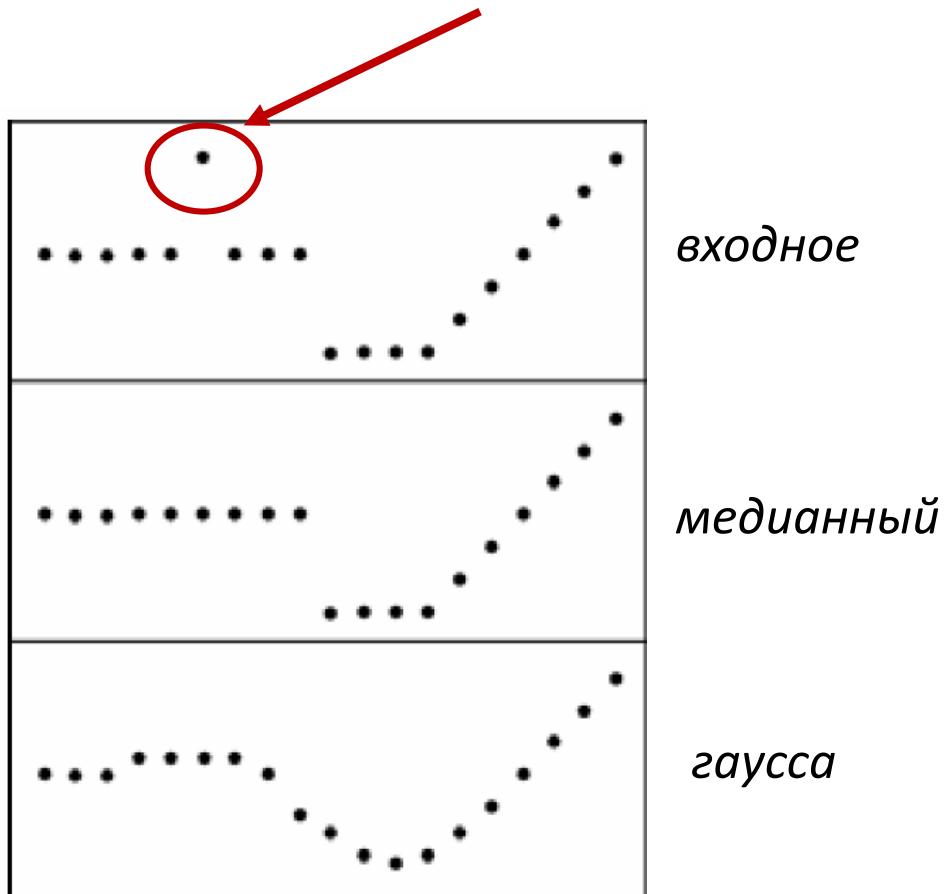
Пиксели, которые «попадают» в матрицу вокруг текущего пикселя, сортируются, и выбирается серединное значение из отсортированного массива. По сути, определяется медиана в отсортированном наборе данных. Это значение и является выходным для текущего пикселя.



Медианный фильтр

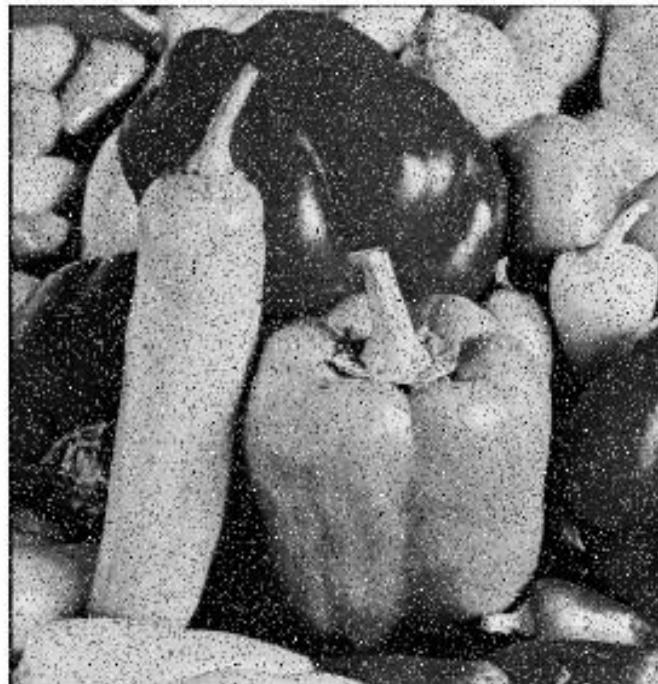
В чем преимущество медианного фильтра перед фильтром гаусса?

Устойчивость к выбросам (outliers)

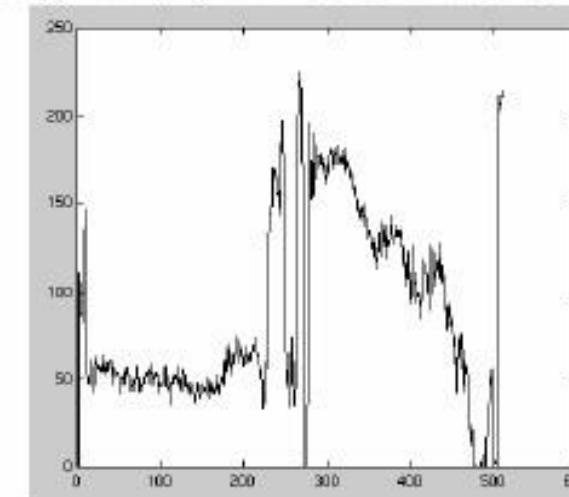
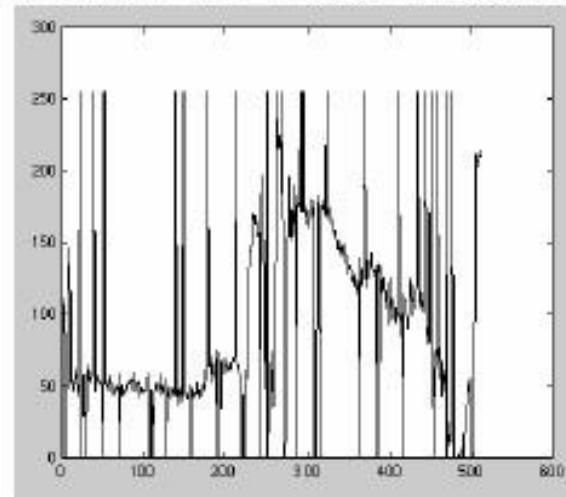


Преимущество медианной фильтрации перед фильтрами размытия (*blur()*, *box()*, *GaussianBlur()*) заключается в том, что «битый» пиксель на темном фоне будет заменен на темный, а не «размазан» по окрестности.

Salt-and-pepper noise



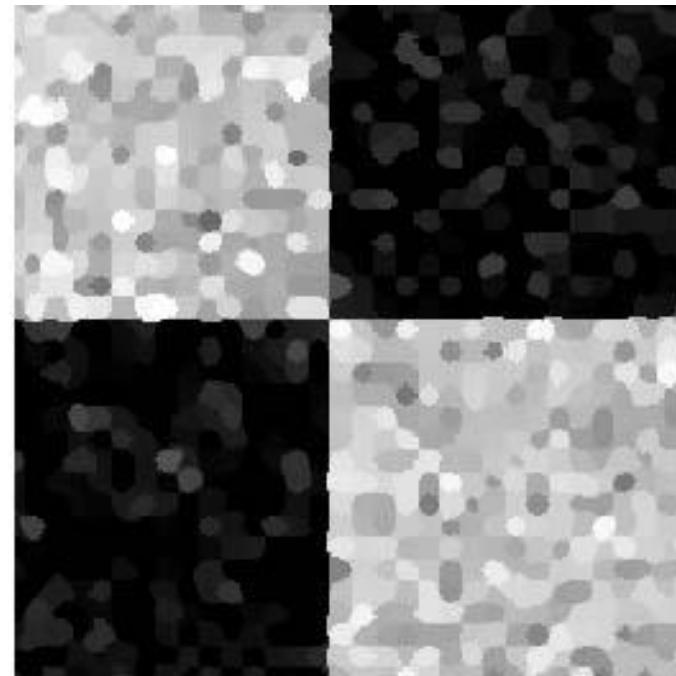
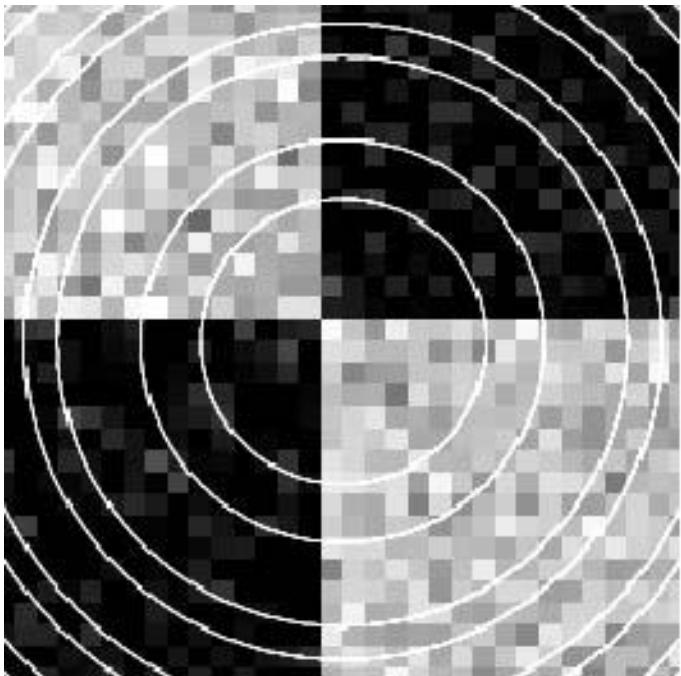
Median filtered



Source: M. Hebert

Медианный фильтр

Результат применения медианного фильтра к изображению с артефактами в виде тонких светлых линий.



Сравнение фильтров

3x3



5x5



7x7



Гауссов

Медианный



Повышение резкости

Что теряется при сглаживании?



-



=



Добавим дополнительно высокие частоты:



$+ \alpha$



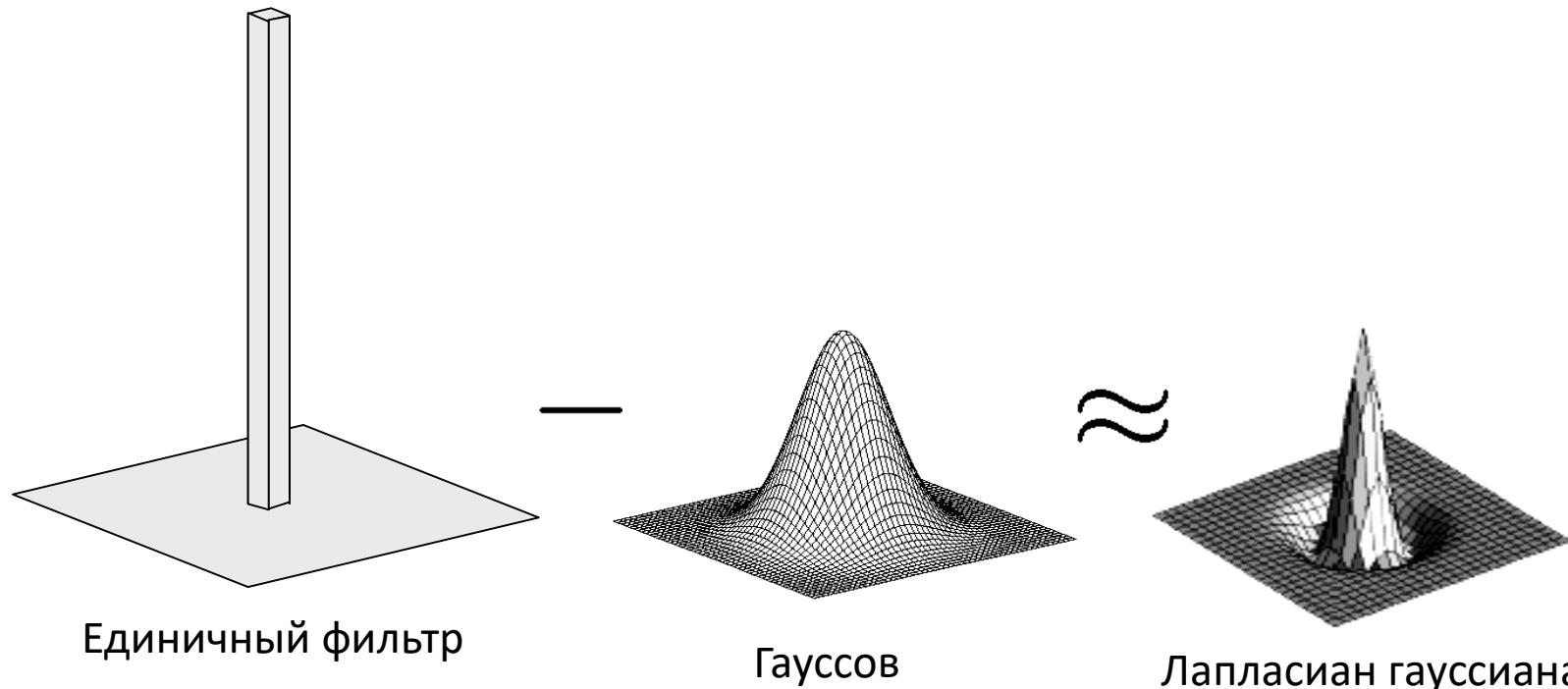
=



Повышение резкости (фильтр Unsharp)

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

↑ ↑ ↑
изображение сглаженное Единичный
 изображение Фильтр



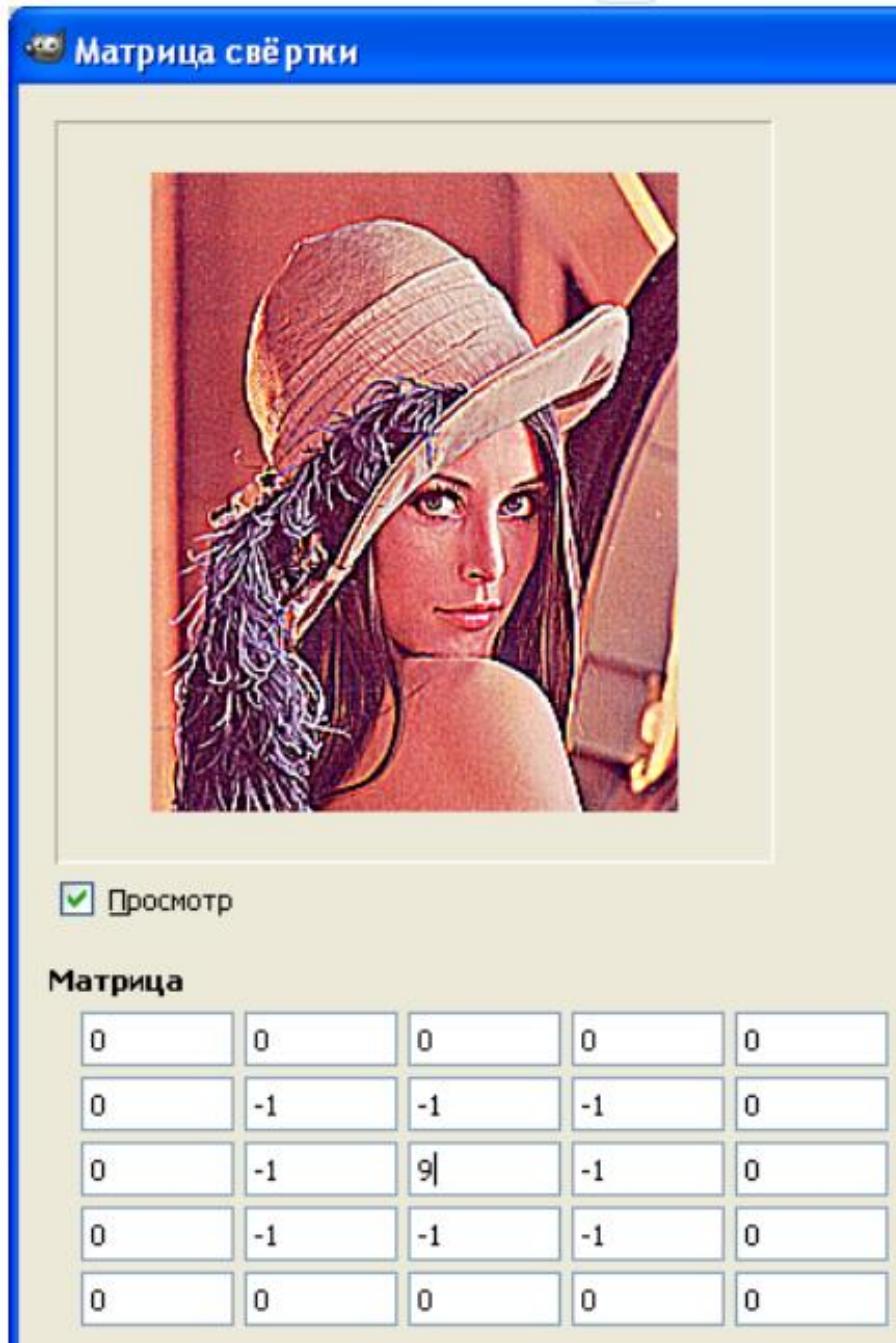
Пример фильтра улучшения чёткости

Ядро свертки

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



В некоторых графических редакторах (в частности в программе *GIMP*) есть фильтр «Матрица свёртки», который упрощает поиск необходимого матричного преобразования.



Тиснение



$$\begin{vmatrix} 2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{vmatrix} \text{ либо}$$

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

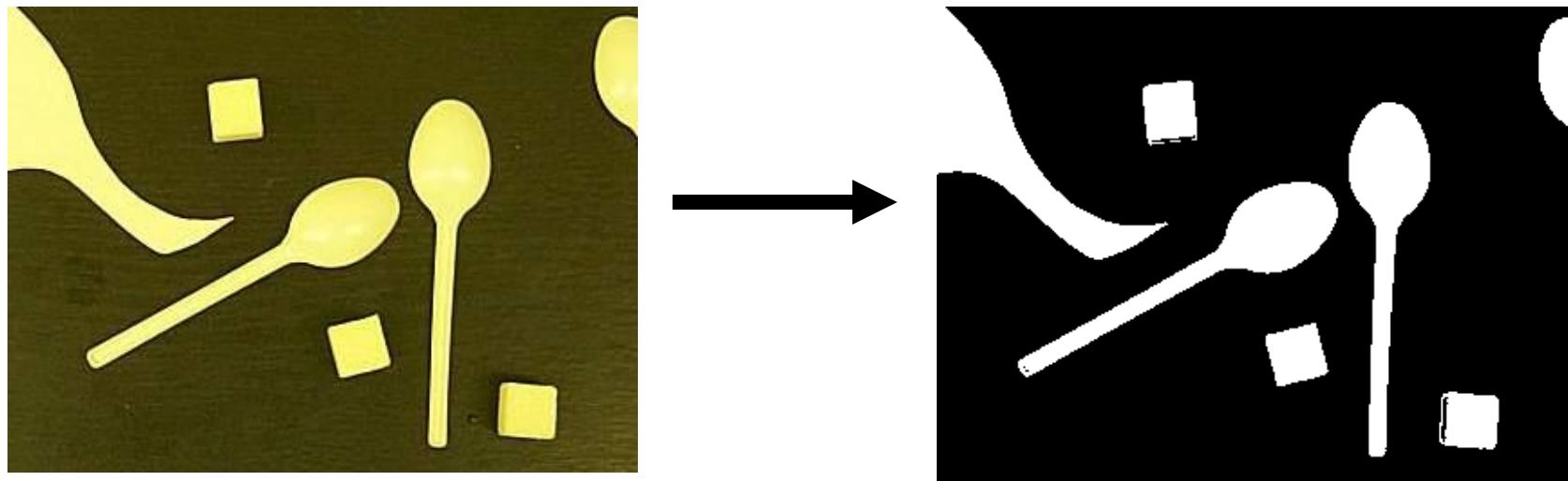
Выделение границ



$$\begin{vmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{vmatrix}$$

Бинаризация изображений

Бинаризация изображений

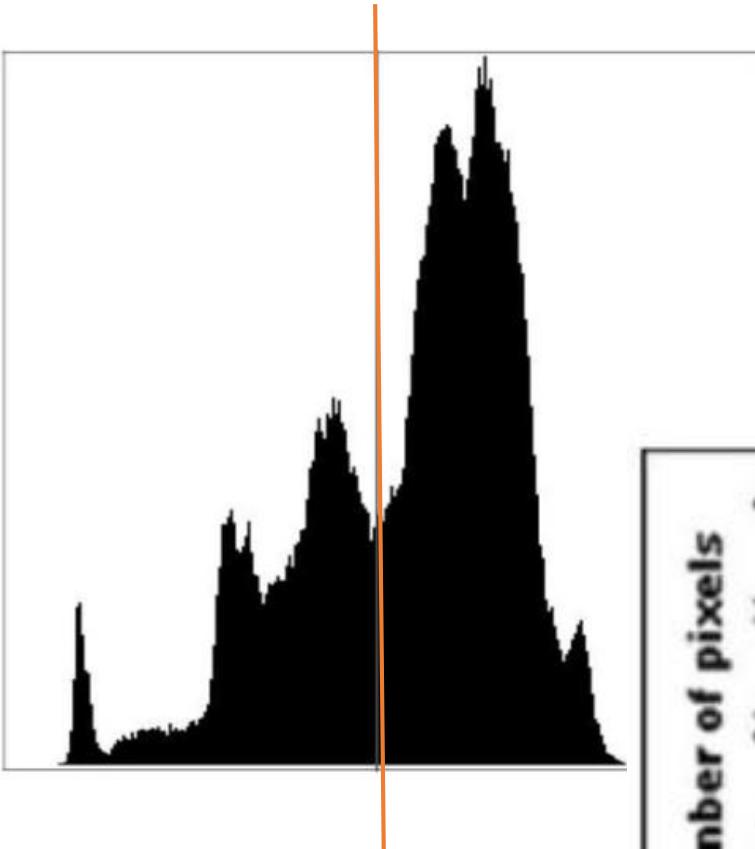


- Бинаризация – построение бинарного изображения по полутоновому / цветному
 - *пороговая фильтрация, которая сводится к выделению областей, яркость которых выше/ниже некоторого порога*
- Пиксель бинарного изображения может значения 0 и 1

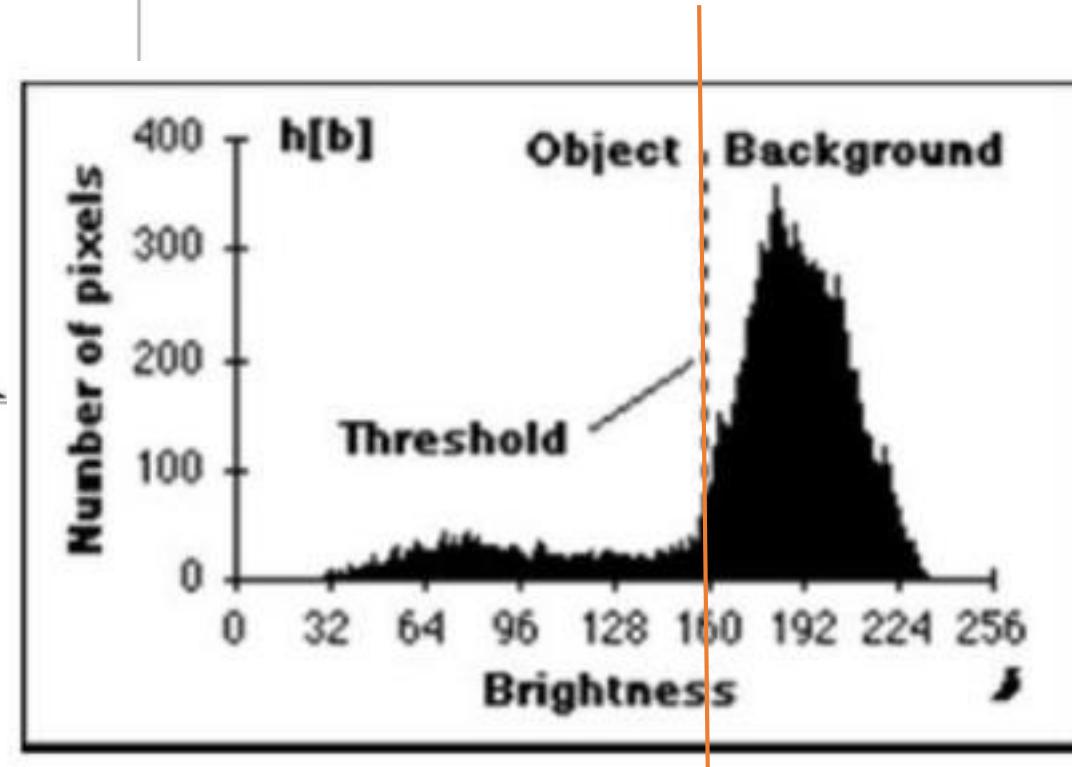
Смысл

- Разделить изображение на фон и контрастные объекты

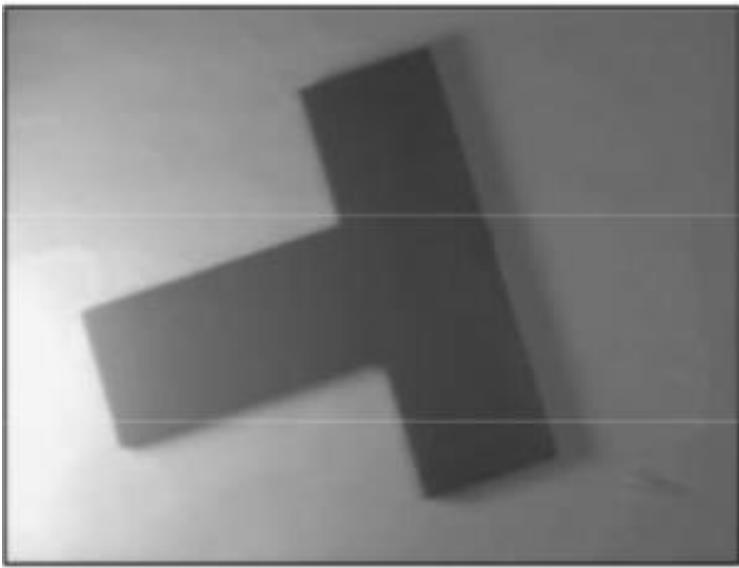
Бинаризация изображений



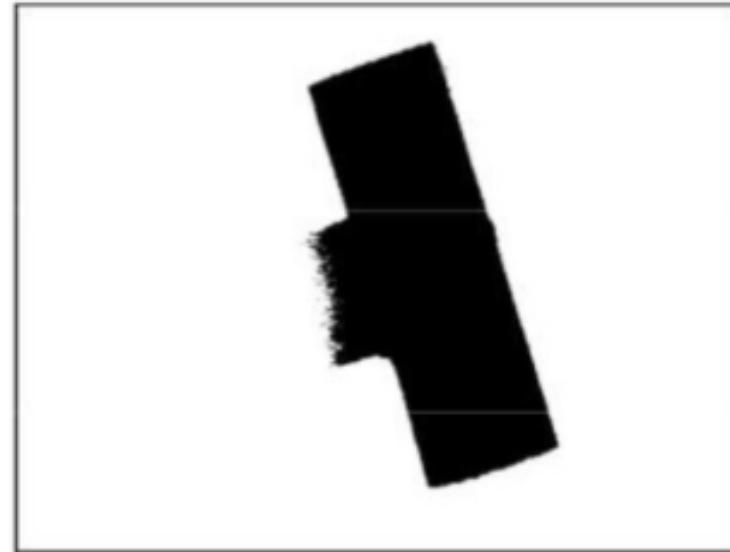
Пороговая
бинаризация



Пороговая бинаризация



В данном случае привела к неудовлетворительному результату



Адаптивная бинаризация

Необходима в случае неравномерной яркости фона/объекта.

- Для каждого пикселя изображения $I(x, y)$:
 1. В окрестности пикселя радиуса r высчитывается индивидуальный для данного пикселя порог T ;
 2. Если $I(x, y) > T + C$, результат 1, иначе 0;

Варианты выбора T :

- $T = mean$
- $T = median$
- $T = (min + max) / 2$

Адаптивная бинаризация



Sonnet for Lena



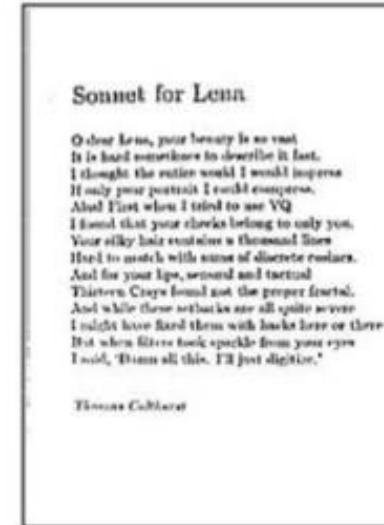
Sonnet for Lena

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 comprise.
Alas! First when I tried to see VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sense of discrete lines.
And for your lips, second and tactical
Thirteen. Cross found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with backs here or there.
But when Slices took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Wyatt

Исходное

$r=7, C=0$

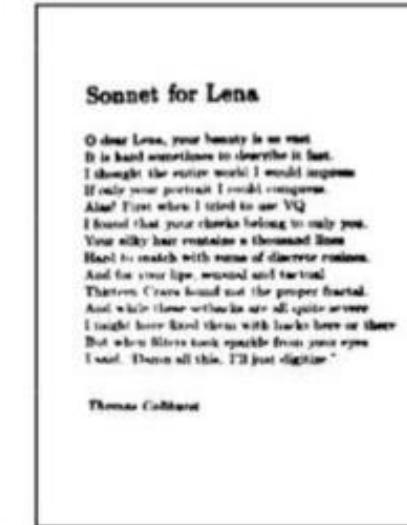


Sonnet for Lena

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 comprise.
Alas! First when I tried to see VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sense of discrete lines.
And for your lips, second and tactical
Thirteen. Cross found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with backs here or there.
But when Slices took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Wyatt

$r=7, C=7$



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 comprise.
Alas! First when I tried to see VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sense of discrete lines.
And for your lips, second and tactical
Thirteen. Cross found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with backs here or there.
But when Slices took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Wyatt

$r=75, C=10$

r – радиус окрестности пикселя,
 C – производная константа

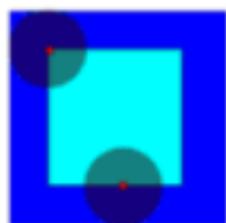
Если $I(x, y) > T + C$, результат 1, иначе 0

Операции математической морфологии

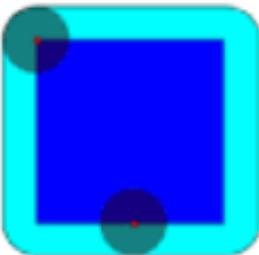
Широко известный способ - устранение шума с помощью операций математической морфологии:

- Сужение (**erosion**) увеличиваются тёмные области
- Расширение (**dilation**) увеличиваются светлые области
- Закрытие (**closing**)
- Раскрытие (**opening**)

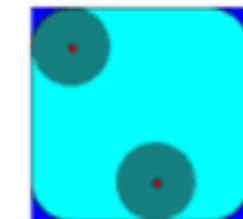
Erosion



Dilation



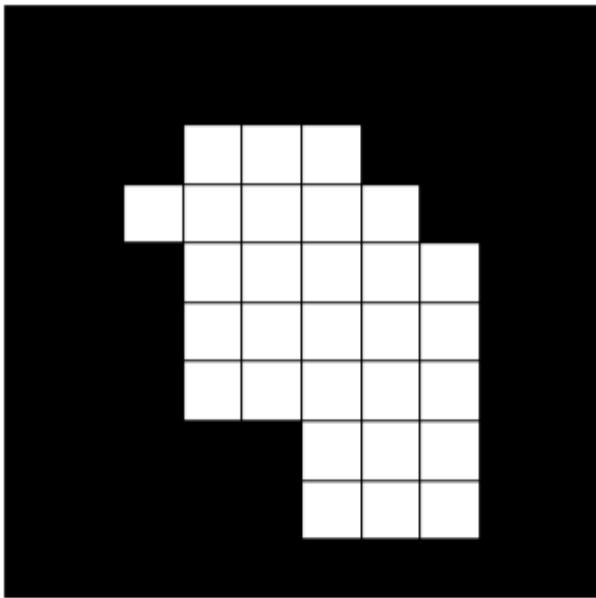
Opening



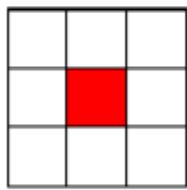
Closing



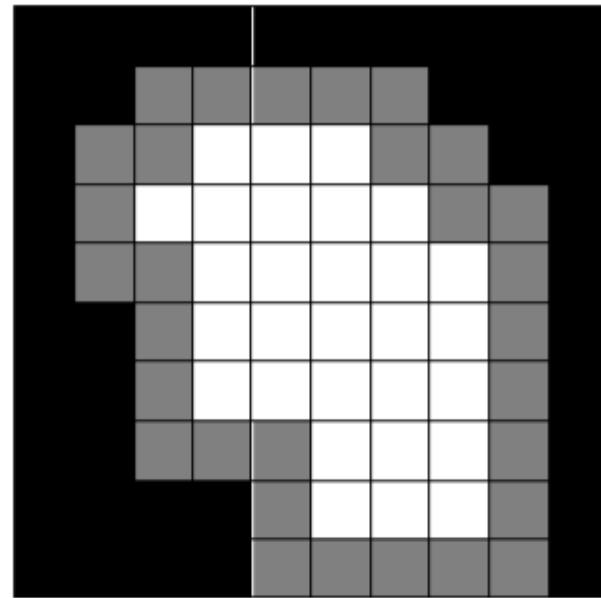
*Математическая морфология
– это теория и техника
анализа и обработки
геометрических структур*



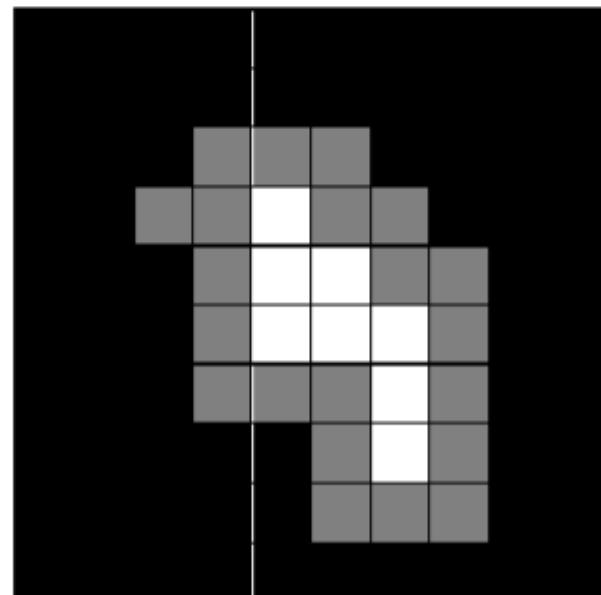
а) исходное изображение



б) шаблон (центр – ведущий элемент)



с) результат дилатации

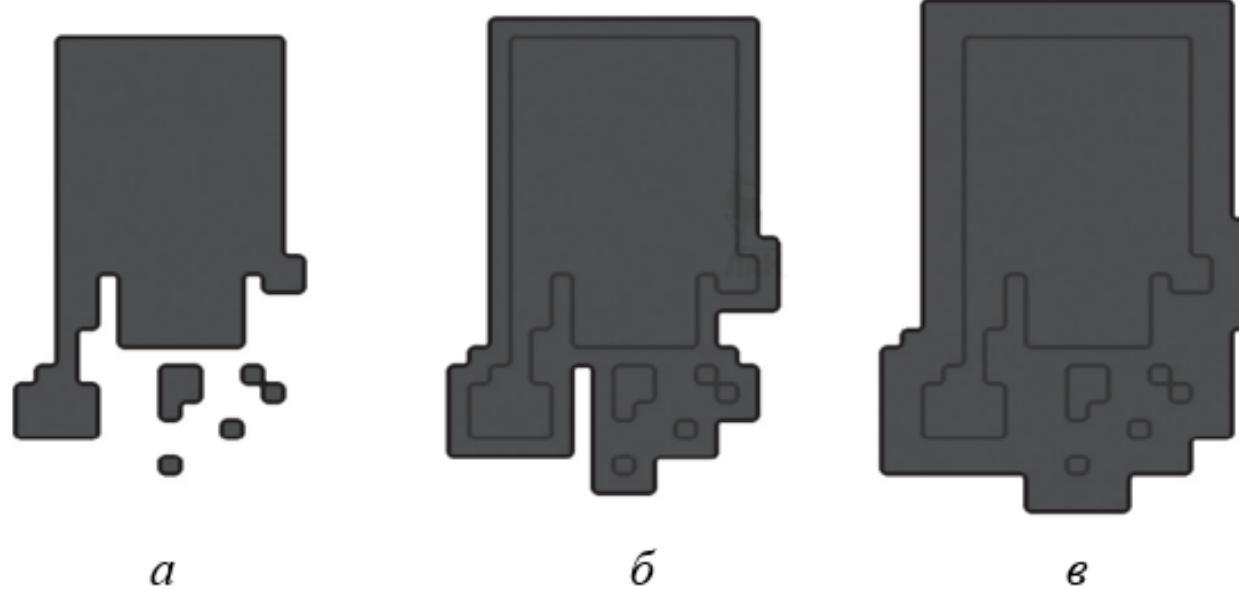


д) результат эрозии

Дилатация

Растягивание (расширение), по идеи, должно устранять шум и **способствовать объединению областей изображения, которые были разделены шумом, тенями.**

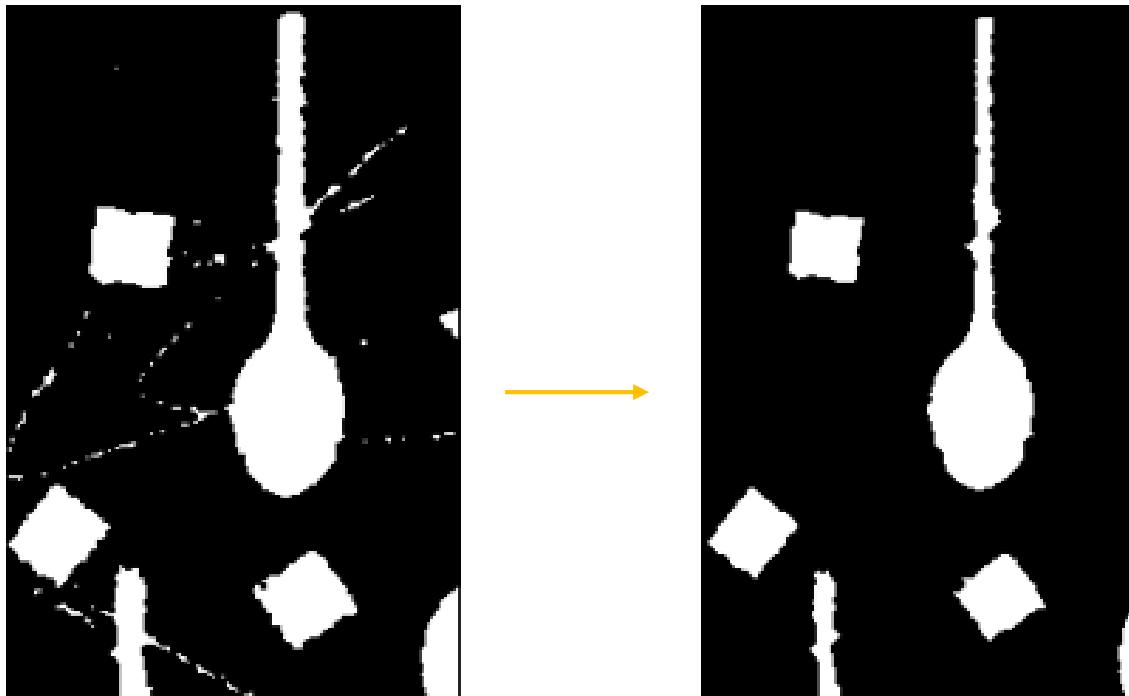
Применение небольшого растягивания должно сплавить эти области в одну.



Результат применения дилатации: а – оригинал; б – дилатация; в – после второго применения

Эрозия

- Эрозия (размытие/сужение) изображения обычно используется для избавления от случайных вкраплений на изображении. **Идея состоит в том, что вкрапления при размытии устраняются, тогда как крупные и соответственно более визуально-значимые регионы остаются.**

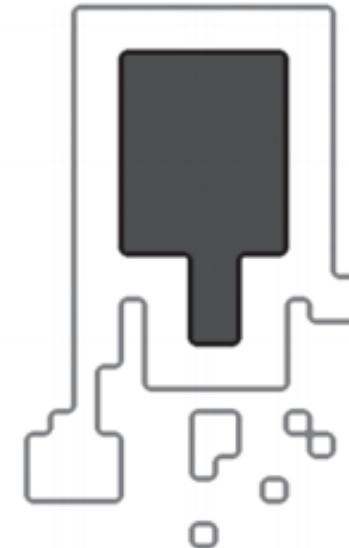




а



б



в

Результат применения эрозии: а – оригинал; б – эрозия; в – после второго
применения

- Морфологическое раскрытие (opening)

$$open(A,B) = (A(-)B)(+)B$$



Сильный шум



Эрозия



Раскрытие

Применение закрытия

- Применим операцию закрытия к изображению с дефектами объектов:



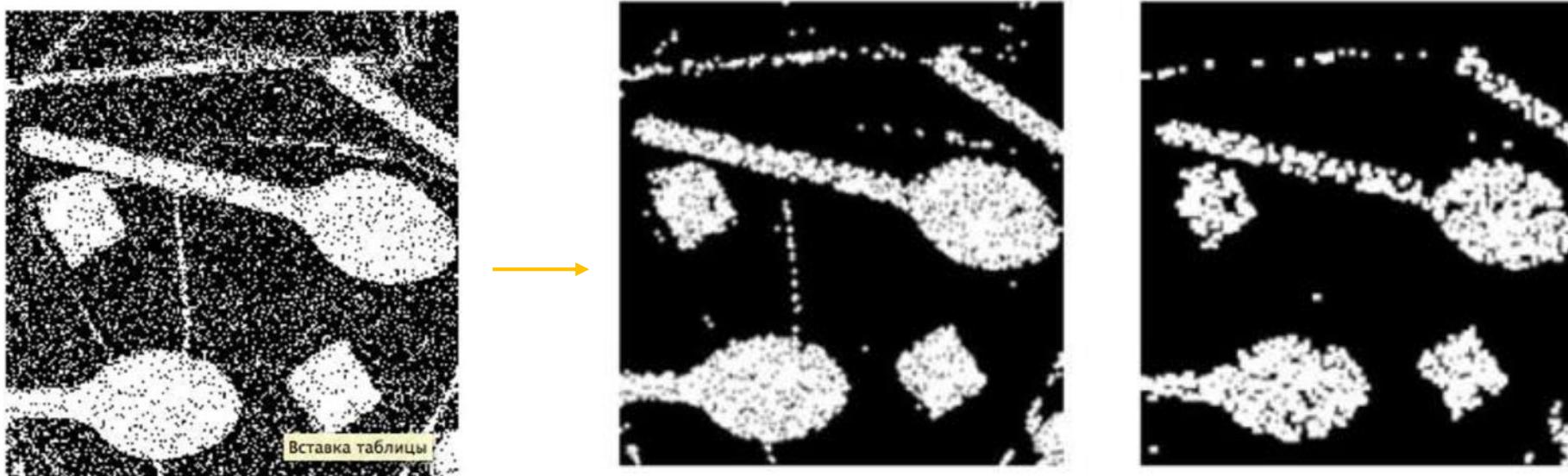
- Морфологическое закрытие (closing)

$$\text{close}(A, B) = (A (+) B) (-) B$$

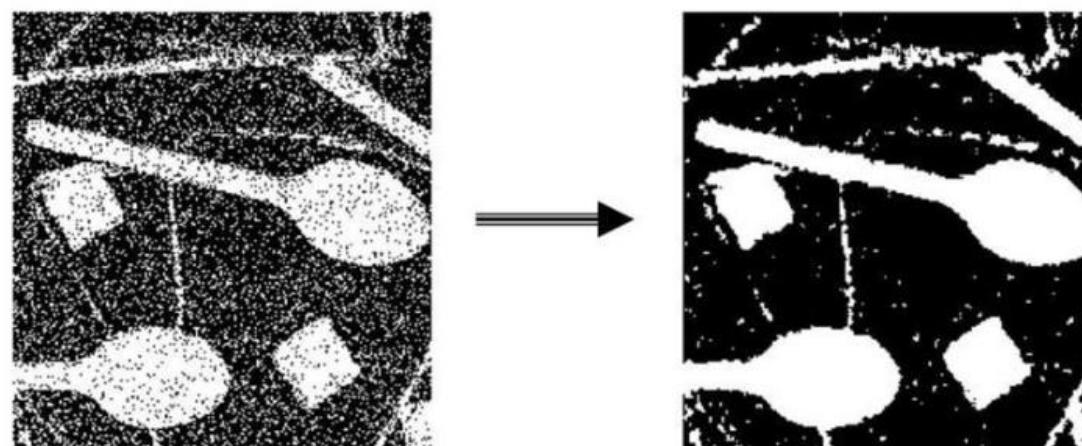
Не лучший пример для морфологии



Применение морфологических операций приведет к таким результатам



В начале желательно применить медианный фильтр



В *OpenCV* эти преобразования реализуются функциями *erode()* и *dilate()*:

```
void erode(  
    const Mat& src,           // входное изображение  
    const Mat& dst,           // выходное изображение  
    const Mat& element,       // ядро, объект типа Mat()  
    Point anchor = Point(-1,-1), // позиция якоря  
    int iterations = 1,         // сколько раз применять  
    int borderType = BORDER_CONSTANT, // экстраполяция границы  
    const Scalar& borderColor = morphologyDefaultBorderValue()  
);
```

Запишем фрагмент кода, применяющий операции эрозии и дилатации к изображению *img*:

```
element = Mat();           // ядро по умолчанию  
erode(img, erodeImg, element); // вычисление эрозии  
dilate(img, dilateImg, element); // вычисление дилатации
```

Операции размыкания и замыкания можно реализовать с помощью многоцелевой функции *morphologyEx()*, параметры которой аналогичны параметрам функций *erode()* и *dilate()*

```
void morphologyEx(  
    const Mat& src,           // входное изображение  
    const Mat& dst,           // выходное изображение  
    int op,                   // оператор  
    const Mat& element,       // ядро, объект типа Mat()  
    Point anchor = Point(-1,-1), // позиция якоря  
    int iterations = 1,         // сколько раз применять  
    int borderType = BORDER_CONSTANT, // экстраполяция границы  
    const Scalar& borderColor = morphologyDefaultBorderValue()  
);
```

Оператор *op* определяет выбранную операцию: *MOP_OPEN* – размыкание; *MOP_CLOSE* – замыкание; *MOP_GRADIENT* – морфологический градиент; *MOP_TOPHAT* – «Верх шляпы»; *MOP_BLACKHAT* – «Черная шляпа».

Пример использования процедуры дилатации для устранения разрывов



До дилатации

0	1	0
1	1	1
0	1	0

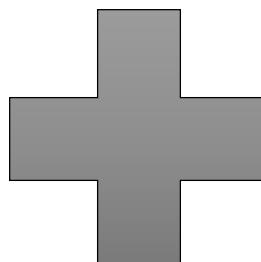
В



После дилатации

0	1	0
1	1	1
0	1	0

В



- Эрозия: $\text{dst}(x,y) = \min_{((x',y') \text{ in } [-3..3])} \text{src}(x+x',y+y')$



- Наращивание: $\text{dst}(x,y) = \max_{((x',y') \text{ in } [-3..3])} \text{src}(x+x',y+y')$



- Размыкание: $\text{dst} = \text{open}(\text{src}) = \text{dilate}(\text{erode}(\text{src}))$



- Замыкание: $\text{dst} = \text{close}(\text{src}) = \text{erode}(\text{dilate}(\text{src}))$

