

ТЕХНИЧЕСКОЕ ЗРЕНИЕ

We perceive this:



picture from:

Gary Bradski, Adrian Kaehler, "Learning OpenCV:
Computer Vision with the OpenCV Library"

But the camera sees this:

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	186	78	88
87	103	115	154	143	142	149	153	173	181	37	37
102	112	106	131	122	130	152	147	128	94	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	99	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	49	69	75	56	41	51	73	55	78	63	44
50	59	37	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
97	61	58	65	75	78	76	73	59	75	69	58

Шишков Александр Валерьевич

Пн – 08:00-09:30, 16:20-21:10

Вт – 19:30-21:10

Сп – 11:20-12:50, 16:20-21:10

Чт – 19:30-21:10

Пт – 16:20-21:10

Сб – 08:00-11:10, 16:20-21:10

Введение

Что такое зрение?

- 1) зрение — восприятие организмом объектов внешнего мира посредством улавливания отражаемого или излучаемого объектами света (*Большой Энциклопедический словарь*)
- 2) зрение — (visio, visus) восприятие света органом зрения и зрительным анализатором, благодаря чему организм получает информацию об объектах (*Большой медицинский словарь*)
- 3) зрение — восприятие организмом внешнего мира, т. е. получение информации о нём, посредством улавливания специальными органами зрения отражаемого или излучаемого объектами света (*Большая советская энциклопедия*)
- 4) зрение — Одно из внешних чувств человека и животного, органом которого является глаз; (*Толковый словарь Ожегова*)
- 5) «To see means to know what is where by looking, David Marr, Vision, 1982

Что такое компьютерное зрение?

Компьютерное зрение — теория и технологии создания машин, которые способны извлекать полезную информацию из изображений.

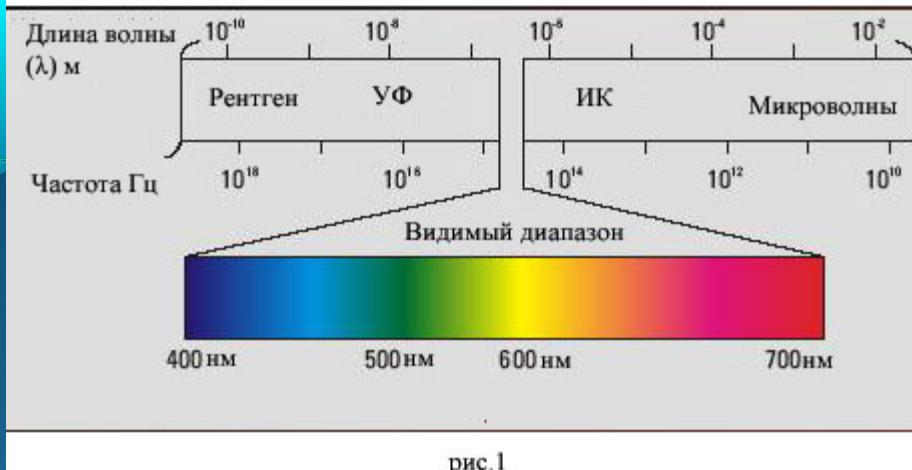


рис.1

Инфракрасное зрение



Термальное зрение

Рентгеновское зрение



Терагерцовое зрение



3D зрение

В чём сложность?

We perceive this:



picture from:

Gary Bradski, Adrian Kaehler, "Learning OpenCV: Computer Vision with the OpenCV Library"

But the camera sees this:

194	210	206	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	181	57	37
182	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
29	43	49	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
87	61	58	65	75	78	76	73	59	75	69	50



Видеонаблюдение.

С
л
о
ж
н
о
с
т
ь

Детектирование движения и запись видео. (видеорегистраторы)

Детектирование только определённых объектов. (метро)

Охрана определённых зон. Охрана предметов. (музеи)

Детектирование оставленных предметов/машин. (вокзалы)

Подсчёт движущихся объектов на сцене. (магазины)

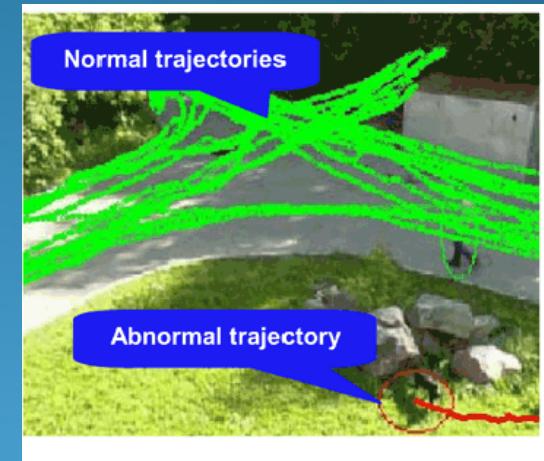
Сопровождение объектов. Определение их скорости.

Анализ траекторий движения. (рекомендации маркетингу).

Сопровождение объекта используя несколько камер.

Детектирование поскользнувшихся и упавших

Обнаружение воровства на кассе.
(POS monitor)



Распознавание номеров

Сложности:

- множество возможных форматов номеров
- малый размер символов
- поворот плоскости номера относительно камеры
- слабый контраст
- грязь, блики

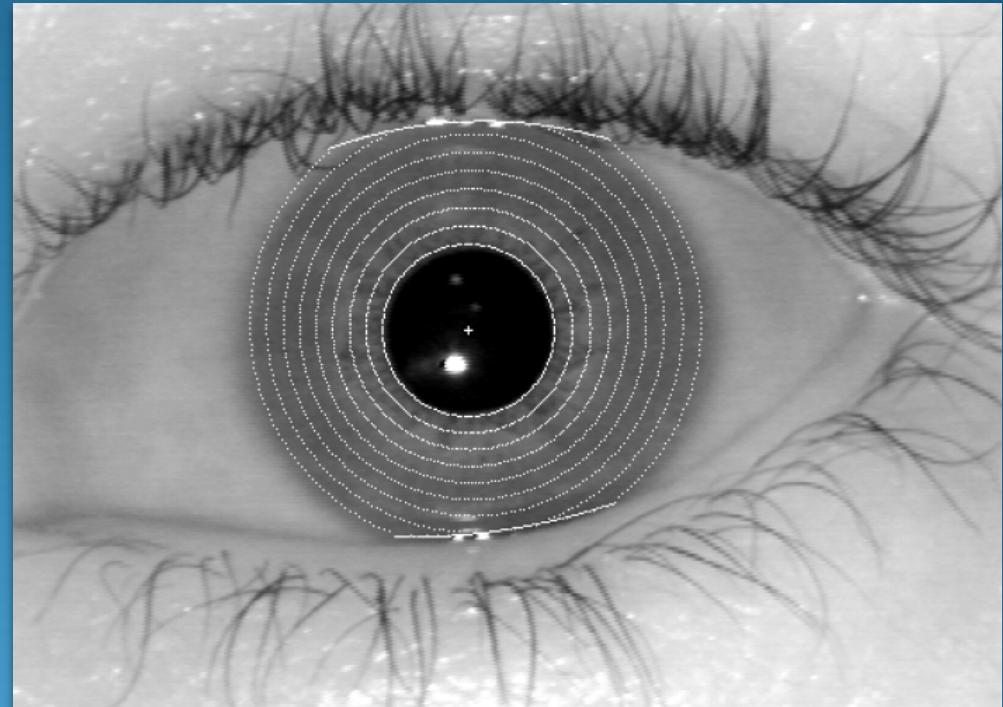


Биометрия

Распознавание
отпечатков пальцев

Распознавание лица
(2D, 3D)

Идентификация по
изображению
роговицы



Медицинские приложения

- Поиск аномалий на медицинских изображениях
- Помощь в проведении анализов
- Классификация рентгеновских снимков
- Детектирование эпи-припадков
- Автоматическая диагностика болезней глаз

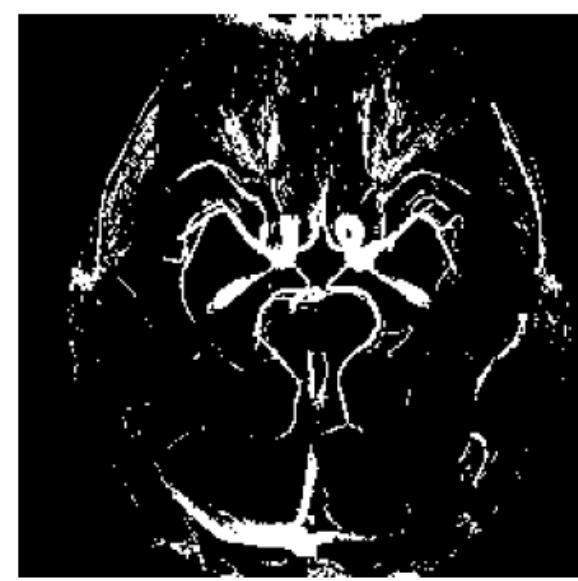


Figure 1.4: Magnetic resonance image (left) where brightness relates to blood flow and binary image (right) resulting from changing all pixels with value 208 or above to 255 and those below 208 to 0. Image courtesy of James Siebert, Michigan State Radiology.

Улучшение качества изображения



Удаление шумов.

Улучшение цветопередачи.

Улучшение контраста.

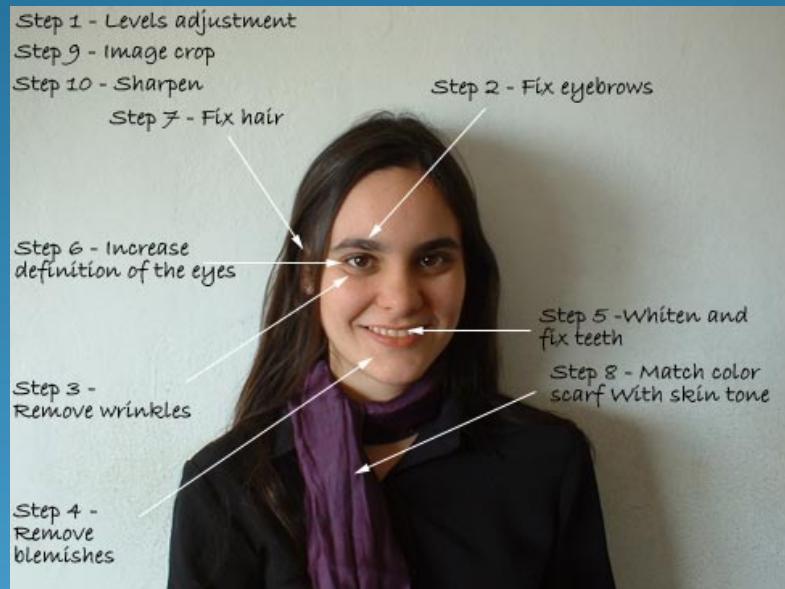
Просветление тёмных фрагментов.

Удаление нежелательных объектов.

Удаление “красных глаз”



Face Beautification



Поиск в архиве

Google

Scree....47 PM.png × описание изображения

Поиск Результатов: примерно 23 (0,50 сек.)

Все результаты

Картинки

Карты

Видео

Новости

Ещё

Поиск по картинке

Похожие

Другие размеры

За всё время

За час

За 24 часа

За неделю

За месяц

За год

За период...

Размер изображения:
405 × 275

Изображения других размеров не найдены.

Страницы с подходящими изображениями

 [ФотоПремия - Лучшие фотографии мира!](#)
fotopremiya.ru/index
Форум, Статистика, Последнее сообщение. О проекте. В этом форуме можно обсуждать любые темы касающиеся развития проекта
FotoPremiya.ru ...

 [Piligrim - Просмотр](#)
fotopremiya.ru/user/5-piligrim
Piligrim. Регистрация: 1
-. Найти темы Публикации
77 ...

 [Творческая мастерская](#)
mayart.ru/gallereya/
image audio video quote
Галереи · ФотоАльбомы · Image · Link ...
366 × 366

 [Видео уроки фотографии](#)

TinEye Services

Image recognition for your applications and enterprise.

Documentation Support Contact Us Login

MobileEngine

Mobile image recognition and identification

Build mobile application with fully integrated image recognition.
Connect users with your product information via a camera phone.

Details Plans & Pricing

MulticolorEngine

Let your users search through your image collection using color.

MobileEngine

Build mobile applications that use high-sensitivity image recognition.

MatchEngine

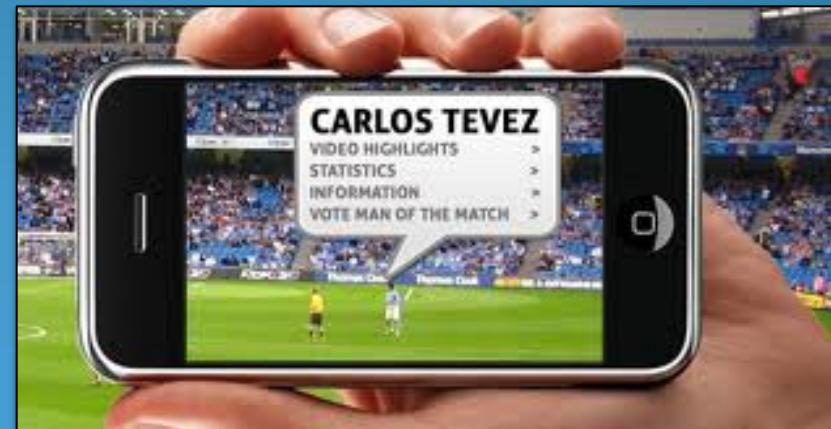
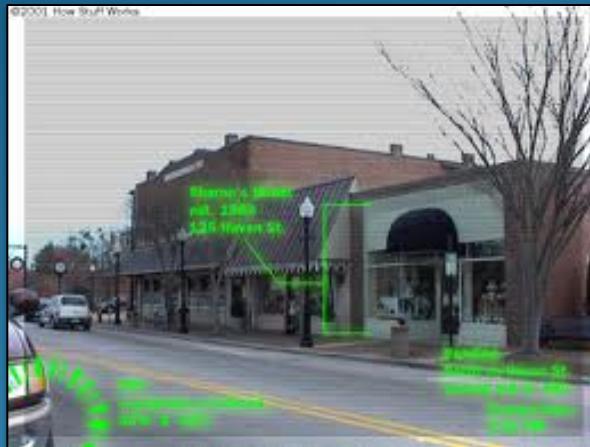
Finds duplicate, modified and even derivative images in your image collection.

TinEye API

Integrate TinEye with your own systems to automatically search the TinEye index.

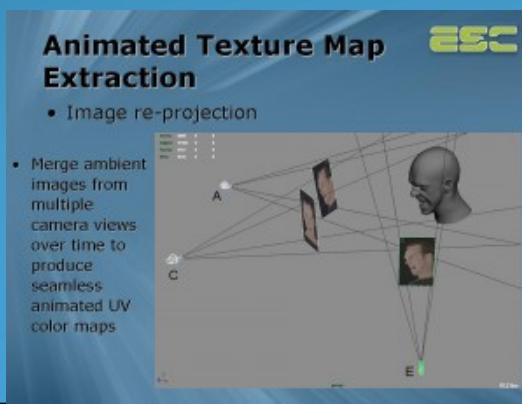
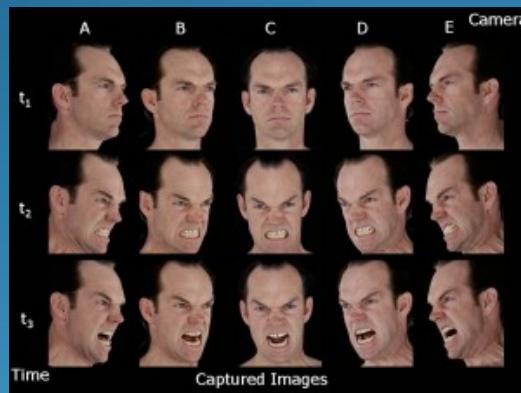


Дополненная реальность

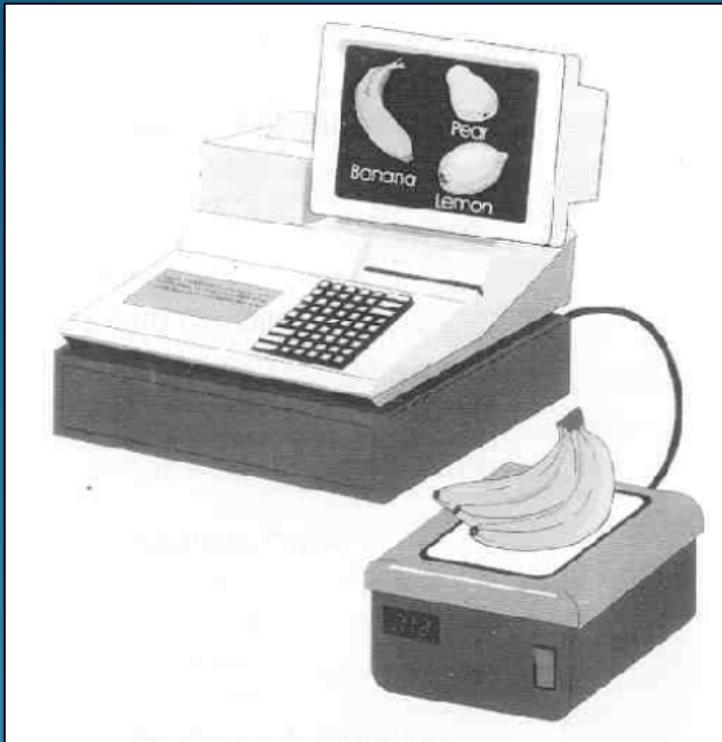


Киноиндустрия

Оценка движения камеры, оценка движения актеров
(optical motion capture systems), синтез
персонажей и сцен



Распознавание объектов по образцу



Machine vision.
Visual inspection.

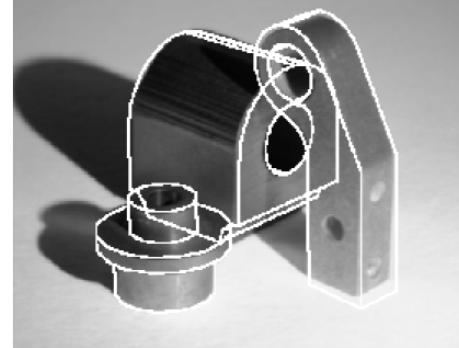


Figure 1.7: An inspection or assembly robot matches stored 3D models to a sensed 2D image (Courtesy of Mauro Costa).

Помощь водителю

- Определение смены полосы движения.
- Определение расстояния до препятствия.
- Нахождение светофора и предупреждение о красном цвете.
- Локализация и распознавание знаков.
- Детектирование пешеходов.
- Детектирование закрытых глаз водителя.



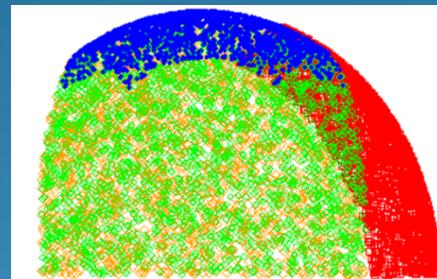
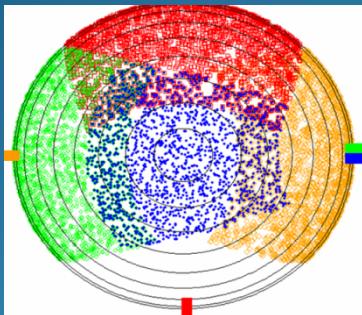
Помощь водителю

Очки расширенной реальности - инструмент ремонта машин



Взаимная калибрация камер и проекторов.

Цифровой планетарий



Центр компьютерного зрения «Аргус»

Argus Planetarium



Система
интерактивной
визуализации



Человеко-машинный интерфейс.



Содержание спецкурса

1. МЕТОДЫ ПОЛУЧЕНИЯ И ОБРАБОТКИ ИЗОБРАЖЕНИЙ

- 1.1. Введение в проблематику. Примеры систем технического зрения.
- 1.2. Формирование изображений. Камера Обскура. Перспективная проекция. Использование линз. Устройство современной цифровой камеры. Получение раstra. Основные искажения при формировании изображения.
- 1.3. Устройство человеческого глаза. Типовое устройство системы технического зрения.
- 1.4. Виды цифровых изображений. Основные форматы хранения изображений.
- 1.5. Инструменты обработки бинарных изображений. Нахождение связанных компонент. Морфологические операции. Инвариантные свойства связанных компонент. Векторизация бинарных изображений. Алгоритм. Применения. Бинаризация изображения. Анализ гистограммы. Метод Отсу.
- 1.6. Обработка и низкоуровневый анализ полутоновых и цветных изображений. Линейная фильтрация изображения. Свертка. Повышение чёткости. Подсчёт градиентов. Фильтры Собеля. Выделение границ объектов. Детектор границ Кани. Процедура трансформации расстояния. Нахождение угловых точек на изображении. Понятие гистограммы и улучшение контрастности. Выравнивание контрастности двух изображений. Эквализация гистограммы.
- 1.7. Мультиспектральные изображения. Виды цветовых пространств. Методы улучшения цветных изображений. Методы сегментации цветных изображений.

2. МЕТОДЫ ВИДЕОАНАЛИЗА

- 2.1. Постановки задач возникающие при видеонаблюдении.
- 2.2. Методы детектирования и оценки движения.
- 2.3. Обучение модели фона. Вычитание фона.
- 2.4. Постановка задачи слежения за объектом в видео потоке.
- 2.5. Численный метод поиска оптимального оптического потока. Применение градиентного метода поиска локального минимума функции.
- 2.6. Слежение за объектом с помощью алгоритма Meanshift.
- 2.7. Предсказание движения с помощью фильтра Калмана.
- 2.8. Детектирование подозрительных траекторий движения.

3. МЕТОДЫ ПОИСКА ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ

- 3.1. Постановка задачи поиска. Инвариантность поиска.
- 3.2. Обзор популярных методов локализации. Методы поиска сверху-вниз и снизу-вверх. Статистические техники.
- 3.3. Поиск шаблона с помощью решения двойственной задачи нахождения клики (максимального полного графа).
- 3.4. Нахождение ключевых точек изображения методом SIFT.
- 3.5. Использование ключевых точек изображения для предсказания положения объекта. Кластеризация в пространстве гипотез для нахождения наиболее вероятного положения объекта. Обобщённое преобразование Хафа.

Содержание курса

4. МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ И РАСПОЗНАВАНИЯ ОБРАЗОВ

- 4.1. Основные понятия распознавания образов. Общая модель классификации. Обучение с учителем и без. Базовые элементы статистики.
- 4.2. Подготовка данных. Методы фильтрации. Метод главных компонент. Метод канонических переменных.
- 4.3. Обзор классификаторов. К- ближайших соседей. Байесовский классификатор. Машина опорных векторов. Деревья решений. Нейронные сети.
- 4.4. Обучение без учителя. Методы кластеризации данных. К-средних. Агломеративная кластеризация. EM –алгоритм.

5. МЕТОДЫ ЛОКАЛИЗАЦИИ И РАСПОЗНАВАНИЯ ЛИЦ

- 5.1. Методы локализации лица. Детектирование лица с помощью каскадного классификатора на основе признаков Хара. Метод Adaboost. Признаки Хаара. Интегральные изображения. Подсчёт признаков Хаара с помощью интегральных изображений.
- 5.2. Методы поиска элементов лица (глаза, нос, рот).
- 5.3. Методы распознавания лиц. Активные модели. Геометрическое сравнение. Поэлементное сравнение. Метод главных компонент. Использование оптического потока.
- 5.4. Организация поиска в базе.

6. ЧИСЛЕННОЕ ОПИСАНИЕ, АНАЛИЗ И СРАВНЕНИЕ ИЗОБРАЖЕНИЙ

- 6.1. Постановка задачи поиска изображений. Практическая значимостью
- 6.2. Цветовые характеристики изображения.
- 6.3. Текстурные характеристики изображения.
- 6.4. Градиентные характеристики изображения.
- 6.5. Расстояние Хаусдорфа.
- 6.6. Различные численные методы сравнения изображений. Гистограммы. Корелограммы. Методы сравнения из стандарта MPEG-7.
- 6.7. Оптимальное хранение цифровой библиотеки. KD-деревья.

Содержание курса

7. МОДЕЛИРОВАНИЕ ВИЗУАЛЬНО НАБЛЮДАЕМЫХ ПРОЦЕССОВ. ЧИСЛЕННЫЕ МЕТОДЫ ОЦЕНКИ МОДЕЛИ

- 7.1. Примеры математического моделирования в задачах технического зрения.
- 7.2. Метод наименьших квадратов. Преобразование Хафа.
- 7.3. Задача оценки модели движущегося человека. Примеры моделей для оценки позы человека и локализации элементов лица.
- 7.4. Стохастические методы оптимизации модели. Метод фильтрации частиц.

8. КАЛИБРАЦИЯ КАМЕР И СТЕРЕОЗРЕНИЕ

- 8.1. Типы калибрации камер. Практическая значимость калибрации.
- 8.2. Модели камеры. Внутренние и внешние параметры камеры.
- 8.3. Обзор методов калибрации.
- 8.4. Стерео зрение. Эпиполярная геометрия. Выравнивание изображений стереопары. Использование структурного света.
- 8.5. Восстановление структуры по движению.
- 8.6. Методы нахождения стереосоответствия. Метод скользящего окна. Использование динамического программирования. Графовый метод.

9. ГЛУБОКОЕ ОБУЧЕНИЕ

- 9.1 Нейронные сети
- 9.2 Обратное распространение ошибки
- 9.3 Сверточные нейронные сети

10. ПРИМЕНЕНИЕ ТЕХНИЧЕСКОГО ЗРЕНИЯ В РОБОТОТЕХНИКЕ

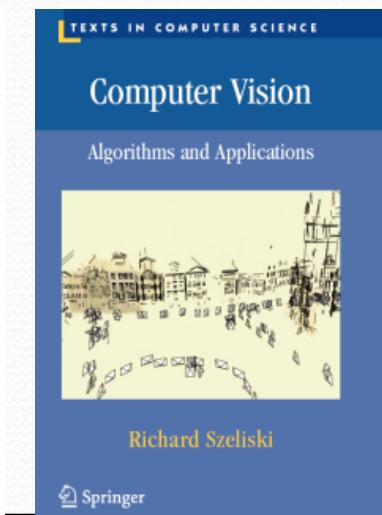
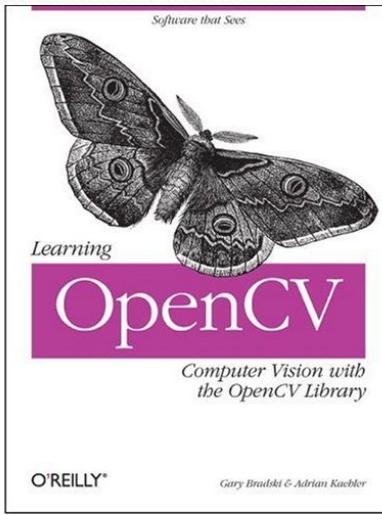
- 10.1. Планирование движений в условии неопределённости.
- 10.2. Задача локализации робота .
- 10.3. Задача составления карты.
- 10.4. Обучение с подкреплением.

Практические занятия

- Практическое занятие: Изучение основных структур данных библиотеки OpenCV (<http://sourceforge.net/projects/opencvlibrary/>).
- Практическое занятие: Освоение инструментов обработки изображений и видео с помощью библиотеки OpenCV.
- Практическое занятие: Создание приложения для улучшения качества изображения.
- Практическое занятие: Создание системы видеонаблюдения
- Практическое занятие: Создание приложения для сравнения различных методов распознавания образов.
- Практическое занятие: Создание системы локализации и распознавания лиц .
- Практическое занятие: Создание системы OCR.
- Практическое занятие: Калибрация камер.
- Практическое занятие: Стерео зрение.
- Практическое занятие: Этапы создания системы слежения за позой человека по данным с нескольких камер.

Литература

- Л. Шапиро, Дж. Стокман, **Компьютерное зрение**; Пер. с англ. — М.: БИНОМ. Лаборатория знаний, 2006.— 752 с.
- Форсайт Д., Понс Ж. "Компьютерное зрение. Современный подход". Пер. с англ. М.: Издательский дом "Вильямс" , 2004. - 928 с.:
- Стюарт Рассел, Питер Норвиг «**Искусственный интеллект: современный подход(AIMA)**», 2-е издание : Пер. с англ. - М.:Издательский дом «Вильямс», 2005.



<http://szeliski.org/Book/>

- Сайт курса: <https://sites.google.com/site/cvnnsu/>
- Компьютерное зрение*

О курсе

Блог

- Задачи
- Зачет
- Конференция nvidia
- Материалы
- Отмена занятия
- Расписание

Контакты

Лекции

Материалы к зачету

- Вопросы
- Задачи

Материалы практических занятий

Полезные ссылки

Посещаемость

Экзаменационные вопросы

Карта сайта

Обсудить

[Перейти в группу](#)

О курсе

Отмена занятия: В связи с эвакуацией в университете 15.03 занятие в 14-40 отменяется.

Отправлено 29.04.2011 2:16 пользователем Alexander Shishkov

Конференция nvidia: В связи с проводимой 1го марта конференцией "Применение гибридных высокопроизводительных вычислительных систем для решения научных и инженерных задач" лекции завтра не будет. Доклады по теме компьютерного зрения: 10.35-10 ...

Отправлено 29.04.2011 2:15 пользователем Alexander Shishkov

Зачет: Последняя возможность сдать зачет в Вас будет в понедельник 27-го лекабря в

Сайт посвящен спецкурсу "Компьютерное зрение", который читается на факультете вычислительной математики и кибернетики ННГУ им. Н.И. Лобачевского.

Курс включает в себя следующие главы:

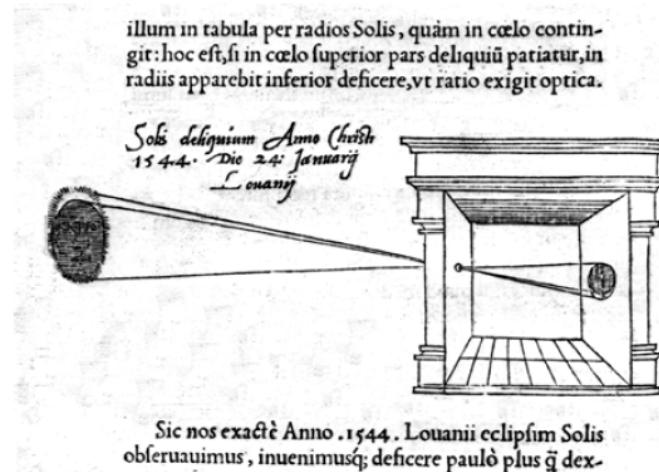
- методы получения и обработки изображения;
- методы видеонаписи;

Введение в проблематику. формирование изображения.

- Введение. Определение тех. зрения. Текущий статус этой области знаний. Цели курса.
- Примеры систем технического зрения. Демонстрации некоторых программ.
- Формирование изображений. Камера Обскура. Перспективная проекция. Использование линз.
- Устройство современной цифровой камеры. Получение растра. Основные искажения при формировании изображения.
- Устройство системы технического зрения.
- Устройство человеческого глаза.
- Виды цифровых изображений.
- Основные форматы хранения изображений.

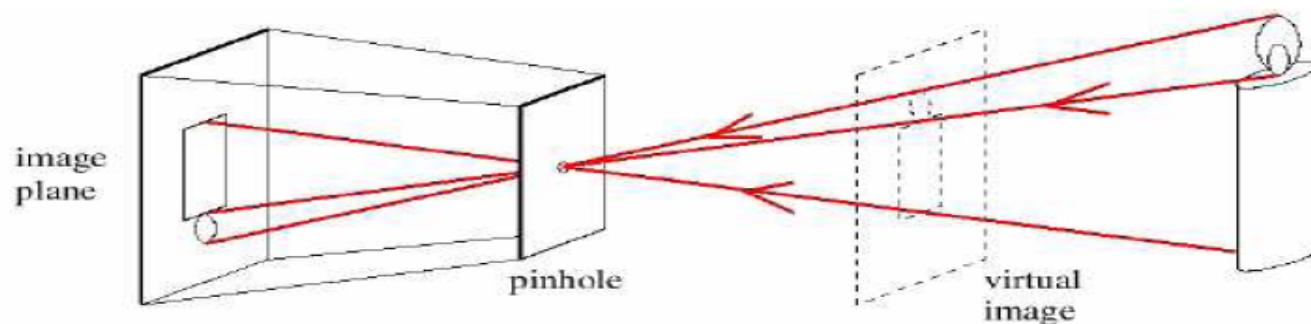
Формирование изображения

Camera Obscura



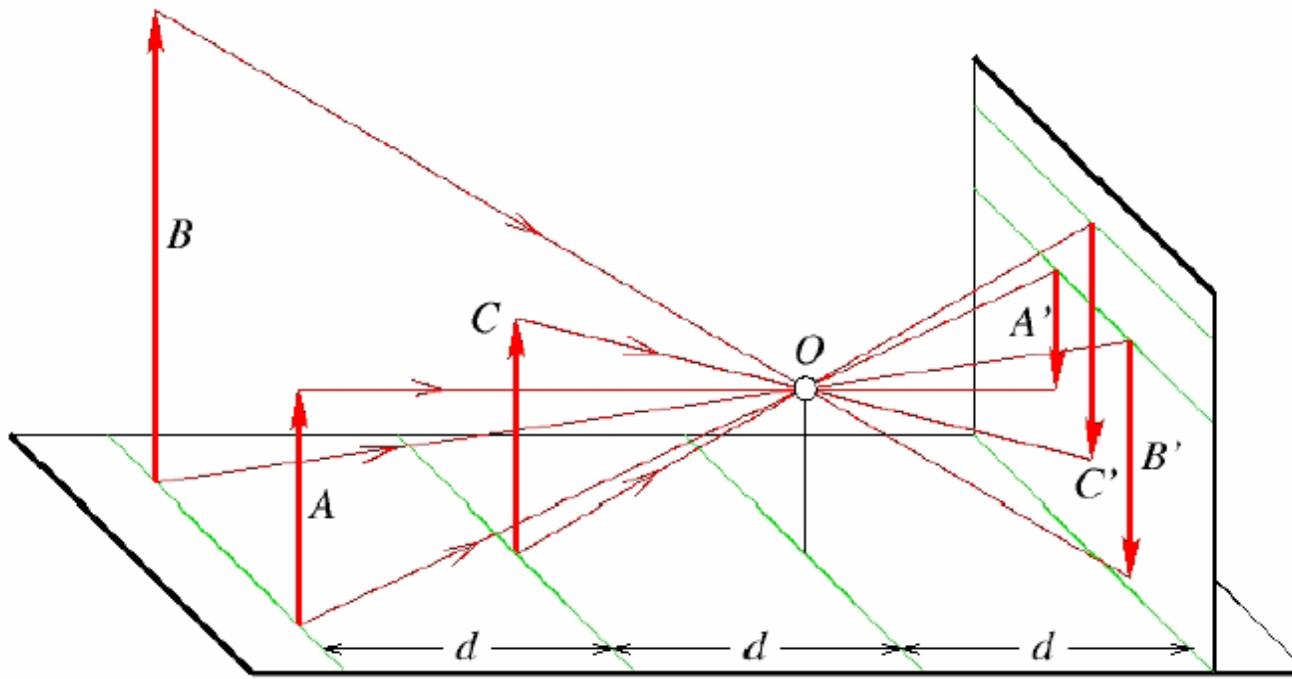
Как
использовали
обскуру?

"When images of illuminated objects ... penetrate through a small hole into a very dark room ... you will see [on the opposite wall] these objects in their proper form and color, reduced in size ... in a reversed position, owing to the intersection of the rays". Da Vinci



Forsyth & Ponce Figure 1.2

Формирование изображений – перспективная проекция

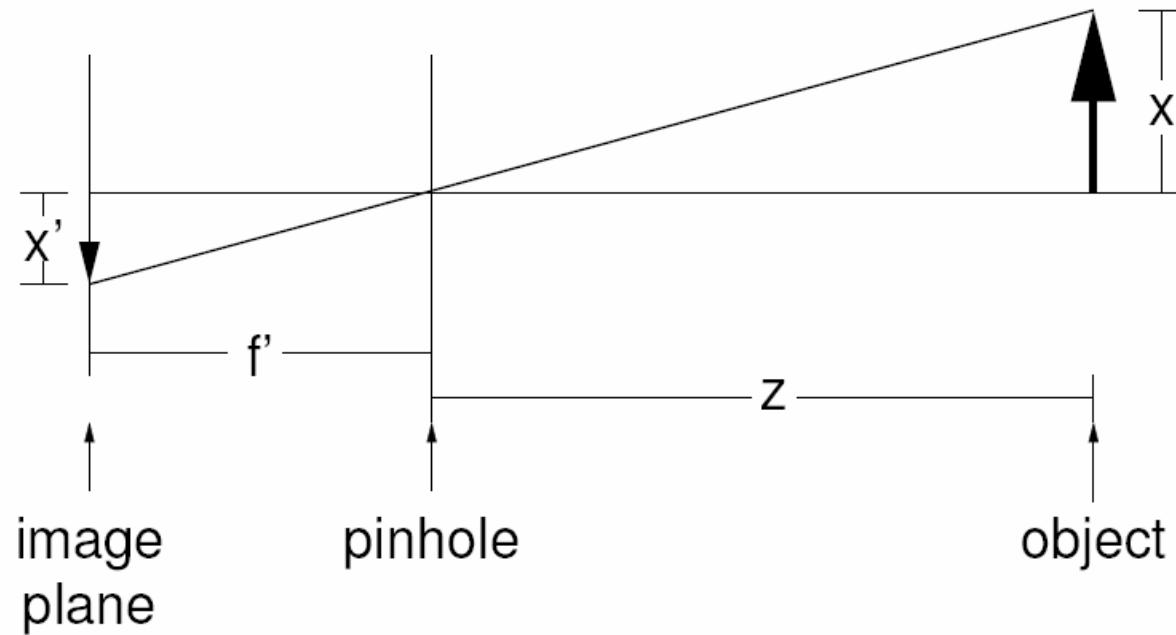


Forsyth & Ponce Figure 1.3a

- Размер проекции объекта обратно пропорционален расстоянию до него.
- Точка проектируется в точку.
- Линия проектируется в линию.
- Углы не сохраняются.
- Параллельные линии могут пересекаться.

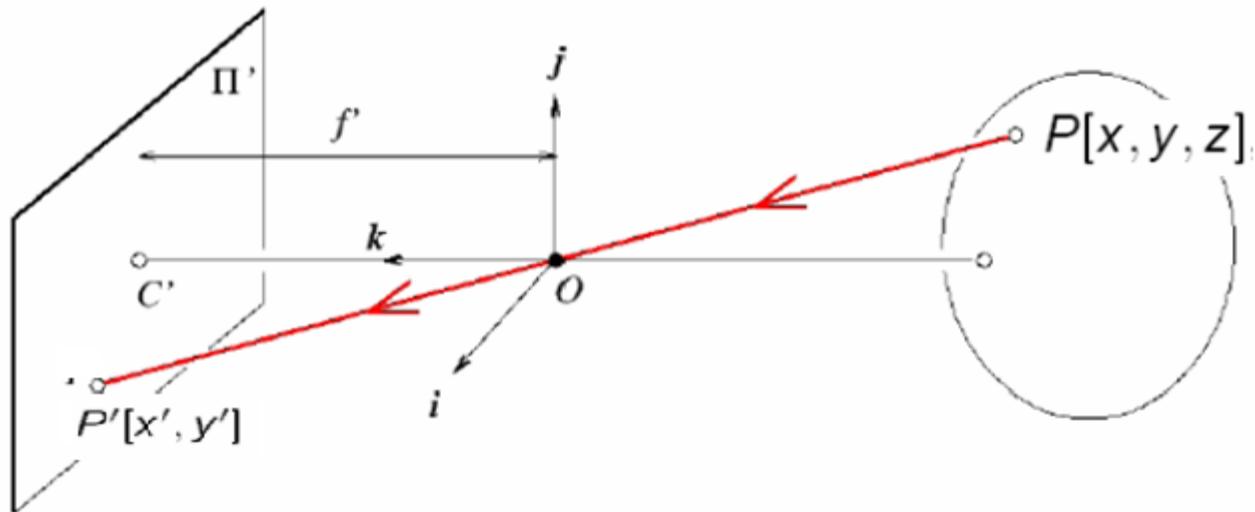
Формирование изображений

Pinhole camera in 2D



$$X' = (f' / Z) X$$

Формирование изображений – перспективная проекция



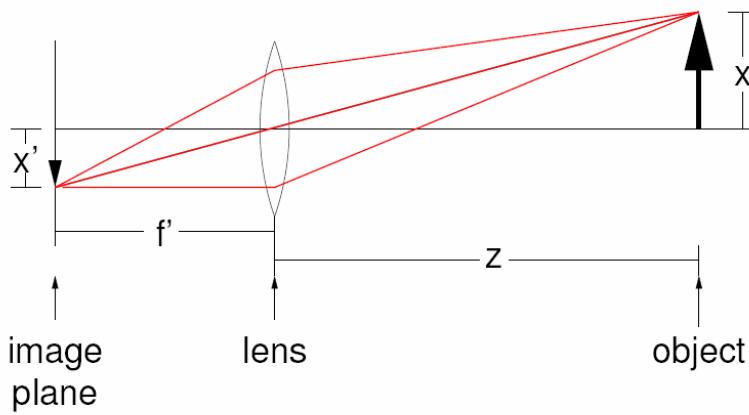
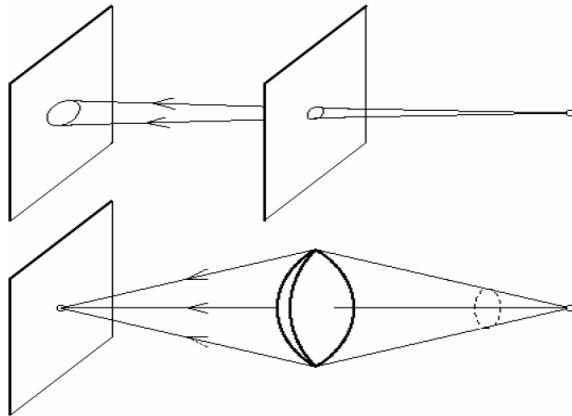
$$x' = f' \frac{x}{z}$$

$$y' = f' \frac{y}{z}$$

Точка схода

- Линия проходящая через точку (X_0, Y_0, Z_0) в направлении (U, V, W) описывается уравнением (X_0+kU, Y_0+kV, Z_0+kW)
- Проекция этой линии записывается как $P_k = (f(X_0+kU)/(Z_0+kW), f(Y_0+kV)/(Z_0+kW))$
- При $k \rightarrow \infty$ получаем “точку схода”
 $P_{\infty} = (fU/W, fV/W)$
- Видно, что точка схода не зависит от (X_0, Y_0, Z_0)

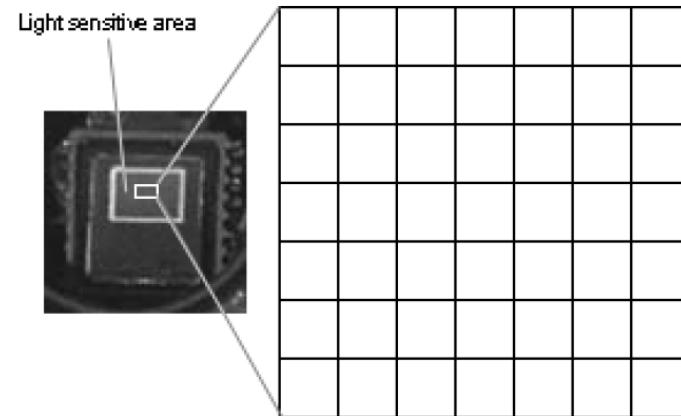
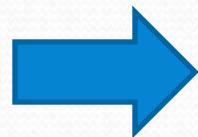
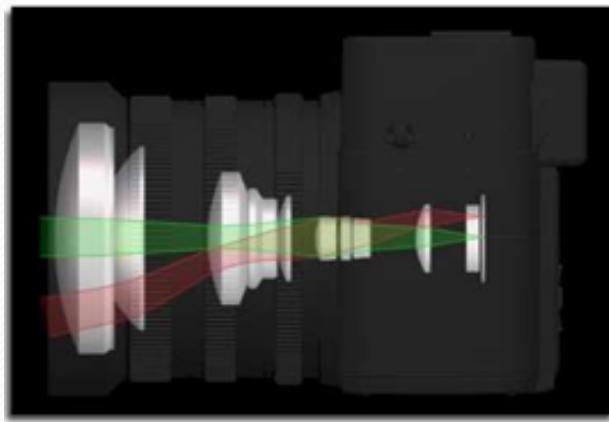
Формирование изображений – использование линз



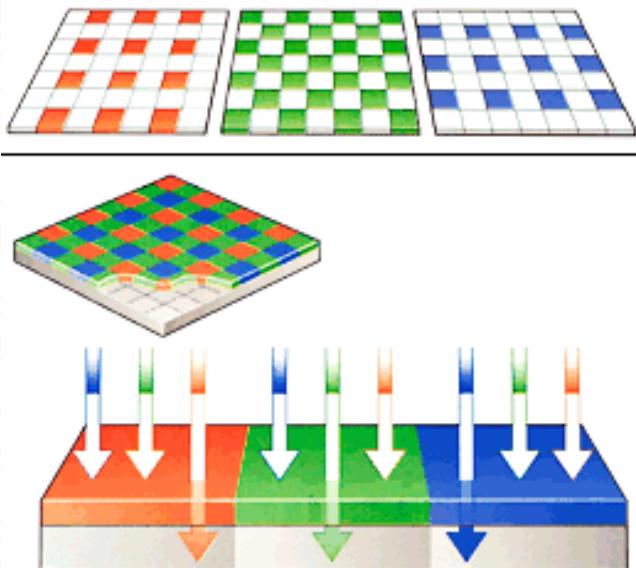
A lens follows the pinhole model for objects that are in focus.

Почему некоторые
объекты могут
быть не в фокусе?

Формирование изображений



<http://huizen.ddsw.nl/bewoners/maan/imaging/camera/ccd1.gif>



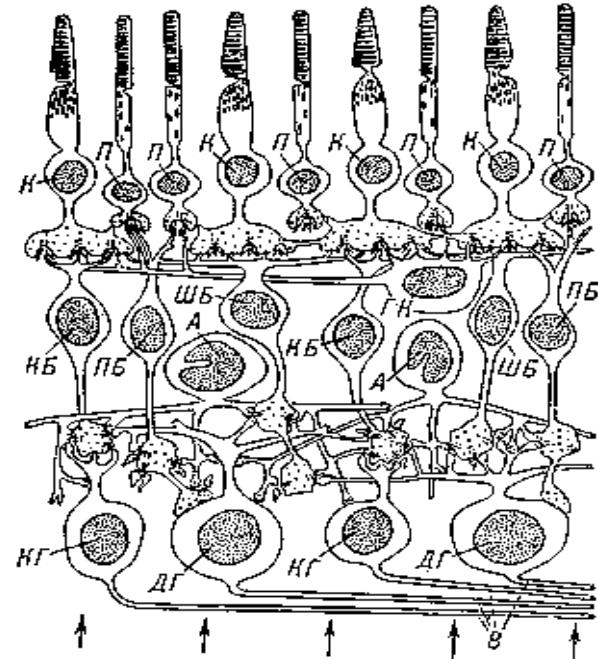
В большинстве сенсоров каждая ячейка "покрыта" **красным, синим или зеленым** фильтром. Каждый фильтр пропускает на светочувствительную ячейку преимущественно свою составляющую света

Виды искажений

- Блики.
- Рассеяние и переотражения света в линзах и внутри объектива.
- Геометрические искажения (радиальная дисторсия).
- В дешёвых камерах плоскость линзы и приёмника могут быть не параллельны.
- Шумы в матрице. Перетекание сигнала.
- “Смазывание” движения.
- Ошибки фокусировки
- Неправильная экспозиция

Устройство глаза

- Зрачок 1..8 мм
- 100 млн палочек и 5 млн колбочек
- 330 нм -730 нм
- Распределение не однородно. (у человека фовеа в среднем равна 1 угловой мин, плотность $1,8 \cdot 10^5$ на 1 мм).
- Есть слепое пятно
- Палочки могут реагировать на один фотон, но связаны одним нейроном. Колбочки наоборот.
- У человека, обезьян и рыб обнаружены колбочки с тремя разными кривыми спектральной чувствительности, максимумы которых у человека находятся в синей, зелёной и красной областях спектра.
- FPS (frames per second): лягушка 15—20 гц, человек — до 50—60 гц, муха — до 250



Зачем два глаза?

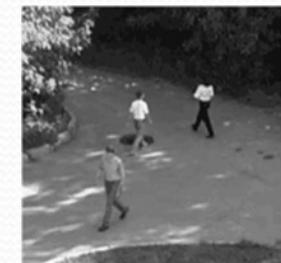
Для эрудиции

- Средние века - наблюдение за воздействием света на соли серебра.
- Нипс в 1816 г. использовал камеру-обскуру и бумагу обработанную хлоридом серебра для получения изображения и закреплял его азотной кислотой.
- В 1889 появилась фотоплёнка
- 1895 – изобретение кино братьями Люмьер
- Цветная фотография 1908 г.
- Телевидение 1920 г.
- Первые признанные работы по комп. зрению в 60-ых годах.

Цифровые изображения

Цифровое изображение – это двумерный массив $I[r,c]$ элементов (пикселей), которые представляют собой одно или нескольких дискретных значений.

- *Определение. Бинарное изображение* – это цифровое изображение $B[r,c]$, пиксели которого принимают значения 0 или 1.
- *Определение. Полутоновое (серое, монохромное, Ч/Б) изображение* – это цифровое изображение $I[r,c]$, у которого каждому пикслю соответствует одно значение интенсивности (яркости).
- *Определение. Мультиспектральное изображение* – это цифровое изображение $M[r,c]$, у которого каждому пикслю соответствует вектор значений. (у цветных изображений размерность вектора = 3).



Форматы изображений

- Заголовок (ширина, высота, дата создания, тип камеры, другие метаданные...) + растр.

Растр (пиксельные данные):

Серое (Ч/Б) - $I_o I_1 I_2 \dots$

Цветное - $R_o G_o B_o R_1 G_1 B_1 R_2 G_2 B_2 \dots$

- PBM, PGM, PPM (Portable Bit map). (заголовок и растр хранятся в ASCII)
- BMP (заголовок и растр хранятся бинарно)
- GIF (Graphics Interchange Format) (под цвет выделяется всего 8 бит + сжатие без потерь LZW, в одном файле может храниться несколько изображений)
- JPEG (Joint Photographic Expert Group) (сжатие с потерями 20:1. Max 64K x 64K, цвет 24 бита)

Обработка изображений и

низкоуровневый анализ.

Инструменты обработки бинарных изображений.

- Нахождение связанных компонент.
- Морфологические операции.
- Инвариантные свойства связанных компонент.
- Векторизация бинарных изображений. Алгоритм.
Применения.
- Бинаризация изображения. Анализ гистограммы.
Метод Отсу.

Бинарные изображения. Анализ и морфологические операции.

Бинарные изображения.



4-ёх и 8-ми связанность



4 Neighborhood



8 Neighborhood

0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	1	0	0
0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	1	0	0
0	1	1	0	1	1	1	0	0
0	1	1	0	1	1	1	0	0
0	0	0	0	0	0	0	0	0

Сколько 4-х связанных и
8-ми связанных компонент
изображено?

Рекурсивный алгоритм поиска связанных компонент

```
procedure recursive_connected_components(B, LB);
{
LB := negate(B);
label := 0;
find_components(LB, label);
print(LB);
}
```

```
procedure find_components(LB, label);
{
for L := 0 to MaxRow
  for P := 0 to MaxCol
    if LB[L,P] == -1 then
      {
        label := label + 1;
        search(LB, label, L, P);
      }
}
```

```
procedure search(LB, label, L, P);
{
LB[L,P] := label;
Nset := neighbors(L, P);
for each (L',P') in Nset
  {
    if LB[L',P'] == -1
    then search(LB, label, L', P');
  }
}
```

Step 1.

-1	-1	0	-1	-1	-1
-1	-1	0	-1	0	0
-1	-1	-1	-1	0	0

Step 2.

1	-1	0	-1	-1	-1
-1	-1	0	-1	0	0
-1	-1	-1	-1	0	0

Step 3.

1	1	0	-1	-1	-1
-1	-1	0	-1	0	0
-1	-1	-1	-1	0	0

Step 4.

1	1	0	-1	-1	-1
1	-1	0	-1	0	0
-1	-1	-1	-1	0	0

Step 5.

1	1	0	-1	-1	-1
1	1	0	-1	0	0
-1	-1	-1	-1	0	0

1	1	0	1	1	1	0	3
1	1	0	1	0	1	0	3
1	1	1	1	0	0	0	3
0	0	0	0	0	0	0	3
4	4	4	4	0	5	0	3
0	0	0	4	0	5	0	3
6	6	0	4	0	0	0	3
6	6	0	4	0	3	3	3

Морфологические операции

Определение. Структурирующим элементом H назовем вспомогательное бинарное изображение, которое задает отношение соседства в исходном бинарном изображении I .

Операция наращивания (*dilate*) - это множество всех копий структурирующего элемента H центрированных в каждом ненулевом пикселе исходного бинарного изображения I .

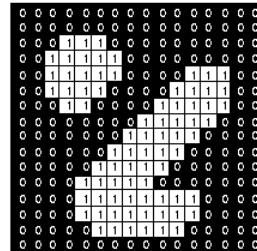
$$I \oplus H = \bigcup_{p \in I} H_p$$

Операция эрозии (*erode*) - сохраняет множество всех пикселей исходного бинарного изображения I , который полностью покрываются структурирующим элементом H .

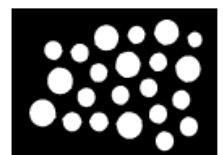
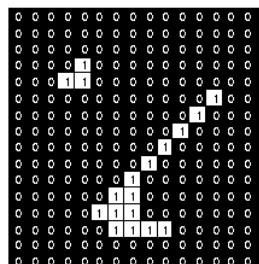
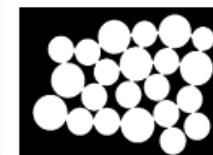
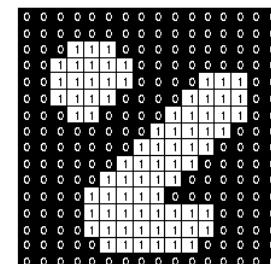
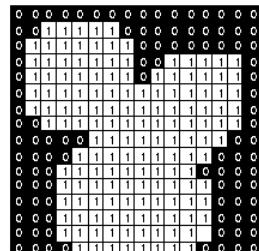
$$I \ominus H = \{p \mid H_p \subseteq I\}$$



4 Neighborhood



8 Neighborhood



Морфологические операции

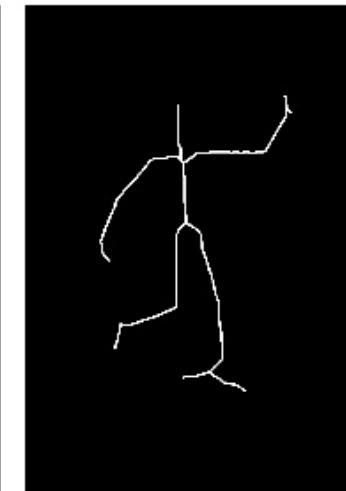
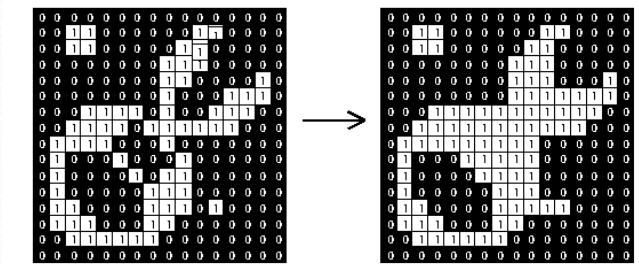
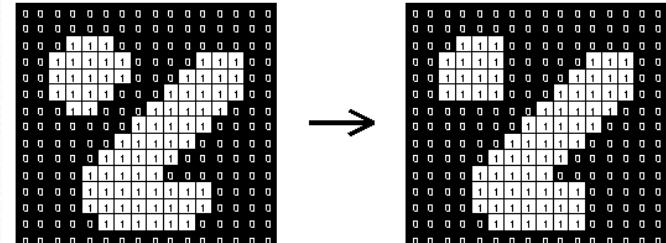
Операция открытия/размыкания (open) - это последовательное применение эрозии и наращивания.

$$I \circ H = (I \ominus H) \oplus H$$

Операция замыкания (close) - это последовательное применение наращивания и эрозии.

$$I \bullet H = (I \oplus H) \ominus H$$

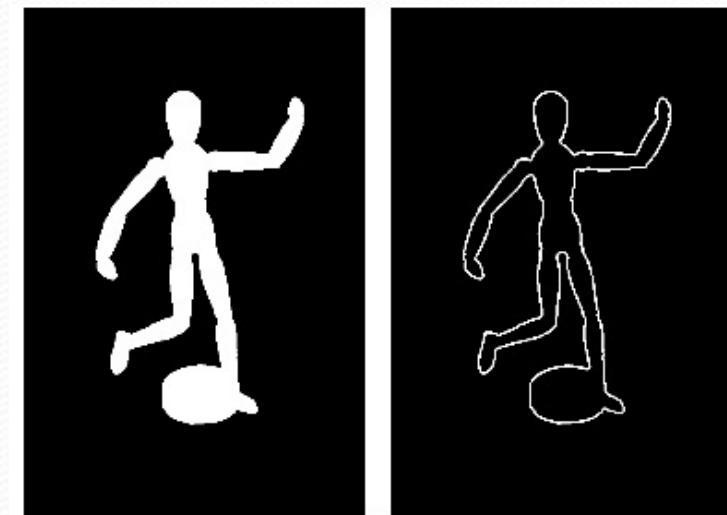
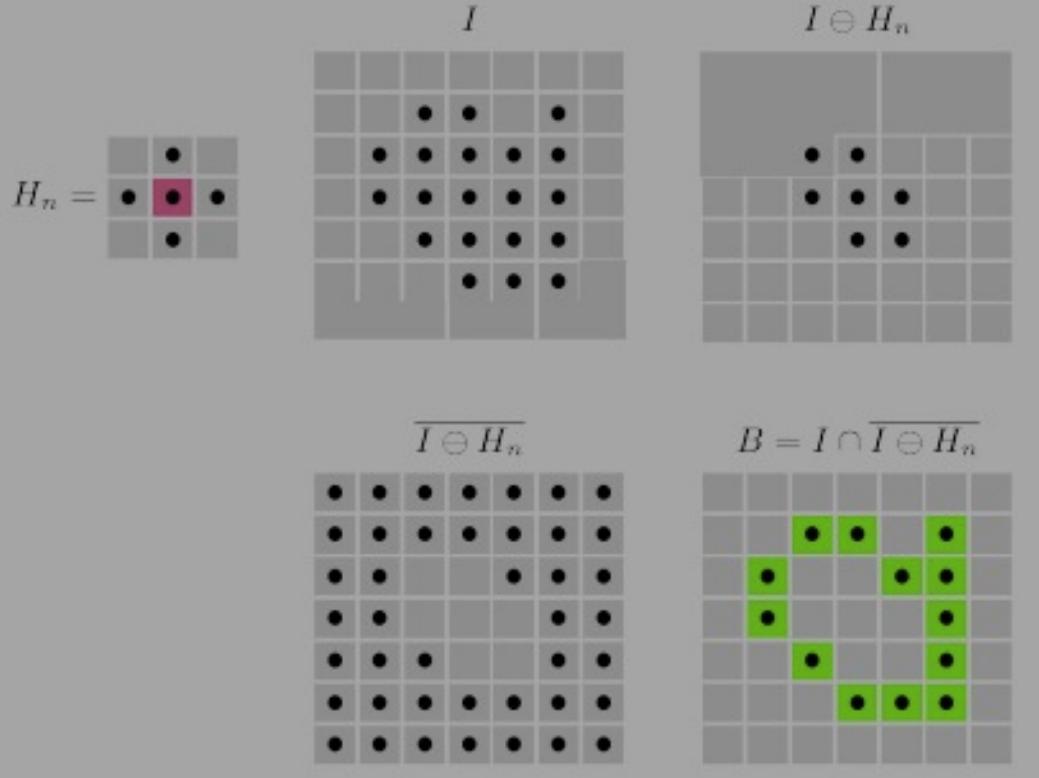
Операция скелетонизации - это последовательное применение эрозии до получения однопиксельных линий.



Repeatedly run erosion, stop when 1-pixel thick

Морфологические операции

Операция оконтурирования (*outline*).



Морфологические операции

Операция удаления шумов (*noise removal*) - это последовательное применение размыкания и замыкания.



Морфологические операции



picture from:
Gary Bradski, Adrian Kaehler, "Learning OpenCV:
Computer Vision with the OpenCV Library"

Morphological gradient:

$$\text{dst} = \text{morph_grad(src, element)} = \text{dilate(src, element)} - \text{erode(src, element)}$$

"Top hat":

$$\text{dst} = \text{tophat(src, element)} = \text{src} - \text{open(src, element)}$$

"Black hat":

$$\text{dst} = \text{blackhat(src, element)} = \text{close(src, element)} - \text{src}$$

Бинарные изображения. Анализ.

Свойства областей.

- Площадь
- Центр тяжести
- 4ёх и 8ми связанный периметр

• Округлость

$$C_1 = \frac{|P|^2}{A}$$

• Описывающий прямоугольник (X,Y,W,H)

• Моменты(центральные и смешанные)

$$\mu_{rr} = \frac{1}{A} \sum_{(r,c) \in R} (r - \bar{r})^2 \quad \mu_{cc} = \frac{1}{A} \sum_{(r,c) \in R} (c - \bar{c})^2$$

$$m_{p,q} = \sum_{i=1}^n I(x, y) x^p y^q$$

инвариантность свойств

$$A = \sum_{(r,c) \in R} 1$$

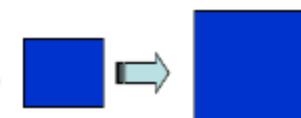
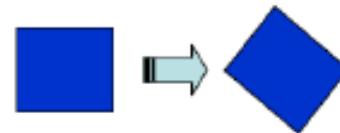
$$\bar{r} = \frac{1}{A} \sum_{(r,c) \in R} r$$

$$\bar{c} = \frac{1}{A} \sum_{(r,c) \in R} c$$

Как отличить бинарное изображение человека от машины?

Инвариантность признаков

- Сдвиг
- Поворот
- Изменение масштаба
- Афинные трансформации
- Изменение освещения

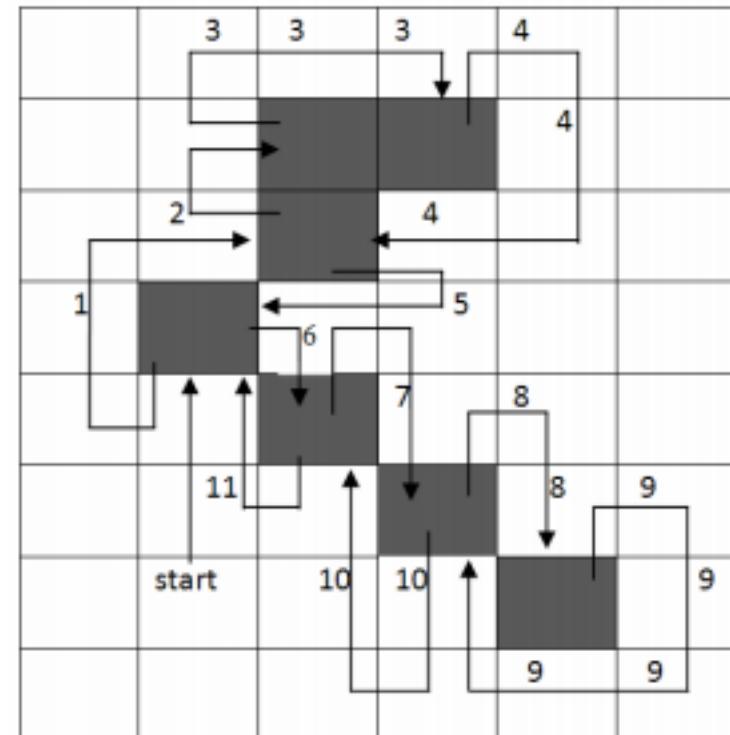


$$(I \rightarrow aI + b)$$
A blue square is transformed into a lighter blue square by changing its lighting, indicated by a grey arrow pointing from the original square to the lighter version.

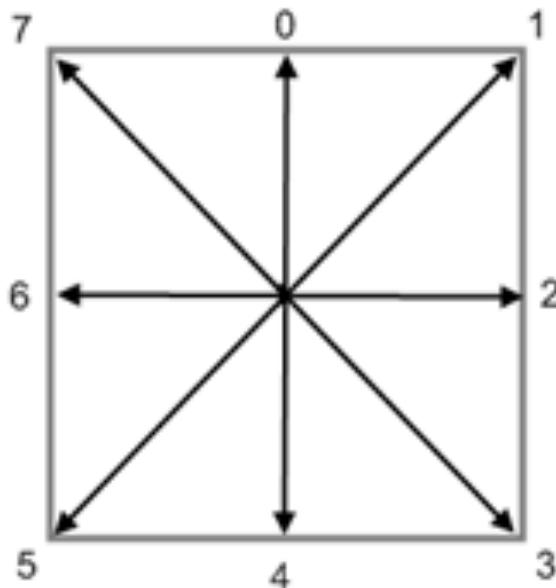
Векторизация бинарных изображений.

- Векторизация – получение векторных данных из растра.
- Переход к векторному представлению позволяет значительно ускорить вычисления, а так же упростить некоторые алгоритмы.
- Алгоритм поиска контуров - Черепаха

Начиная с первого ненулевого пикселя обходятся по часовой стрелке все его соседи пока не встретимся следующий ненулевой пиксель. При этом обход начинается с соседа с которого “вошли” в текущий ненулевой пиксель. Процесс заканчивается когда встретим пиксель с которого начали.

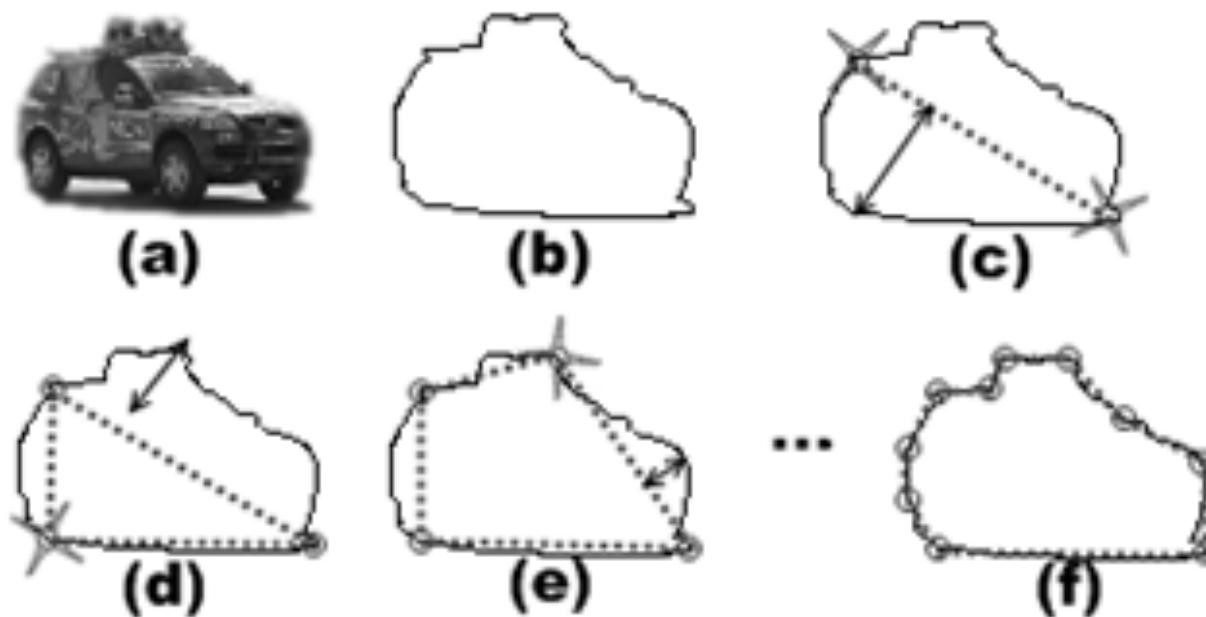


Код Фримана – альтернатива массиву 2D точек



picture from:
Gary Bradski, Adrian Kaehler, "Learning OpenCV:
Computer Vision with the OpenCV Library"

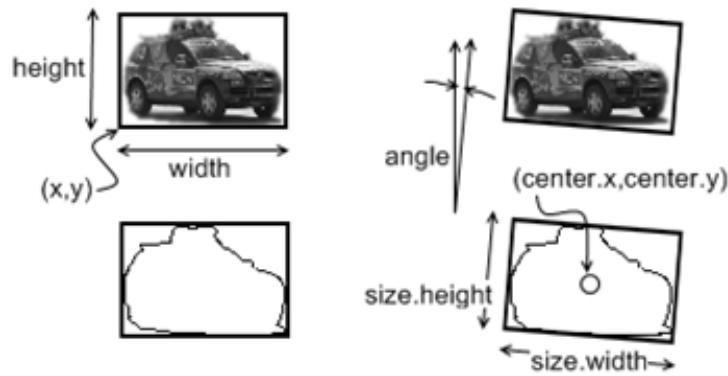
Итеративный алгоритм аппроксимации контура



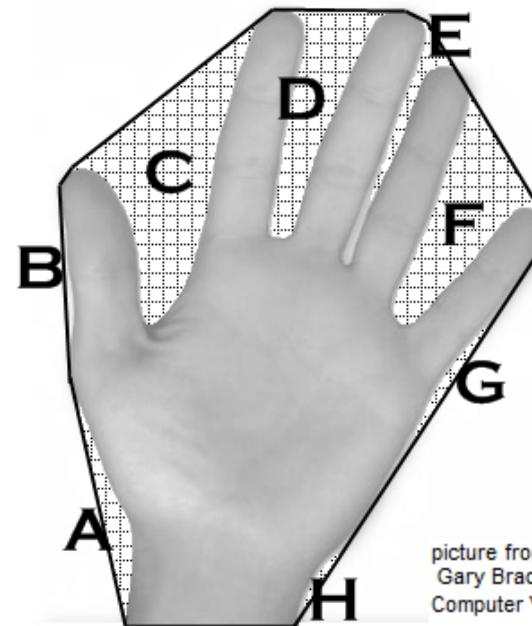
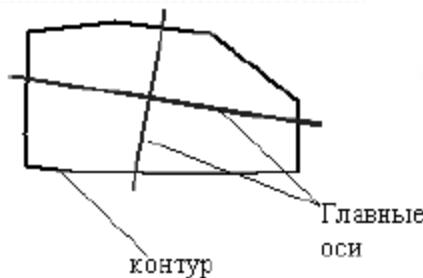
picture from:
Gary Bradski, Adrian Kaehler, "Learning OpenCV:
Computer Vision with the OpenCV Library"

Свойства контура

- Описывающий и минимальный прямоугольники



- Дефекты выпуклости
- Главные оси



picture from:
Gary Bradski, Adrian Kaehler, "Learning OpenCV:
Computer Vision with the OpenCV Library"

Получение бинарных изображений из полуточновых

混沌干涉仪干涉仪干涉仪干涉仪干涉仪干涉仪干涉仪



混沌干涉仪干涉仪干涉仪干涉仪干涉仪干涉仪干涉仪

Получение бинарных изображений из полуточновых.

- $B(x,y) = 1$, если $A(x,y) >$ порог, иначе $B(x,y) = 0$



a) original image



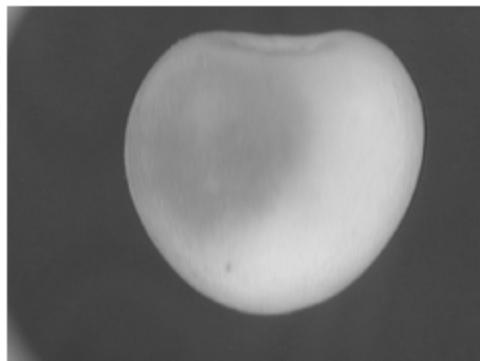
b) pixels below 93



c) pixels above 93

- Анализ гистограммы

$$h(m) = |\{(r,c) \mid I(r,c) = m\}|$$



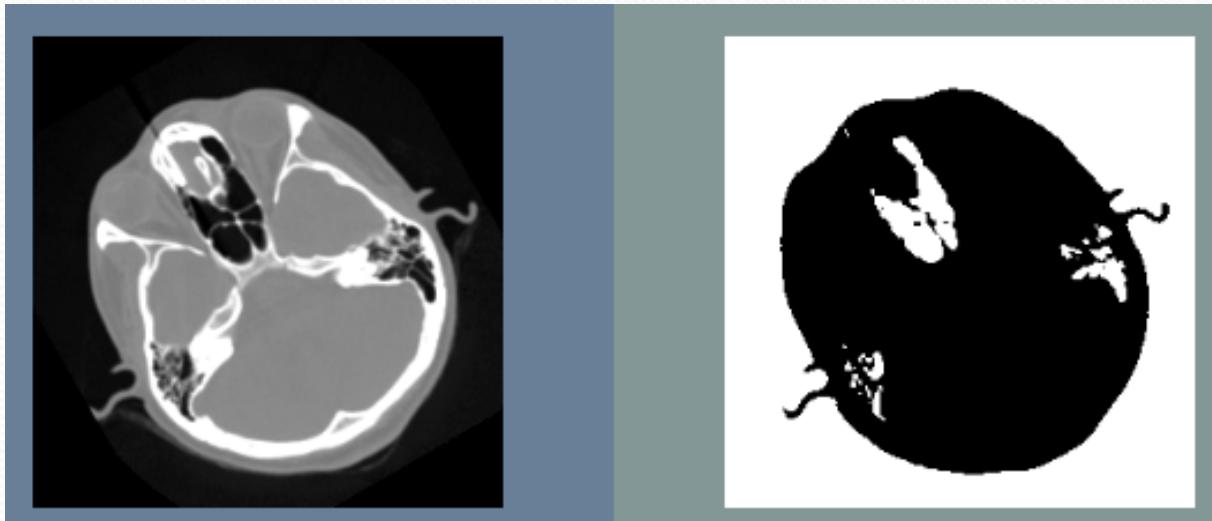
Как найти оптимальный порог? Какие критерии?

Получение бинарных изображений из полуточновых. Метод Отсу.

- Метод основан на минимизации внутригрупповой дисперсии.

$$\min_T \sigma(pSrc(x, y) \leq T) + \sigma(pSrc(x, y) > T))$$

$$\sigma(pSrc(x, y)) = \sqrt{\frac{\sum_{\substack{x, y \geq 0 \\ x < roisize.width \\ y < roisize.height}} (pSrc(x, y) - mean)^2}{roisize.height \cdot roisize.width}}$$

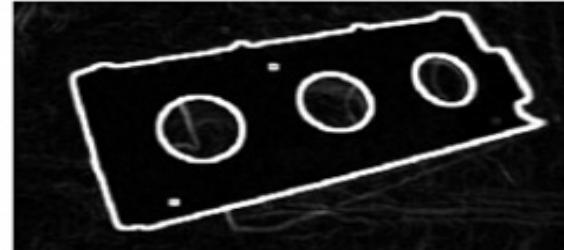


Обработка и низкоуровневый анализ полутоновых и цветных изображений.

- Линейная фильтрация изображения. Свертка.
- Повышение чёткости.
- Подсчёт градиентов.
- Выделение границ объектов.
- Процедура трансформации расстояния.
- Нахождение угловых точек на изображении.
- Понятие гистограммы и улучшение контрастности.
- Выравнивание контрастности двух изображений.
Эквализация гистограммы.
- Мультиспектральные изображения. Виды цветовых пространств.
- Методы улучшения цветных изображений.
- Методы сегментации цветных изображений.

Обработка полутонаовых изображений.

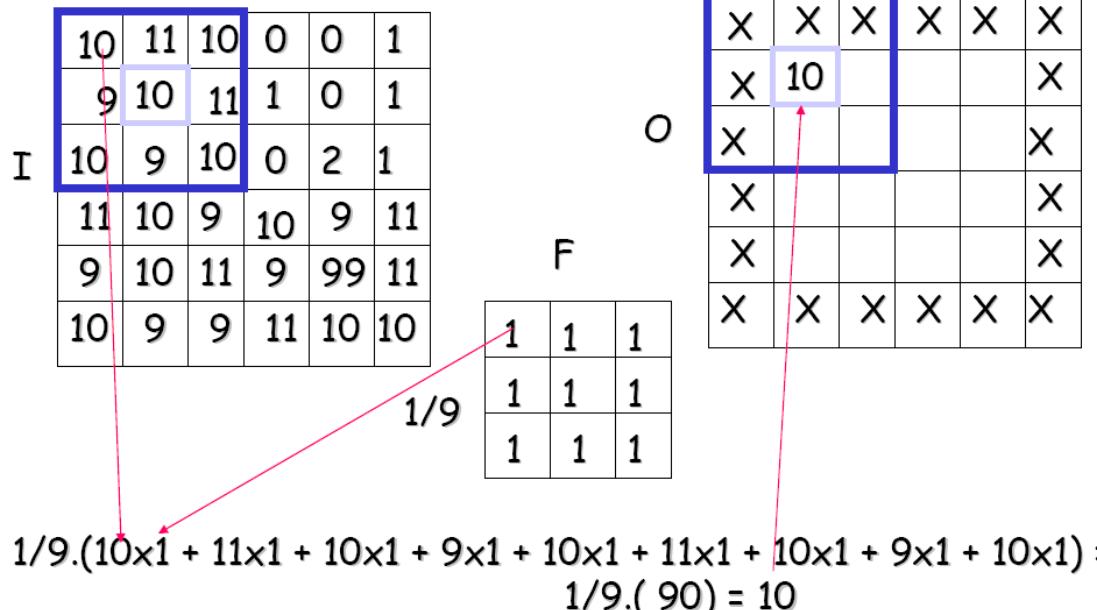
- Задачи фильтрации: удаление шумов, повышение чёткости, осветление тёмных участков.
- Задачи обнаружения низкоуровневых признаков: рёбра (для поиска края дороги/полосы, для векторизации карты или для контроля качества изделий)



Линейная фильтрация

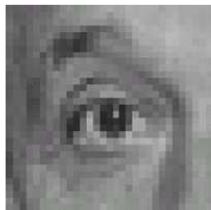
- Линейность: $F(I_1+I_2) = F(I_1)+F(I_2)$
- Пиксели отфильтрованного изображения являются взвешенной суммой пикселов исходного изображения

$$I'(X, Y) = \sum_{i=-k}^k \sum_{j=-k}^k F(i, j) I(X - i, Y - j)$$



Сумма элементов в ядре свёртки равна 1!

Линейная фильтрация

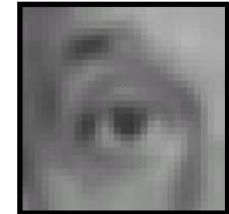


Original

0	0	0
0	1	0
0	0	0



Filtered
(no change)

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


Blur (with a
box filter)



Original

0	0	0
0	0	1
0	0	0



Shifted left
By 1 pixel



Original

$$\begin{array}{|ccc|} \hline 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \\ \hline \end{array} - \frac{1}{9} \begin{array}{|ccc|} \hline 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ \hline \end{array}$$

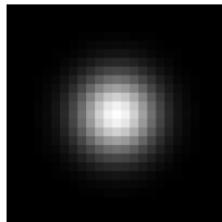
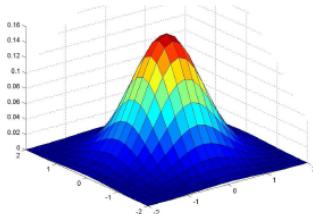


Sharpening filter

- Accentuates differences with local average
- Also known as Laplacian

Фильтрация изображений

- Гауссовская фильтрация



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 × 5, $\sigma = 1$

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

- Медианная фильтрация (медленнее гауссовской, но лучше сохраняет границы)

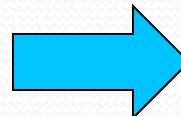
Медиана{5, 50, 1, 10, 100} = ?

Билатеральная фильтрация изображений

$$I^{\text{filtered}}(x) = \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|)$$

where:

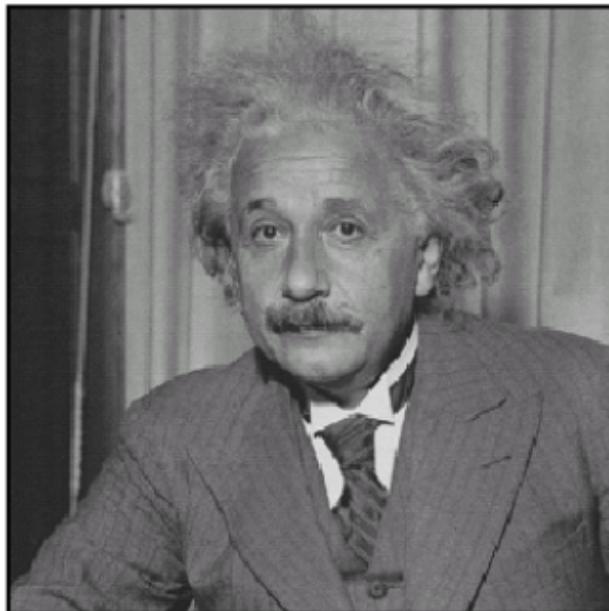
- I^{filtered} is the filtered image;
- I is the original input image to be filtered;
- x are the coordinates of the current pixel to be filtered;
- Ω is the window centered in x ;
- f_r is the range kernel for smoothing differences in intensities. This function can be a Gaussian function;
- g_s is the spatial kernel for smoothing differences in coordinates. This function can be a Gaussian function;



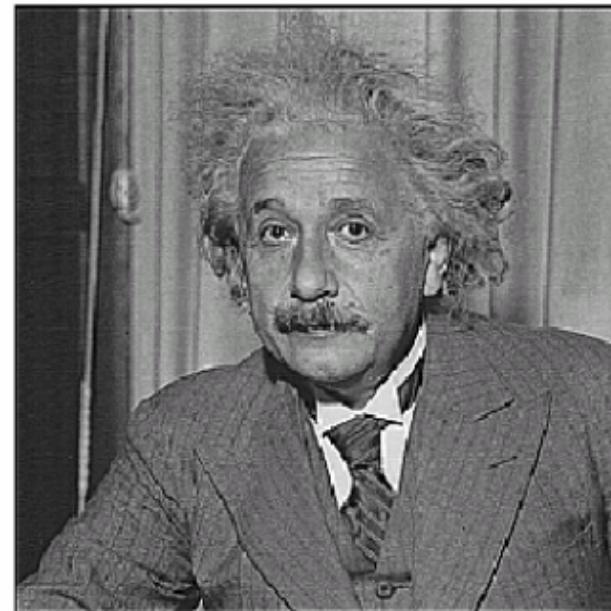
Повышение чёткости

(рекомендуется после сильного шумоподавления или когда фокус неудачный)

- $\text{Image}_1[i,j] = \text{Image}[i,j] - (\text{Image} * \text{Gaussian})[i,j]$
- $\text{Result}[i,j] =$
 $\text{Image}[i,j] \text{ if } |\text{Image}_1[i,j]| \leq T$
 $\text{Image}[i,j] + k * \text{Image}_1[i,j] \text{ if } |\text{Image}_1[i,j]| > T, \quad k > 0$



before



after

Подсчёт градиентов

$$\frac{\partial f}{\partial x} = \lim_{\varepsilon \rightarrow 0} \left(\frac{f(x + \varepsilon, y) - f(x, y)}{\varepsilon} \right) \rightarrow \frac{\partial f}{\partial x} \approx \frac{f(x_{n+1}, y) - f(x_n, y)}{\Delta x}$$

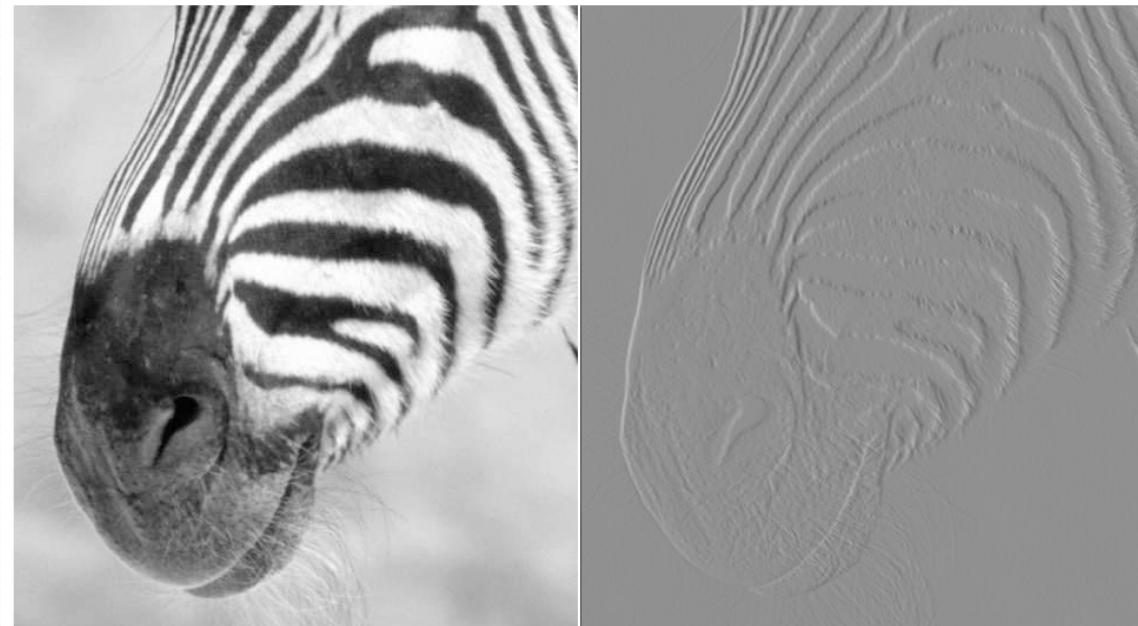
-1	1
----	---

Фильтры Собеля более устойчивые к шуму градиенты

$$\frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\frac{1}{8} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$[I_x, I_y]$$



Лапласиан

$$dst(x,y) = d^2src/dx^2 + d^2src/dy^2$$

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



Интегральное изображение

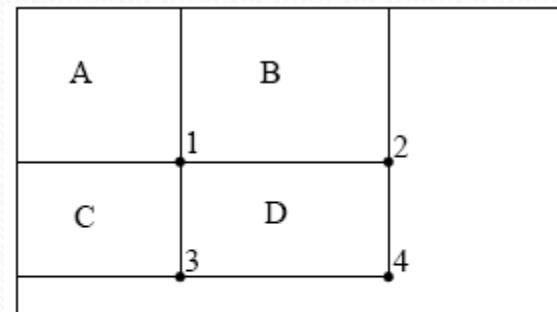
$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'),$$

5	2	3	4	1
1	5	4	2	3
2	2	1	3	4
3	5	6	4	5
4	1	3	2	6

Original image

0	0	0	0	0	0
0	5	7	10	14	15
0	6	13	20	26	30
0	8	17	25	34	42
0	11	25	39	52	65
0	15	30	47	62	81

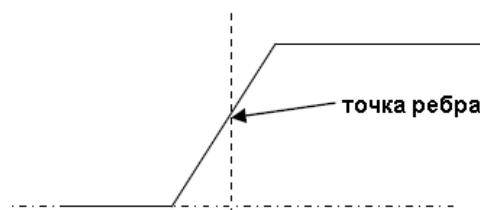
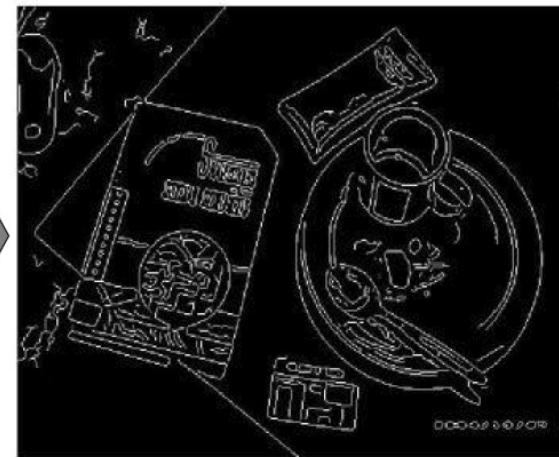
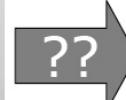
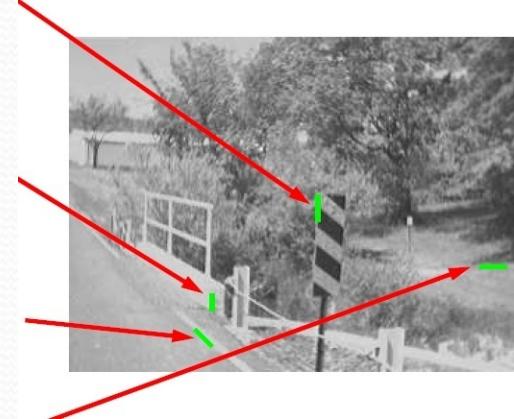
Integral image



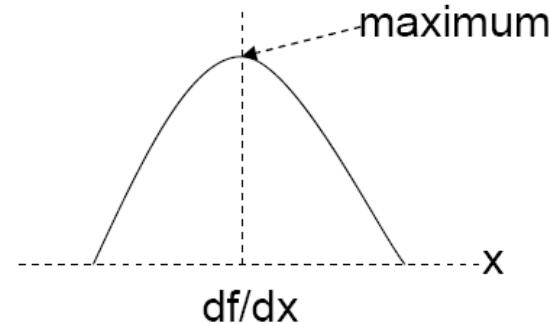
Сумма внутри D равна $4 + 1 - (2 + 3)$.

Выделение краёв/рёбер

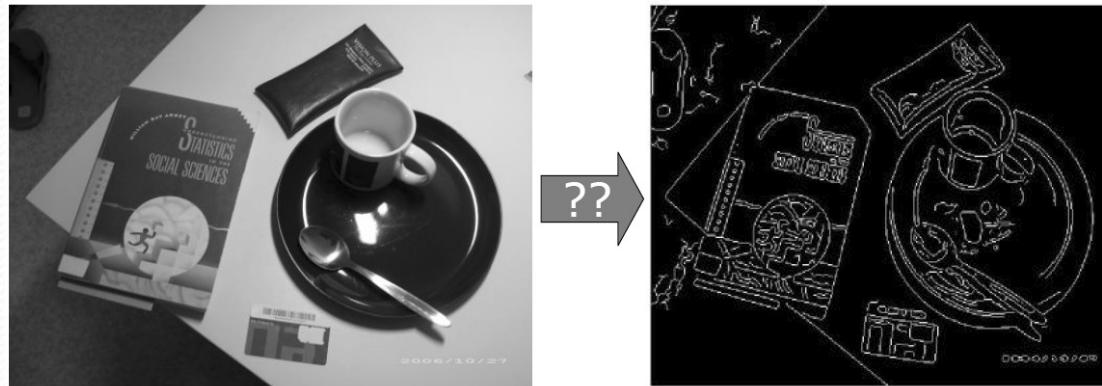
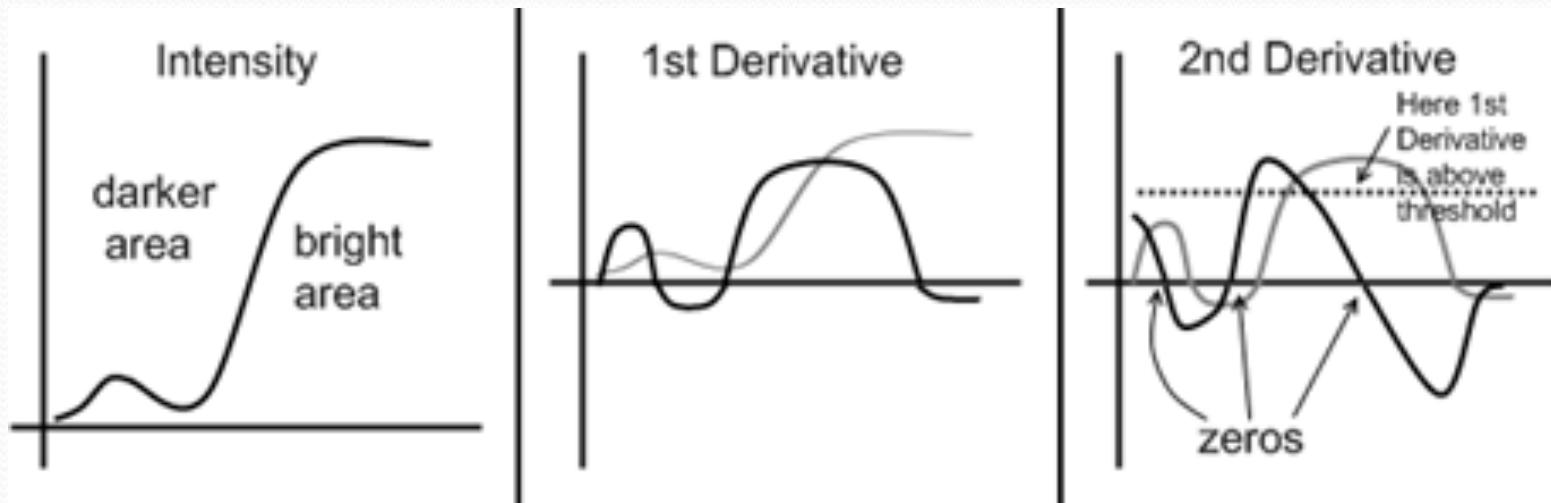
Цель: найти резкие не шумовые
перепады интенсивности,
соответствующие границам объектов на
сцене.



интенсивность



Выделение краёв с помощью Лапласиана



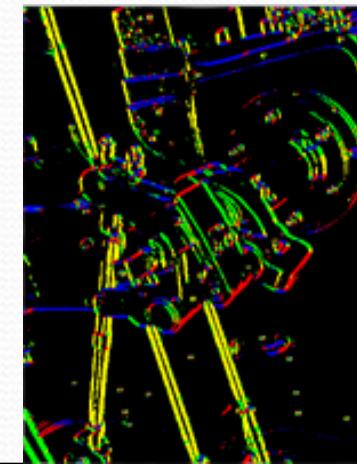
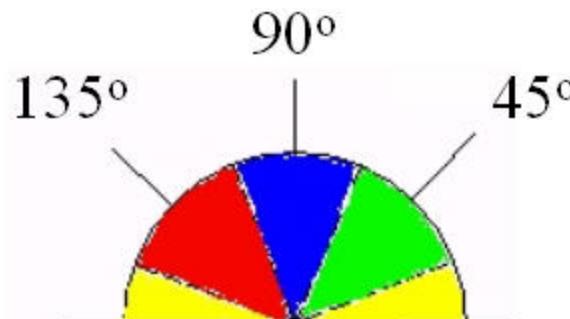
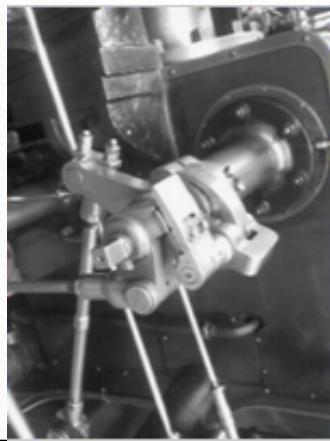
Метод Canny

- Шаг 1. Находятся градиенты \mathbf{G}_x и \mathbf{G}_y
- Шаг 2.
Считываются магнитуда и направление градиентов

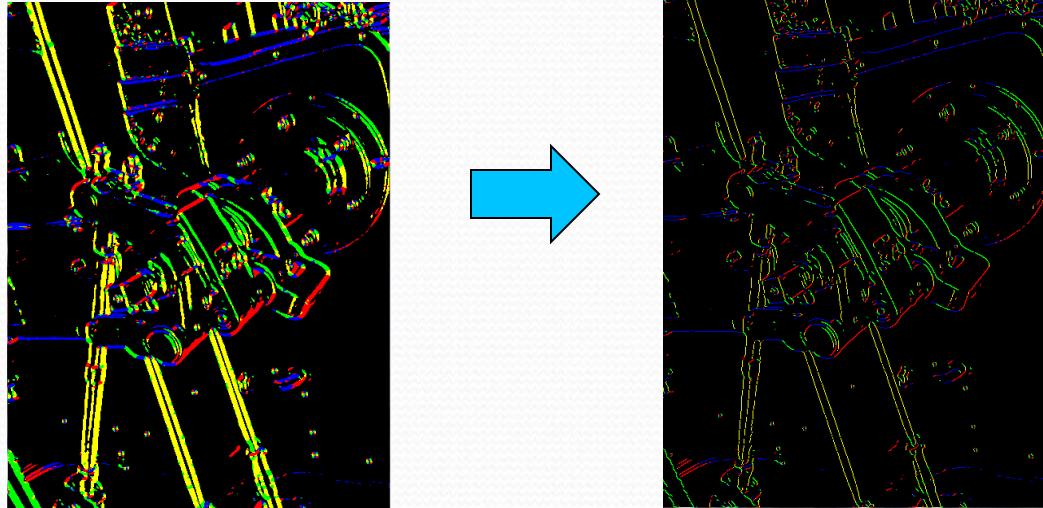
$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2} \quad \Theta = \arctan\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)$$

- Рассматриваются только градиенты с магнитудой $> T_1$
- Все направления градиентов дискретизируются

$$d_k = 0^\circ, 45^\circ, 90^\circ, 135^\circ$$



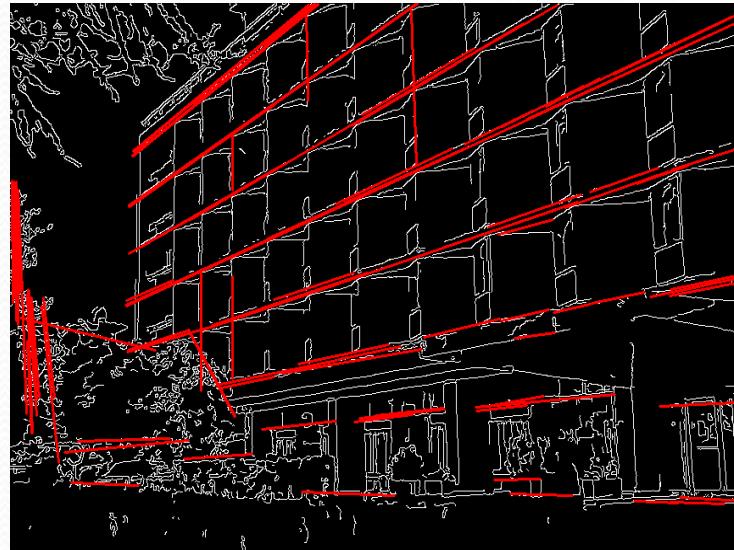
- Шаг 3. Подавление не максимумов вдоль ребра. Рассматриваются два соседа в направлении градиента и нелокальные максимумы обнуляются.



- Шаг 4. Там где край объекта имеет слабый градиент алгоритм может “оборвать” ребро. Рассматривается дополнительный порог T_2 . Будем считать, что пиксели с магнитудой градиента $> T_2$ точно принадлежат краю объекта. Помещаем все краевые пиксели, полученные на шаге 3 с магнитудой градиента $> T_2$ в стек. Из стека поочередно выбираем краевой пиксел и рассматриваем его окрестность. Если в окрестности находится пиксель с магнитудой в интервале $[T_1, T_2]$, то добавляем его в массив краевых точек и добавляем в стек.

Применение рёбер:

нахождение примитивов



- Применяется в 3D реконструкции, оценке полосы движения, классификации объектов...

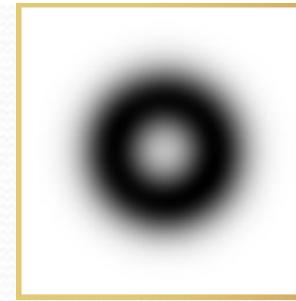
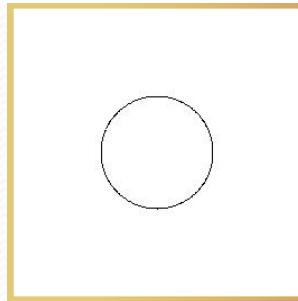
Трансформация расстояния

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

Binary Image

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	2	2	2	1	0
0	1	2	3	2	1	0
0	1	2	2	2	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

Distance transformation



Используется для решения многих задач технического зрения :
сравнение изображений, шаблонный поиск , анализ формы.

Как осуществить
шаблонный поиск



Figure 6-13. Here we first ran a Canny Edge Detector, with the `param1=100` and `param2=200`. The distance transform was then run with the output scaled by a factor of 5 to aid in visibility.

Домашнее задание 1: реализовать фильтрацию изображения с использованием описанных выше техник

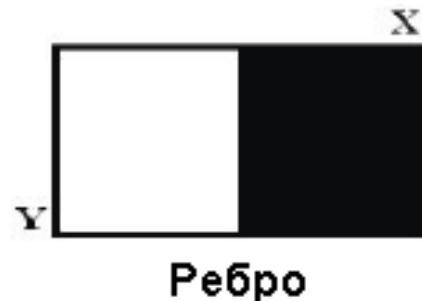
1. Исходное цветное изображение преобразуется в полуточновое
2. Считываются градиенты изображения
3. По градиентам находятся рёбра изображения методом Канни
4. Для найденных рёбер строится карта расстояний $D[i,j]$
5. В каждом пикселе $[i,j]$ производится фильтрация усреднением с размером фильтра равным $k^*D[i,j]$. Т.е. чем дальше от ребра, тем больше апертура (больше подавление шума).
6. Для существенного ускорения вычислений в случае фильтрации усреднением можно использовать интегральные изображения!

Нахождение углов на изображении

- Угол и ребро.



Угол

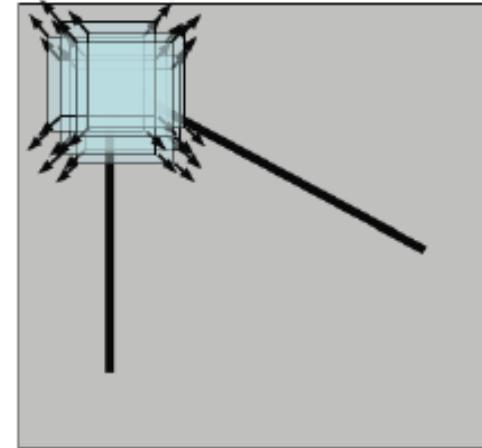
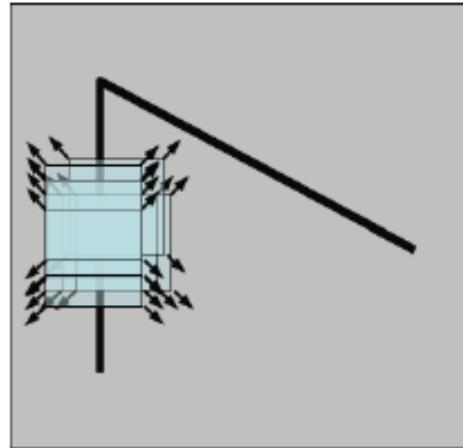
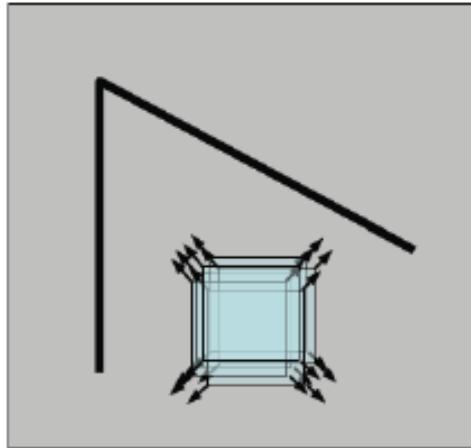


Ребро

Как отличить
угол от ребра?

- Позиции угловых точек и их количество являются важными характеристиками изображения. Часто используются в 3D, в калибрации, в задаче поиска объекта, в задаче слежения за объектом.

Нахождение углов на изображении

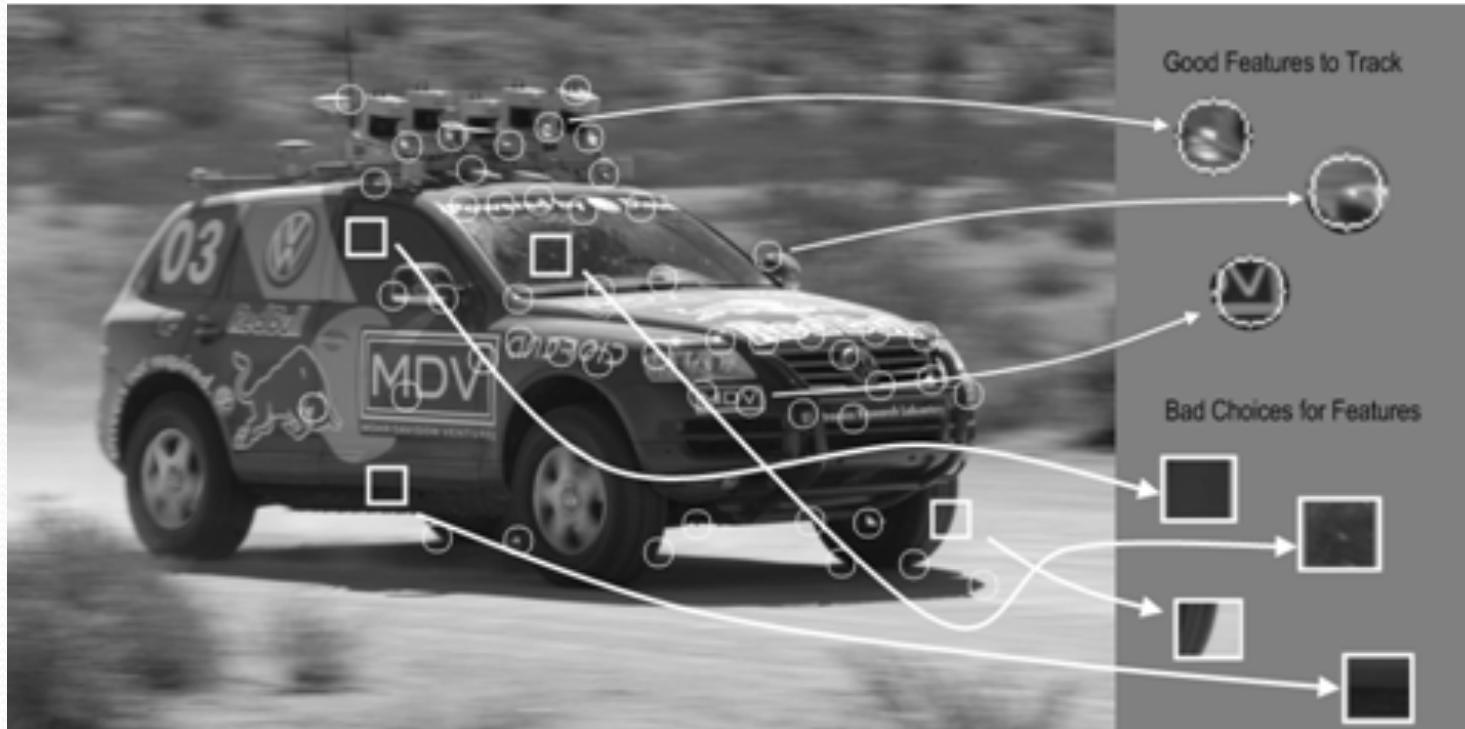


Однородная
область –
нет
изменений в
любом
направлении

Ребро –
нет
изменений в
направлении
ребра

Угловая
точка –
есть
изменения в
любом
направлении

Угловые точки являются опорными в задаче слежения



- Кругами обозначены “хорошие” точки

picture from:
Gary Bradski, Adrian Kaehler, "Learning OpenCV:
Computer Vision with the OpenCV Library"

Алгоритм нахождения угловых точек (Moravec)

Вход: изображение $I(x,y)$, размер окна , порог T

Выход: координаты угловых(особых) точек

1. Для каждого пикселя (x, y) и сдвига (u, v) вычисляется

$$V_{u,v}(x,y) = \sum_{\forall a,b \text{ in the window}} (I(x+u+a, y+v+b) - I(x+a, y+b))^2$$

где сдвиг (u,v) может принимать значения: $(1,0), (1,1), (0,1), (-1,1), (-1,0), (-1,-1), (0,-1), (1,-1)$

2. Строится карта “самоподобности” пикселей $C(x, y)$:

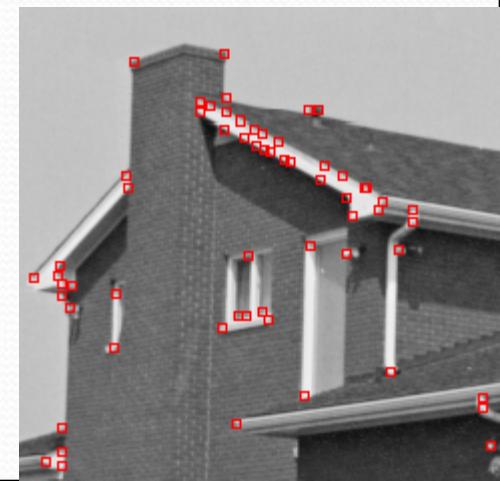
$$C(x, y) = \min(V_{u,v}(x, y))$$

3. Значения $C(x, y) < T$ обнуляются.

4. Осуществляется процедура подавления не-максимумов.

5. Оставшиеся точки считаются особыми.

	B1	B2	B3
A1	A2	A3	B6
B4	B5		
A4	A5	A6	B9
B7	B8		
A7	A8	A9	



Нахождение угловых точек (Harris)

$$S(x, y) = \sum_u \sum_v (I(u, v) - I(u + x, v + y))^2$$

$$I(u + x, v + y) \approx I(u, v) + I_x(u, v)x + I_y(u, v)y$$

$$S(x, y) \approx \sum_u \sum_v (I_x(u, v)x + I_y(u, v)y)^2 \quad S(x, y) \approx \begin{pmatrix} x & y \end{pmatrix} A \begin{pmatrix} x \\ y \end{pmatrix}$$

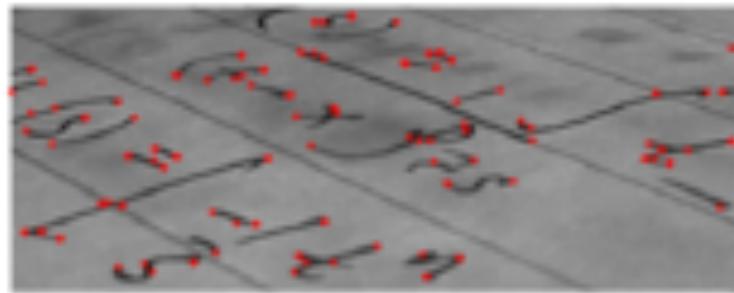
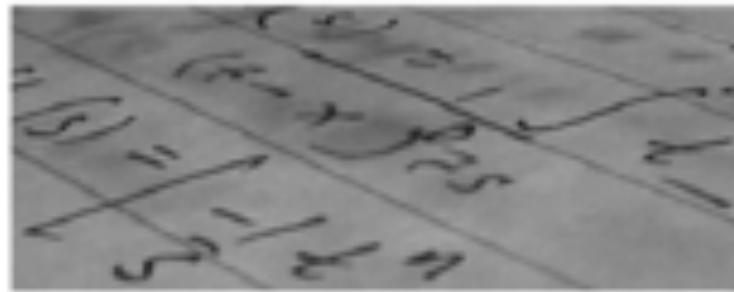
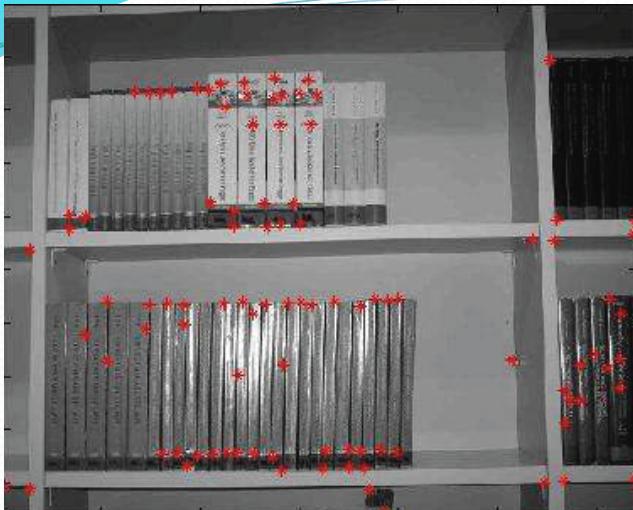
$$A = \sum_u \sum_v \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix}$$

У матрицы А вычисляются собственные числа λ_1, λ_2

1. Если $\lambda_1 \sim 0$ & $\lambda_2 \sim 0$ – однородная область.
2. Если $\lambda_1 \sim 0$ & λ_2 велико, то точка на ребре.
3. Если λ_1 велико & λ_2 велико, то точка является угловой.

Вместо подсчёта λ_i быстрее посчитать

$$M_c = \lambda_1 \lambda_2 - \kappa (\lambda_1 + \lambda_2)^2 = \det(A) - \underbrace{\kappa \text{trace}^2(A)}_{\text{0.04 - 0.15}}$$

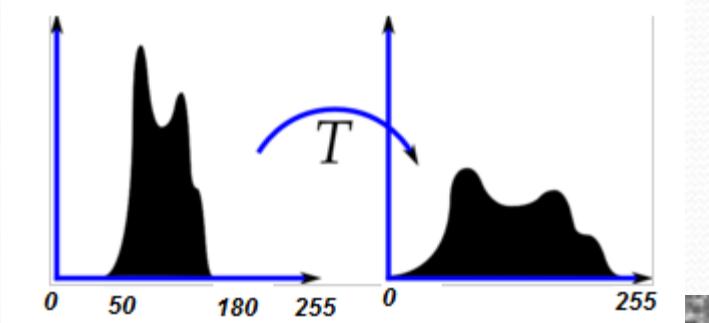


Ikdemo.exe

Улучшение изображения

- Улучшение контраста
- Подавление шума
- Просветление тёмных участков
- Улучшение цвета

Улучшение контраста



Пусть \max/\min –
максимальное/минимальное
значение интенсивности

```
k = 255/(\max - \min);  
b = -(min*255)/(\max - \min);  
for (i = 0; i < height; i++)  
{  
    for (j = 0; j < width; j++)  
    {  
        dst [i, j] = src [i,j]*k + b;  
    }  
}
```

Выравнивание контраста: Эквалайзация гистограммы

(Например , применяется в распознавании номеров)

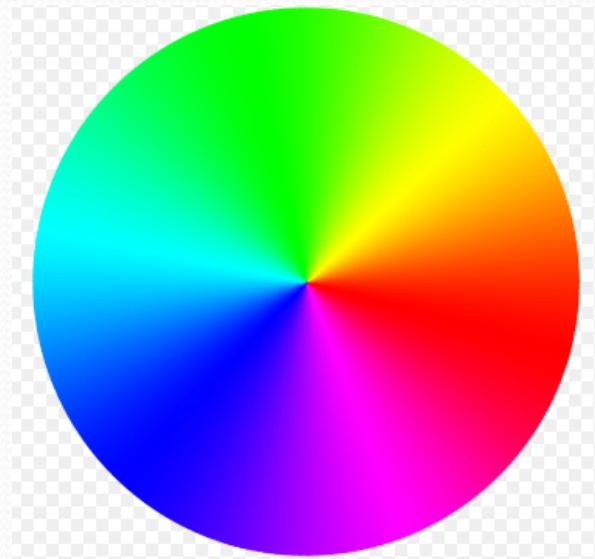
- Вычисление гистограммы H для исходного изображения
- Нормализация H так что сумма столбцов равна 255
- Вычисление интеграла гистограммы
 $H'(i) = \text{sum}_{0 \leq j \leq i} H(j)$. (интерполяционная кривая)
- Получение выходного изображения по интегралу
 $\text{dst}(x,y)=H'(\text{src}(x,y))$



Этот метод выравнивает яркость и повышает контраст

Обработка мультиспектральных изображений.

- Цветовое пространство представляет собой модель представления цвета, основанную на использовании цветовых координат. Цветовое пространство строится таким образом, чтобы любой цвет был представим точкой, имеющей определённые координаты, причём так, чтобы одному набору координат соответствовал один цвет.
- Часто каналы рассматриваются и обрабатываются независимо. (например, в задаче фильтрации)

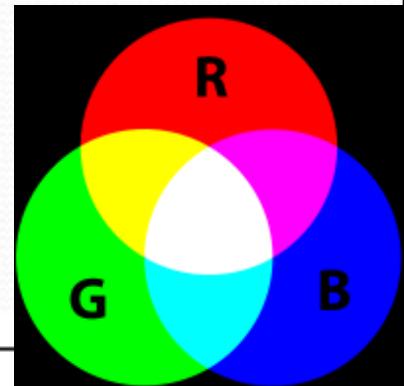
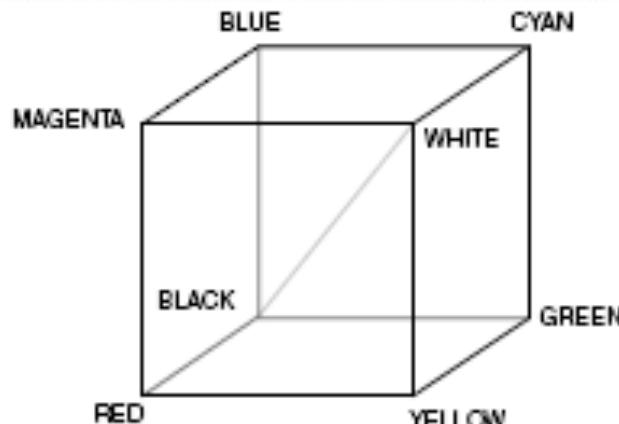


Цветные изображения.

Линейные цветовые пространства

Пространство RGB .

- Современные устройства формирования изображений обычно выдают изображение в формате RGB.
- RGB является доминантным цветовым пространством в графике. Однако RGB не всегда удобно для построения систем тех. зрения



	Nominal Range	White	Yellow	Cyan	Green	Magenta	Red	Blue	Black
R	0 to 255	255	255	0	0	255	255	0	0
G	0 to 255	255	255	255	255	0	0	0	0
B	0 to 255	255	0	255	0	255	0	255	0

- RGBA – это четырёхмерное пространство RGB с дополнительным каналом “прозрачность”

Пространство YUV

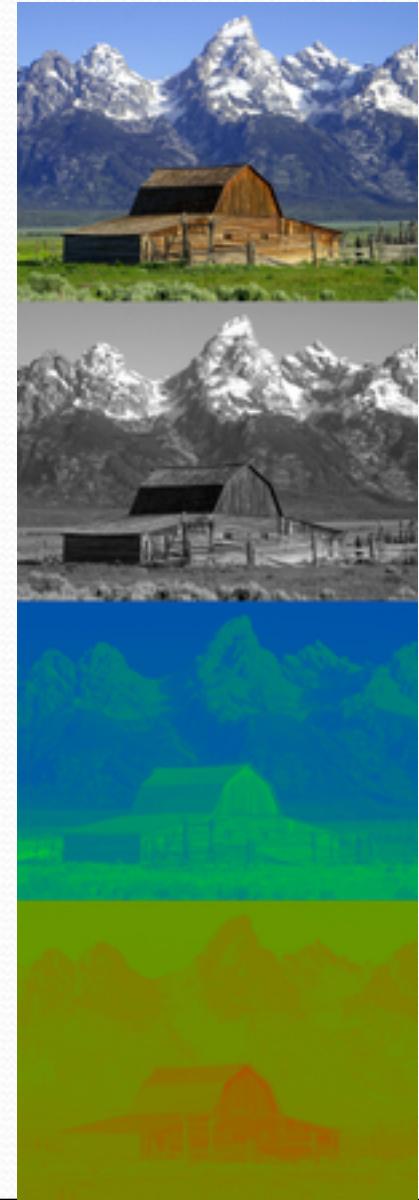
Y – яркость,
U,V – компоненты отвечающие за цвет

В телевидении и при сжатии часто применяют
схемы 4:2:2 или 4:1:1

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

$$U = -0.147 \times R - 0.289 \times G + 0.436 \times B$$

$$V = 0.615 \times R - 0.515 \times G - 0.100 \times B$$



Нелинейные цветовые пр-ва (HSV)

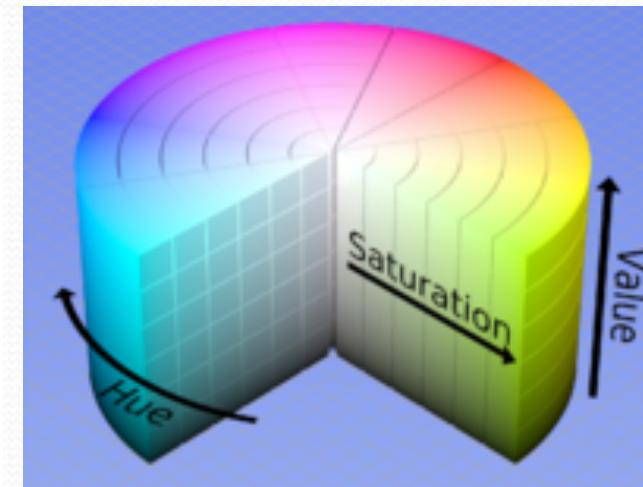
Более понятное описание цвета –
интенсивность (яркость)+ оттенок (Hue/тон)+ насыщенность (saturation).

```
V <- max (R, G, B)
S <- (V-min(R,G,B)) / V      if V≠0, 0 otherwise

(G - B) *60/S,   if V=R
H <- 180+(B - R) *60/S,   if V=G
240+(R - G) *60/S,   if V=B

if H<0 then H<-H+360

On output 0≤v≤1, 0≤s≤1, 0≤h≤360.
```



Какой канал трансформировался?

Равномерные цветовые пространства (LAB)

WiKi: Lab color is designed to approximate human vision

```
// convert R,G,B to CIE XYZ
| X |   | 0.412453  0.357580  0.180423 | | R |
| Y | <- | 0.212671  0.715160  0.072169 | *| G |
| Z |   | 0.019334  0.119193  0.950227 | | B |

X <- X/Xn, where Xn = 0.950456
Z <- Z/Zn, where Zn = 1.088754
L <- 116*Y1/3           for Y>0.008856
L <- 903.3*Y            for Y<=0.008856

a <- 500*(f(X)-f(Y)) + delta
b <- 200*(f(Y)-f(Z)) + delta
where f(t)=t1/3          for t>0.008856
      f(t)=7.787*t+16/116 for t<=0.008856
```

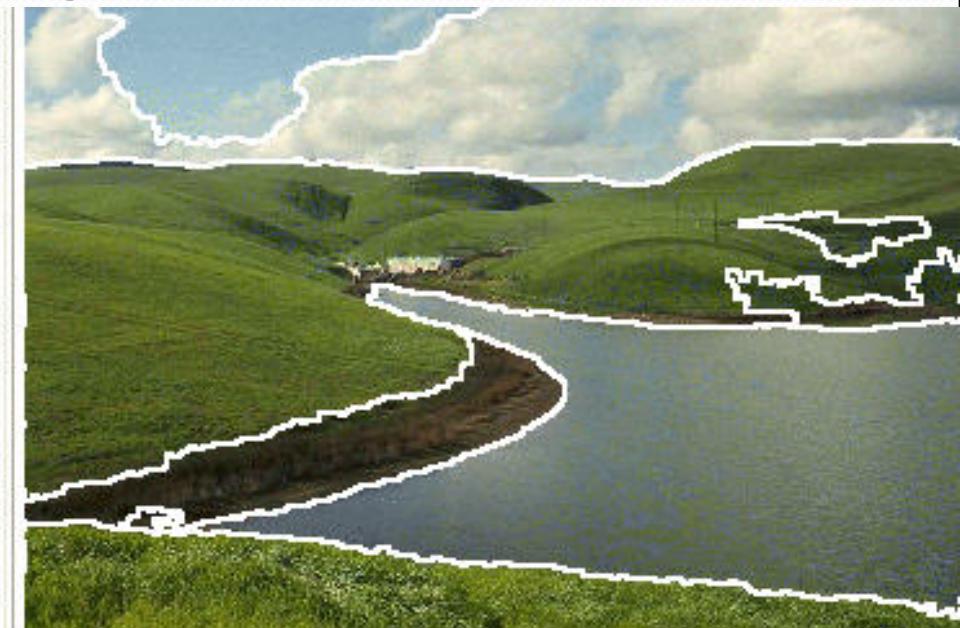
where delta = 128 for 8-bit images,

On output 0≤L≤100, -127≤a≤127, -127≤b≤127

Преобразование
нелинейно!!!

Цветовая сегментация

- В задаче поиска объектов, классификации сцены или сравнения изображений часто необходимо разбить изображение на непересекающиеся регионы, содержащие близкие по цвету пиксели.



FloodFill сегментация

Соседние пиксели принадлежат одному сегменту если:

(1)

$I(\text{seed.x}, \text{seed.y}) - \text{lo_diff} \leq I(x, y) \leq I(\text{seed.x}, \text{seed.y}) + \text{up_diff}$,

Режим фиксированного диапазона

(2)

$I(x', y') - \text{lo_diff} \leq I(x, y) \leq I(x', y') + \text{up_diff}$,

Режим плавающего диапазона

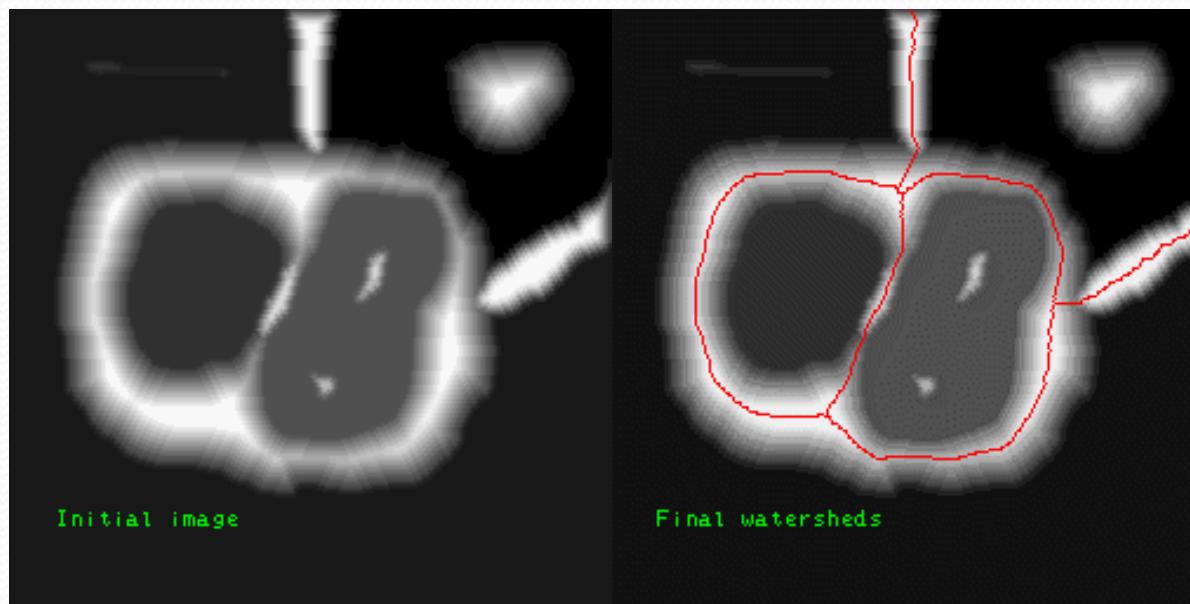


picture from:
Gary Bradski, Adrian Kaehler, "Learning OpenCV:
Computer Vision with the OpenCV Library"

ffilldemo.exe

Watershed сегментация

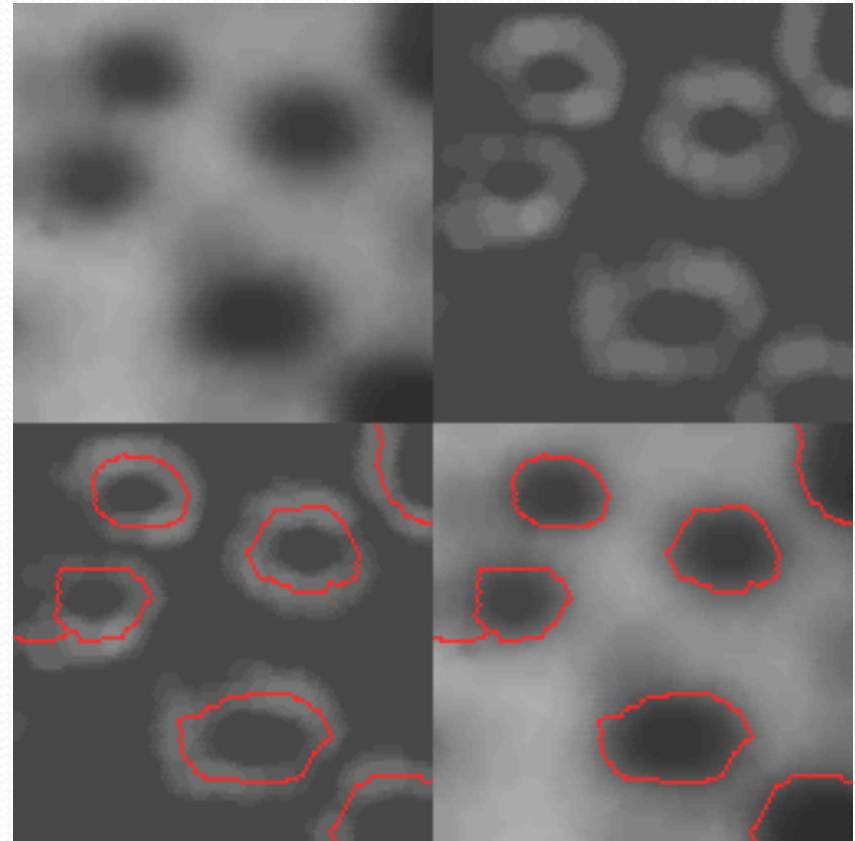
- Изображение рассматривается как 3D поверхность
- Если заливать изображение начиная с локальных минимумов и запретить при этом воды из разных источников смешиваться получим границы водоразделов.



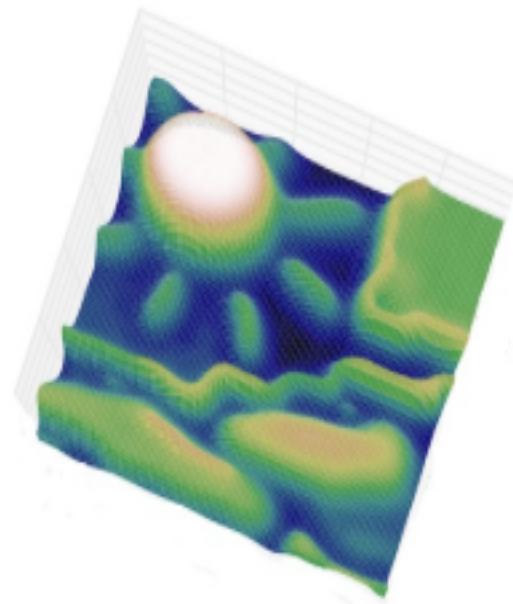
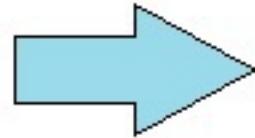
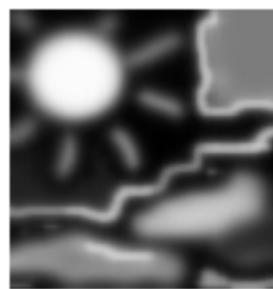
Watershed (на градиентах)

- Считается магнитуда градиентов.
- Осуществляется watershed на магнитудах градиентов.

(можно применить в медицинских приложениях)



Проблема: Много лишних сегментов



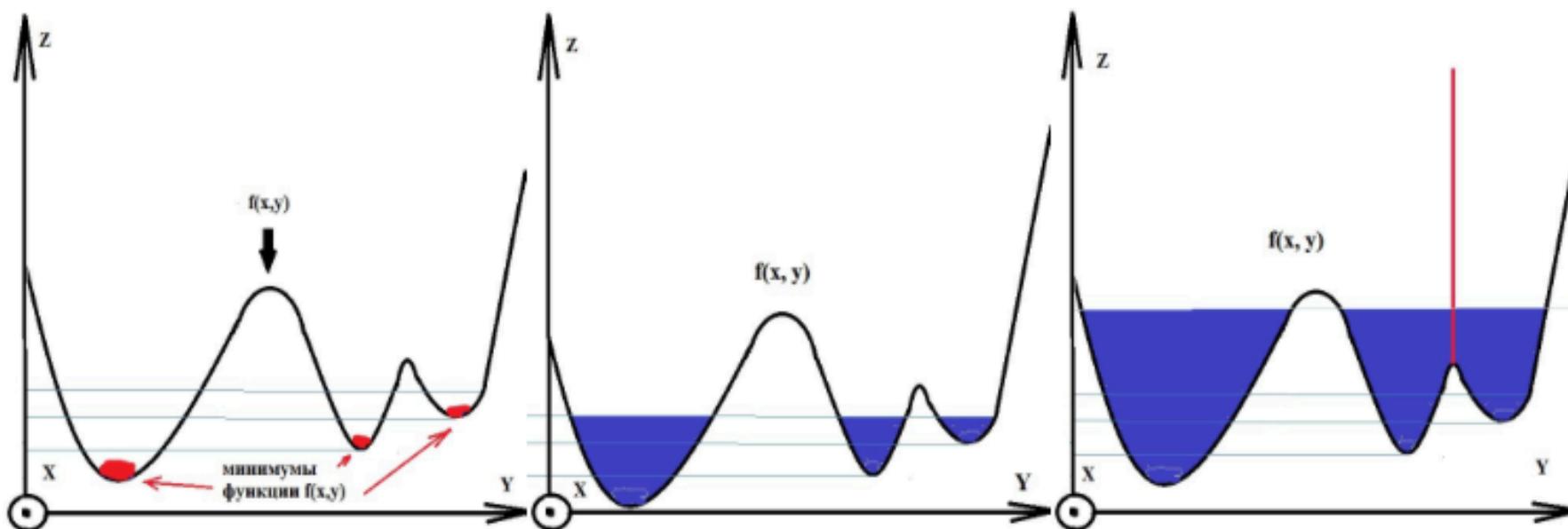
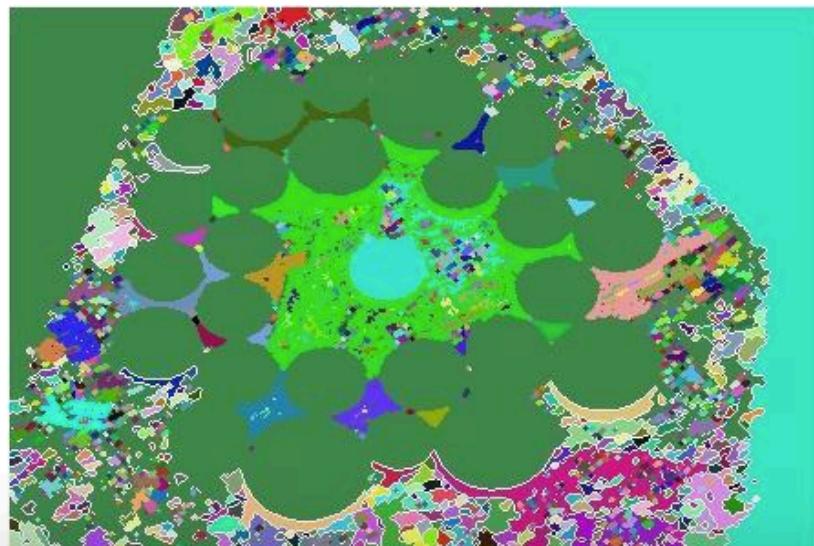


Рис.1. Иллюстрация процесса заполнения водой



Рис. 2. Исходное изображение



Watershed (использование маркеров)

- Отличие: заливка осуществляется только из маркированных точек.

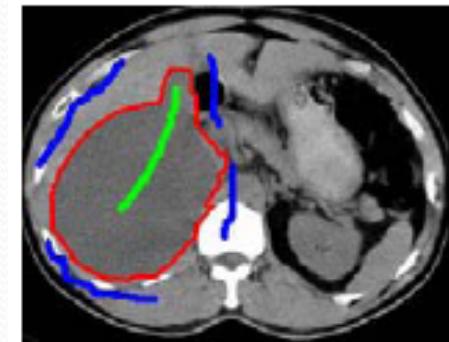
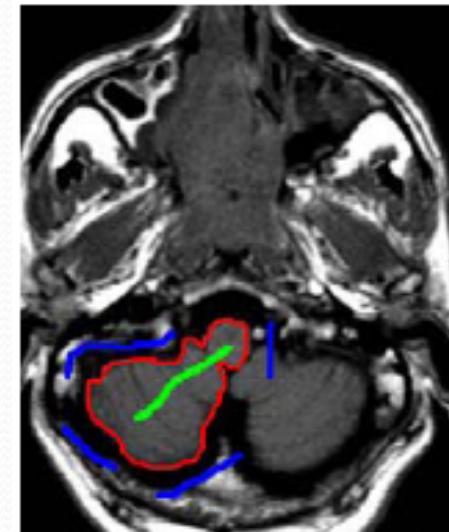
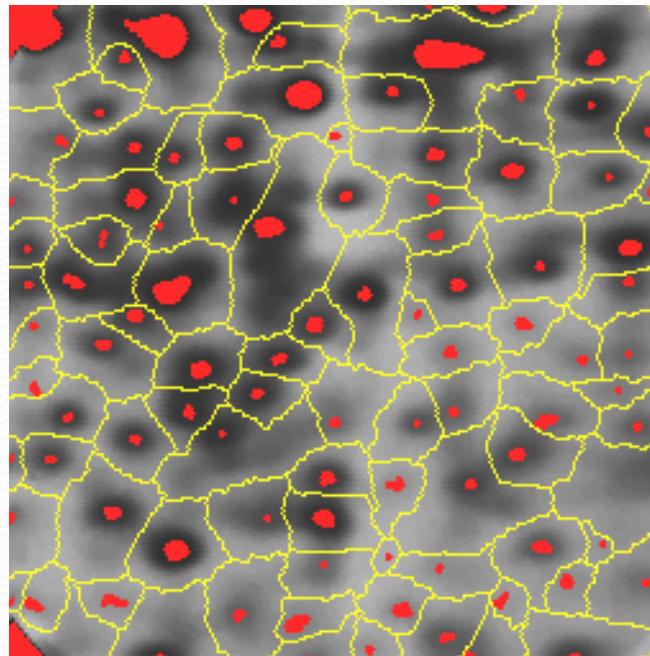




Рис. 4. Изображение с маркерами

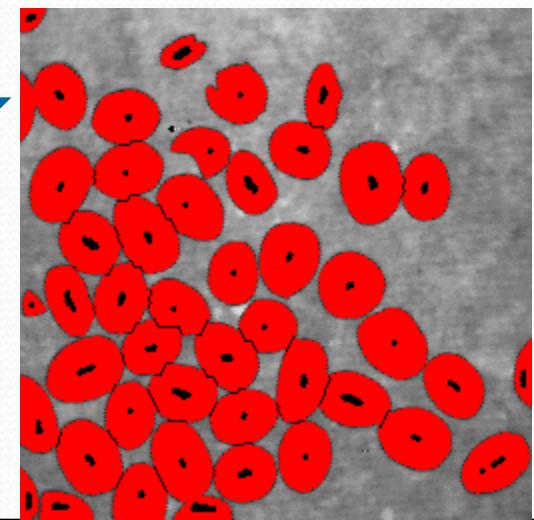
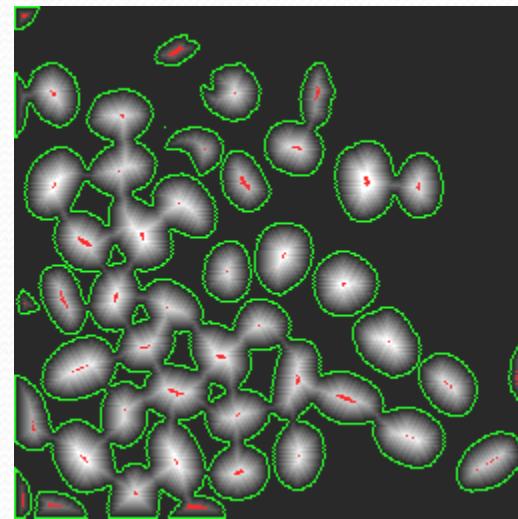
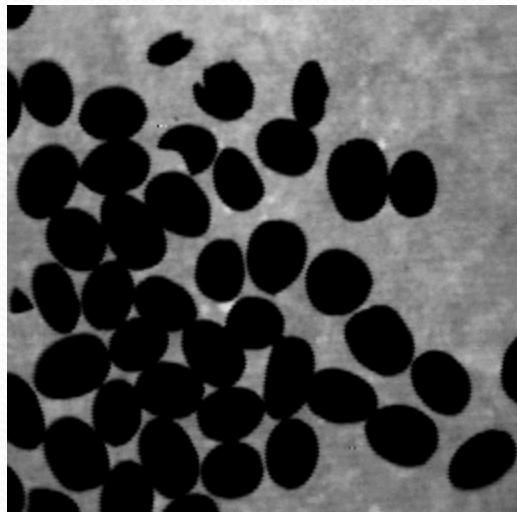


Watershed (Реализация)

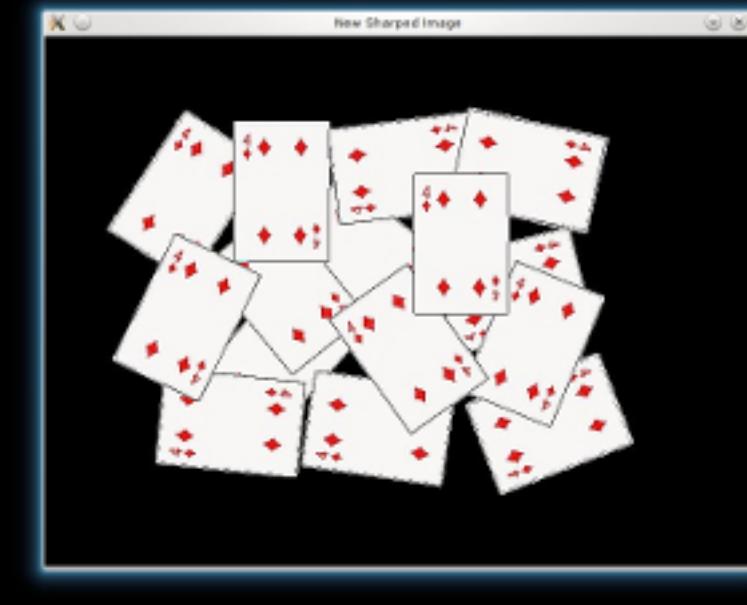
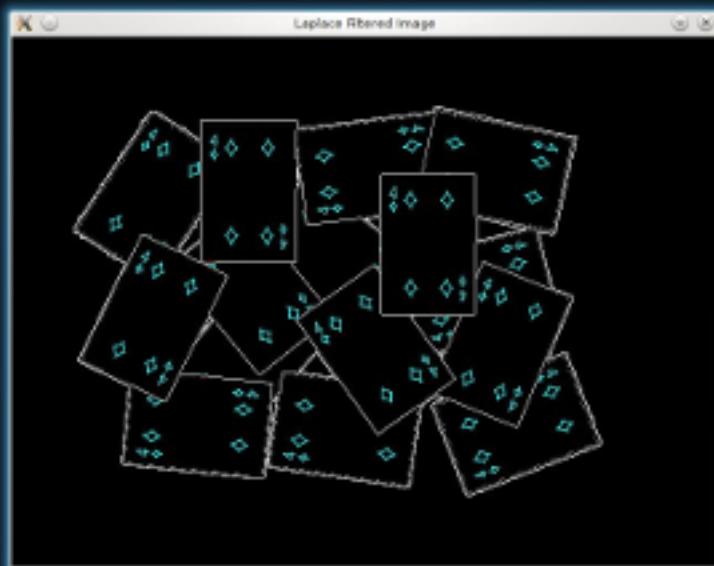
- Формируется гистограмма расстояний (цветовое отличие) от точек-маркеров до соседних немаркированных пикселей.
- Выбирается пиксель из ненулевого бина гистограммы, соответствующий наименьшему рассогласованию пикселей по цвету.
- Этому пикслю приписывается соответствующий номер сегмента.
- Гистограмма расстояний обновляется учитывая всех соседей обновлённого пикселя.

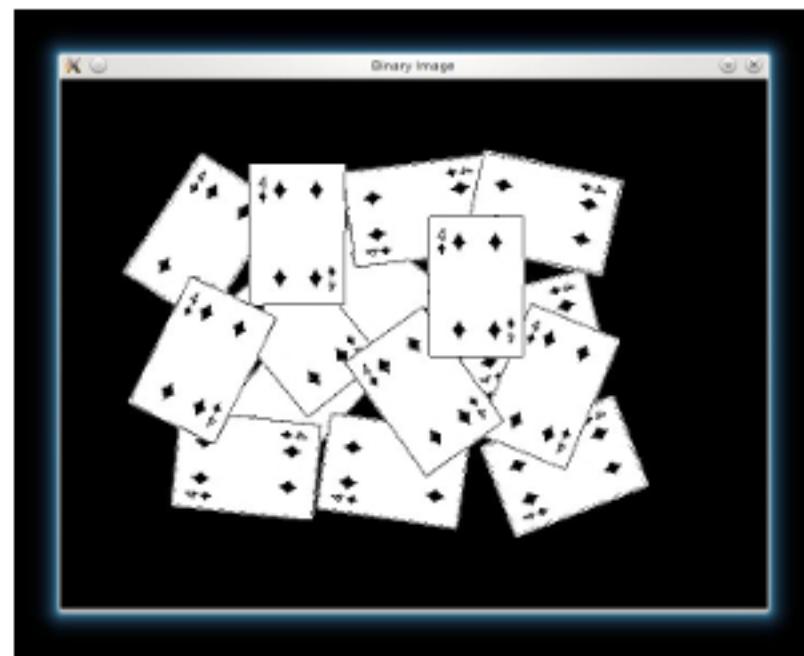
watershed.exe

Watershed (комбинация с трансформацией расстояния)

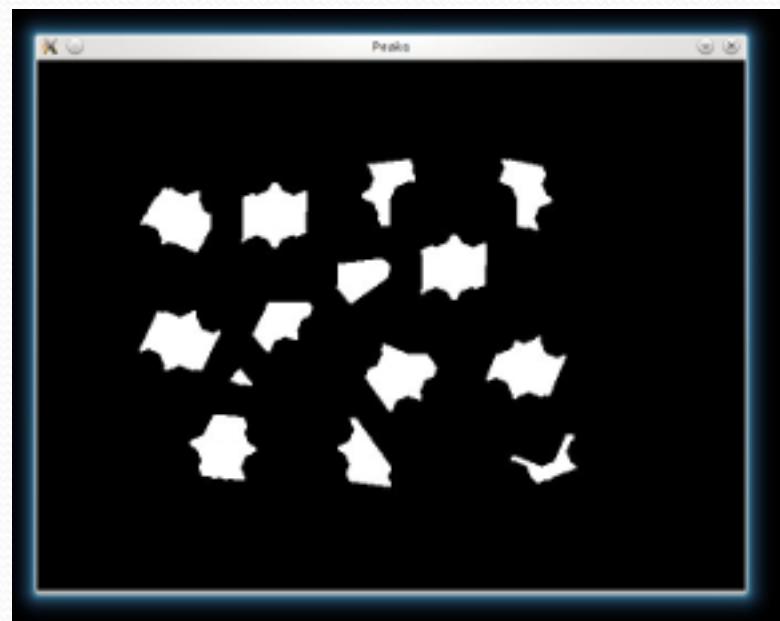


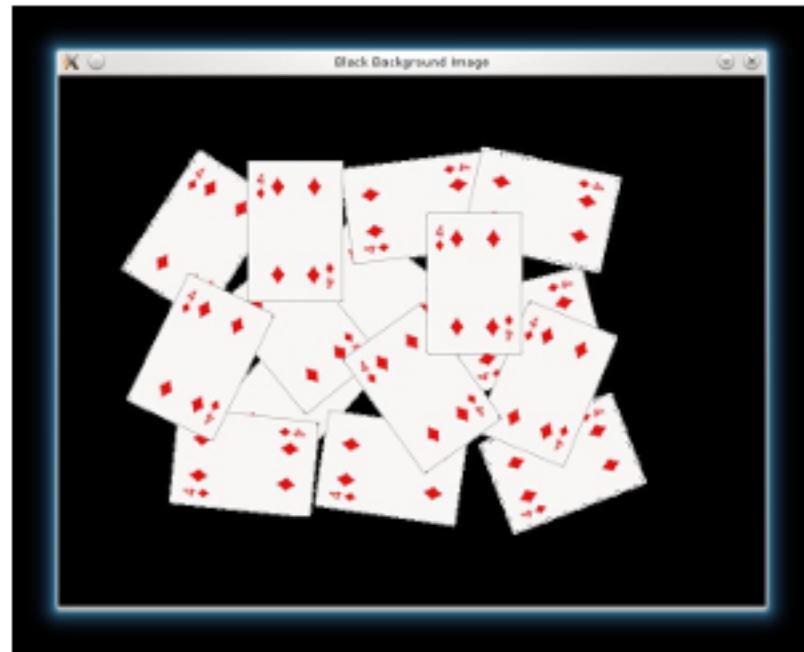
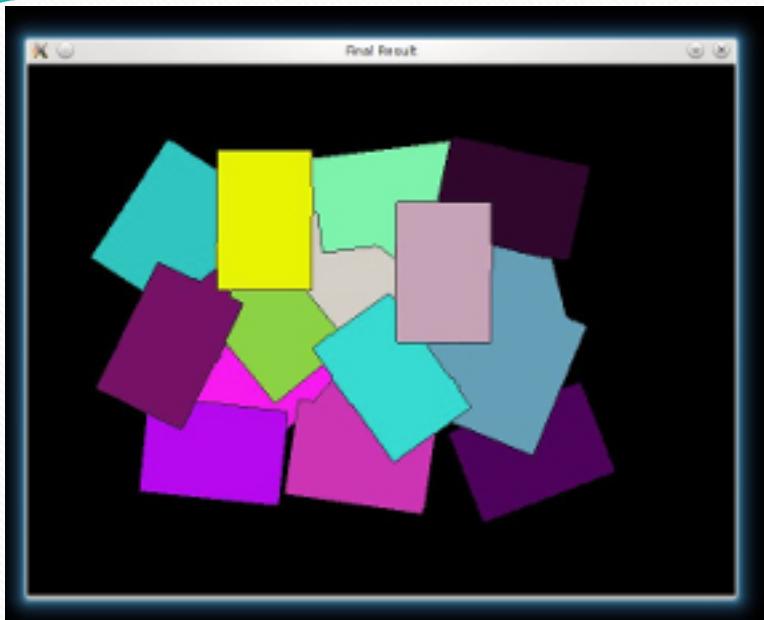








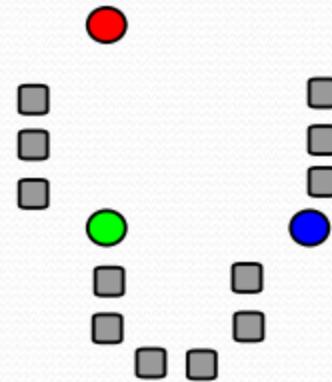




К-средних

Нужно минимизировать среднеквадратичное отклонение на точках каждого кластера:

$$V = \sum_{i=1}^k \sum_{x_j \in S_i} (x_j - \mu_i)^2$$



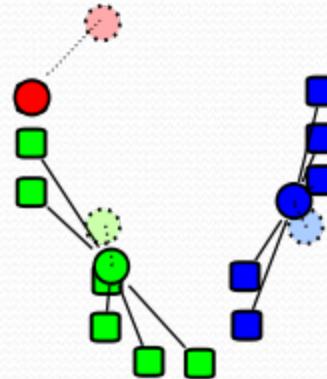
Исходные точки и случайно выбранные начальные точки.

где

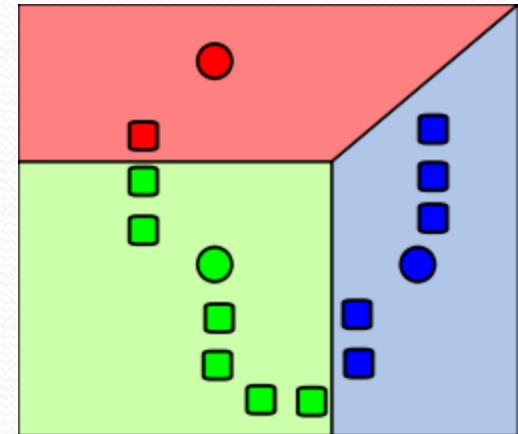
k - число кластеров,

S_i - полученные кластеры,

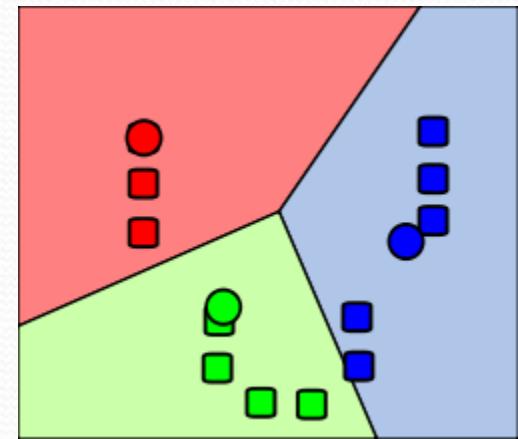
μ_i - центры масс векторов.



Вычисление новых центров кластеров.



Точки, отнесённые к начальным центрам.



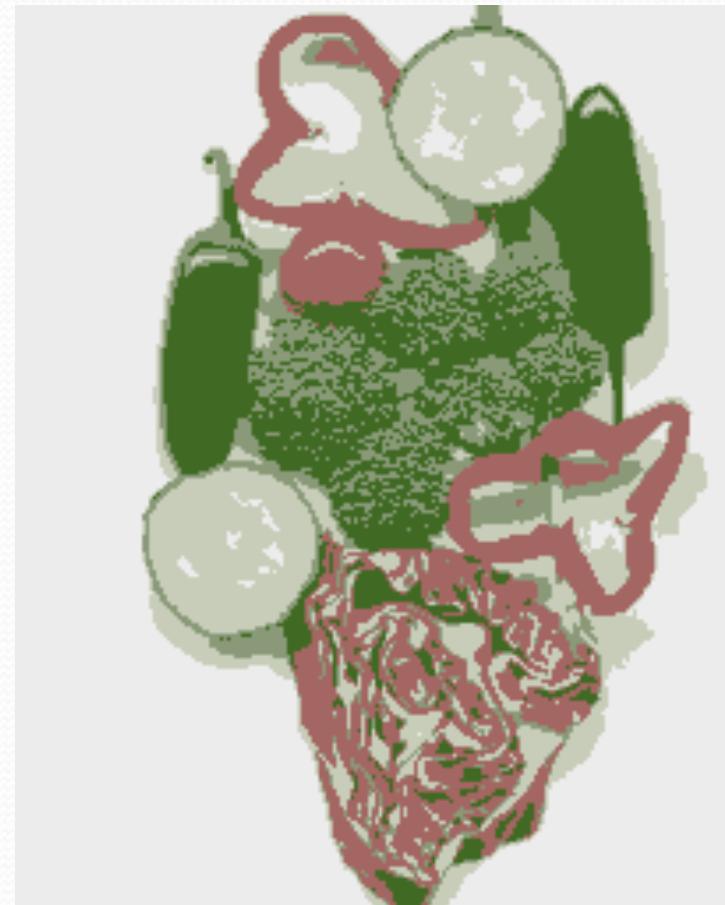
Предыдущие шаги повторяются, пока алгоритм не сойдётся.

Сегментация К-средних

- Выбираем ожидаемое число цветных объектов N .
- Все пиксели рассматриваем как точки в RGB пространстве.
- Среди этих точек ищутся N кластеров методом К-средних.
- Пиксели изображения сравниваются с найденными центрами кластеров и принадлежат сегменту ассоциированному с ближайшему по цвету кластеру.

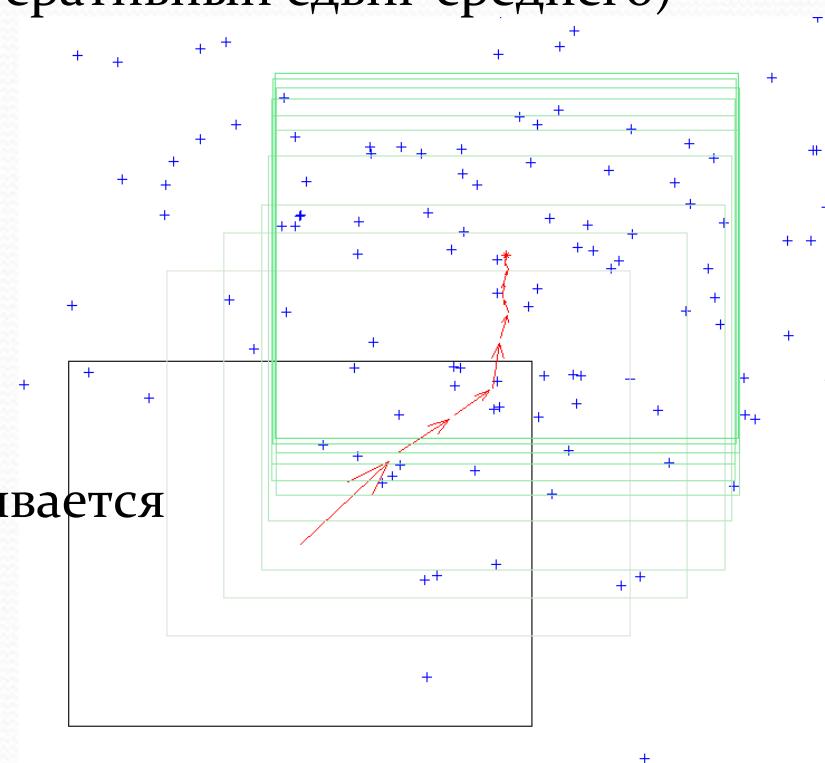
Эту сегментацию лучше делать в LAB пространстве.

Пример: k-means K = 2



Сегментация Mean_shift

- Шаг 1. Фильтрация (используется итеративный сдвиг среднего)
 - Для каждого пикселя в окрестности выбираются похожие по цвету пиксели.
 - Вычисляется их геом. центр.
 - Центр окна переносится в новый геом. центр.
 - Операция повторяется.
 - Цвет исходного пиксела приравнивается к цвету полученного геом. центра.
- Шаг 2. Все пиксели, которые по цвету сошлись в одну точку принадлежат одному сегменту (суперпиксель).
- Шаг 3. Соседние суперпиксели объединяются в один больший сегмент если их цвета близки.





<http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html>

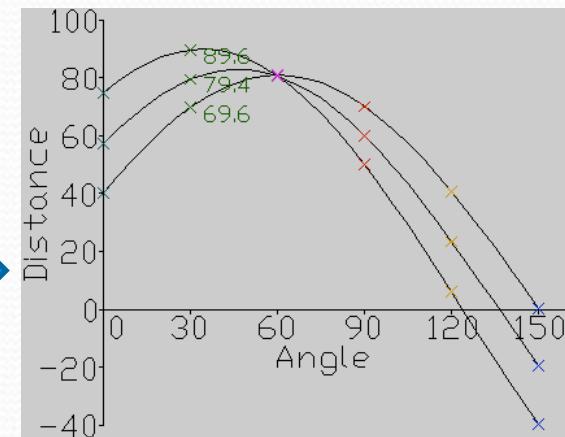
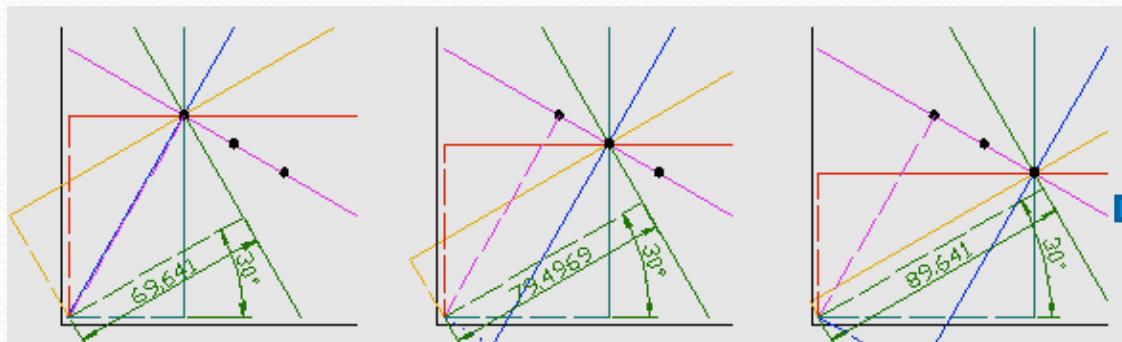
Поиск примитивов.

Преобразование Хафа

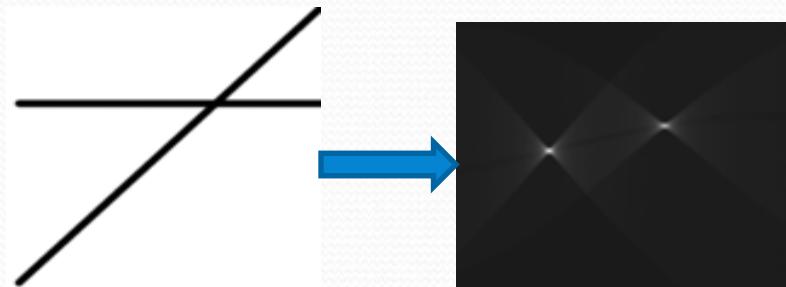
- Задача - найти линии на изображении формируемые точками бинарного изображения

$$y = \left(-\frac{\cos \theta}{\sin \theta} \right) x + \left(\frac{r}{\sin \theta} \right)$$

- Каждая точка бинарного изображения формирует множество гипотез о том какие линии через неё могут проходить.



- Все гипотезы аккумулируются.
- Выбираются локальные максимумы в пространстве гипотез.



Применение: 3D реконструкция, поиск полосы дороги, ...

ПОИСК ЛИНИЙ

```
procedure accumulate_lines(S,A);
{
A := 0;
PTLIST := NIL;
for R := 1 to NLINES
  for C := 1 to NPIXELS
    {
      DR := row_gradient(S,R,C);
      DC := col_gradient(S,R,C);
      GMAG := gradient(DR,DC);
      if GMAG > gradient_threshold
        {
          THETA := atan2(DR,DC);
          THETAQ := quantize_angle(THETA);
          D := abs(C*cos(THETAQ) - R*sin(THETAQ));
          DQ := quantize_distance(D);
          A[DQ,THETAQ] := A[DQ,THETAQ]+GMAG;
          PTLIST(DQ,THETAQ) := append(PTLIST(DQ,THETAQ),[R,C])
        }
    }
}
```

S[R, C] is the input gray-tone image.

NLINES is the number of rows in the image.

NPIXELS is the number of pixels per row.

A[DQ, THETAQ] is the accumulator array.

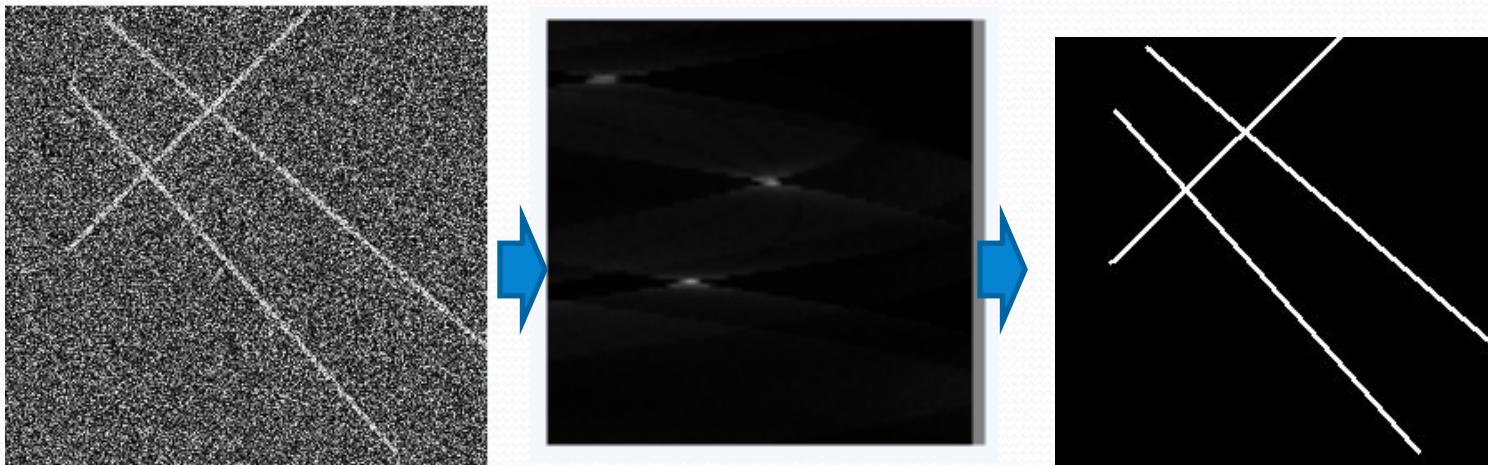
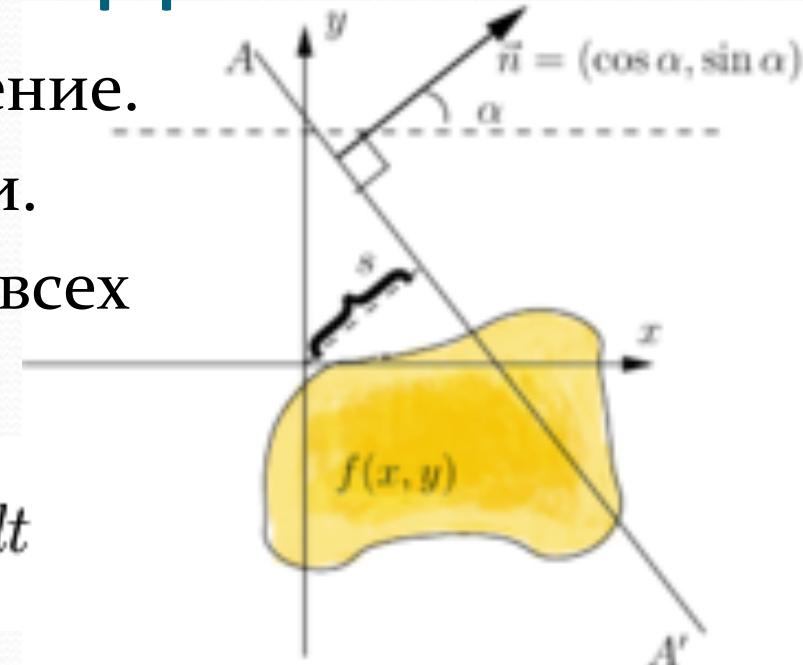
DQ is the quantized distance from a line to the origin.

THETAQ is the quantized angle of the normal to the line.

Преобразование Радона

- Вход – полуточновое изображение.
- Задача - найти светлые линии.
- Вычисляется интеграл вдоль всех прямых

$$\mathcal{R}[f](\alpha, s) = \int_{-\infty}^{\infty} f(x(t), y(t)) dt$$



Частотный анализ изображений

- Преобразование Фурье. Разложение изображения по гармоникам.

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j 2\pi(ux+vy)} dx dy$$

- Дискретный вариант

$$F[u, v] \equiv \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} I[x, y] e^{-\frac{j 2\pi}{N}(xu+vy)}$$

$$I[x, y] \equiv \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F[u, v] e^{\frac{j 2\pi}{N}(ux+vy)}$$

- $F(0,0)$?

Спектр и фаза

- Спектр

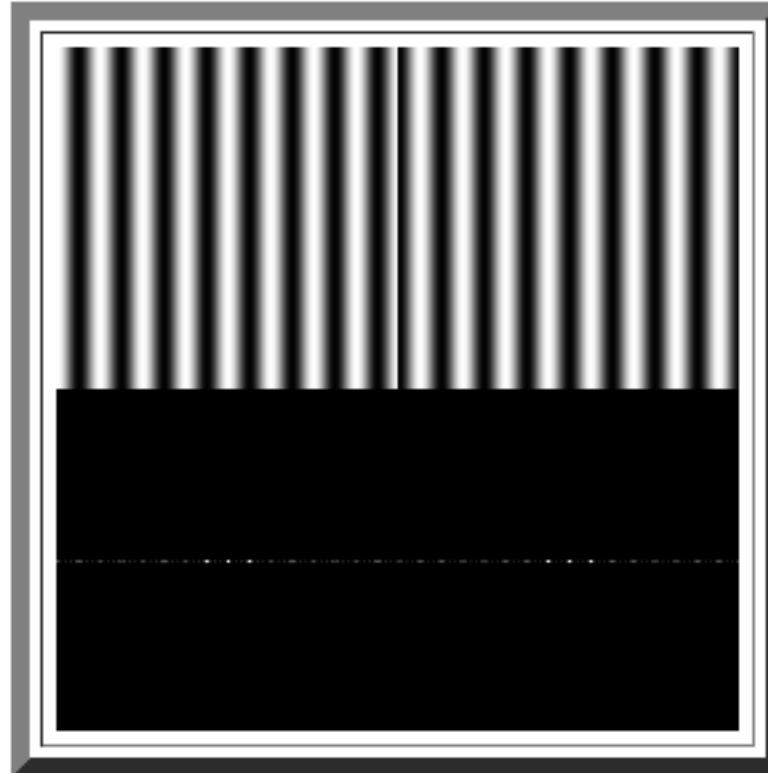
$$e^{-j \cdot 2\pi(ux+vy)} = \cos(2\pi(ux + vy)) - j\sin(2\pi(ux + vy))$$

$$P(u, v) \equiv (\operatorname{Real}(F(u, v))^2 + \operatorname{Imaginary}(F(u, v))^2)^{1/2}$$

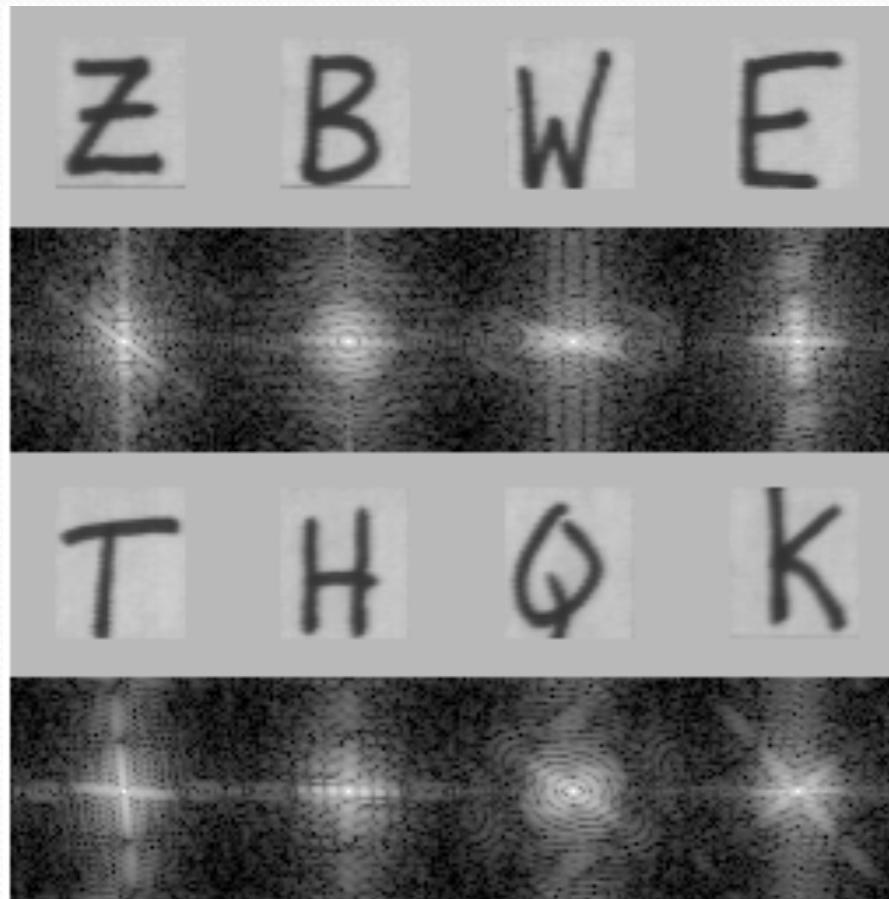
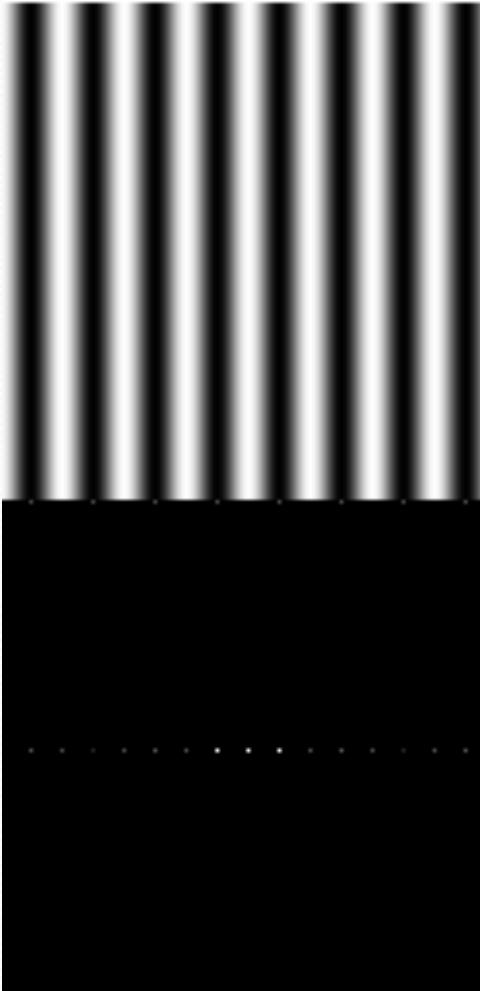
$$P(-u, -v) = P(u, v)$$

- Фаза

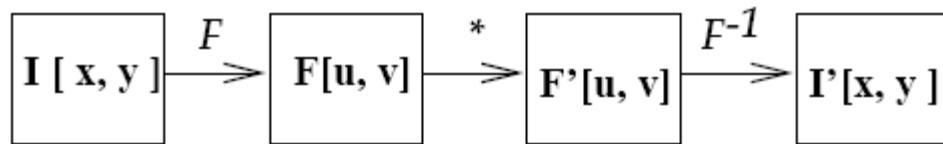
$$\operatorname{PHASE}(F) = \operatorname{ATAN}(\operatorname{IMAGINARY}(F)/\operatorname{REAL}(F))$$



Примеры спектра



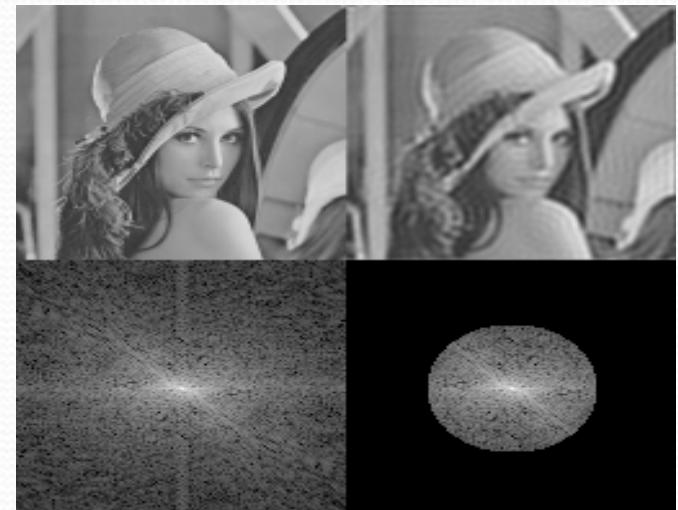
Полосно-пропускающий фильтр



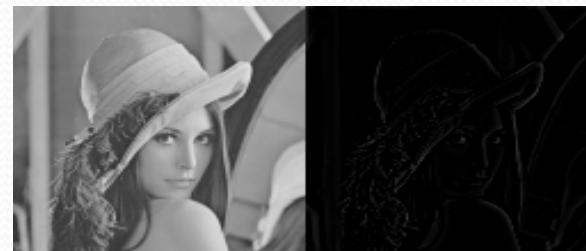
- Подавление высоких частот

$$G(k, l) = F(k, l)H(k, l)$$

$$H(k, l) = \begin{cases} 1 & \text{if } \sqrt{k^2 + l^2} < D_0 \\ 0 & \text{if } \sqrt{k^2 + l^2} > D_0 \end{cases}$$



- Подавление низких частот:



Теорема о свёртке

- Преобразование Фурье над двумя свернутыми функциями пропорционально результатам преобразований Фурье над каждой функцией в отдельности

$$\mathbf{F}(f(x,y) \star h(x,y)) \stackrel{\sim}{=} \mathbf{F}(f(x,y))\mathbf{F}(h(x,y)) = \mathbf{F}(u,v)\mathbf{H}(u,v)$$

- Алгоритм фильтрации изображения $f(x,y)$ фильтром $h(x,y)$
 1. Получить преобразования Фурье $F(u,v), H(u,v)$
 2. Поточечно перемножить $F(u,v)$ и $H(u,v)$ получить $F'(x,y)$
 3. Применить обратное преобразование Фурье для получения сглаженного изображения.

Применение преобразования Фурье

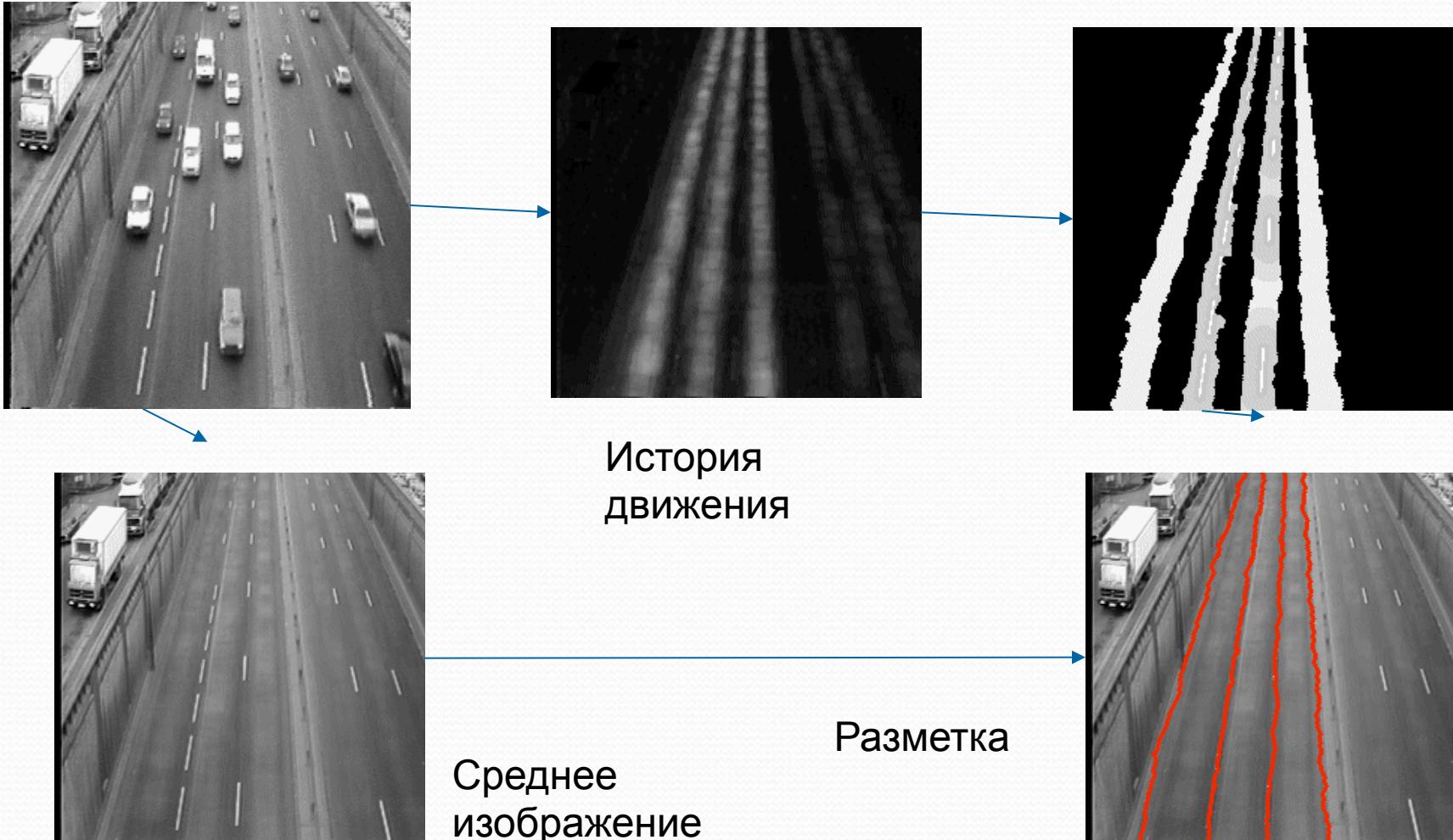
- Полосная фильтрация
- Сжатие изображения
- Описание текстуры
- Быстрый поиск объекта на изображении
(используя теорему о свёртке)

Квинтэссенция изученного

- Камера Обскура. Перспективная проекция
 $x = X^*f/Z$, $y = Y^*f/Z$
- Бинарные, полуточновые, цветные изображения
- Связанные компоненты их свойства. Морфология.
- Линейная фильтрация. Свёртка.
- Подсчёт градиентов. Фильтры Собеля.
- Края объектов (ребра) - экстремальные точки первой производной
- Угловые точки - точки без “самоподобности”.
- Улучшение контрастности. Эквализация гистограммы.

- Виды цветовых пространств – RGB, YUV, HSV
- Сегментация изображений
- Метод к-средних
- Метод “сдвиг среднего”
- Анализ в частотной области.
Преобразование Фурье.
- Библиотека OPENCV: cxcore, cv, highgui.

Watershed (пример: выделение полос)



Морфологические операции (gray and color)



- Эрозия:

$$(I \ominus H)(u, v) = \min_{(i,j) \in H} \{I(u + i, v + j) + H(i, j)\}$$

- Наращивание:

$$(I \oplus H)(u, v) = \max_{(i,j) \in H} \{I(u + i, v + j) + H(i, j)\}$$

- Размыкание:

$$I_2 = \text{open}(I, H) = \text{dilate}(\text{erode}(I, H), H)$$



- Замыкание:

$$I_2 = \text{close}(I, H) = \text{erode}(\text{dilate}(I, H), H)$$