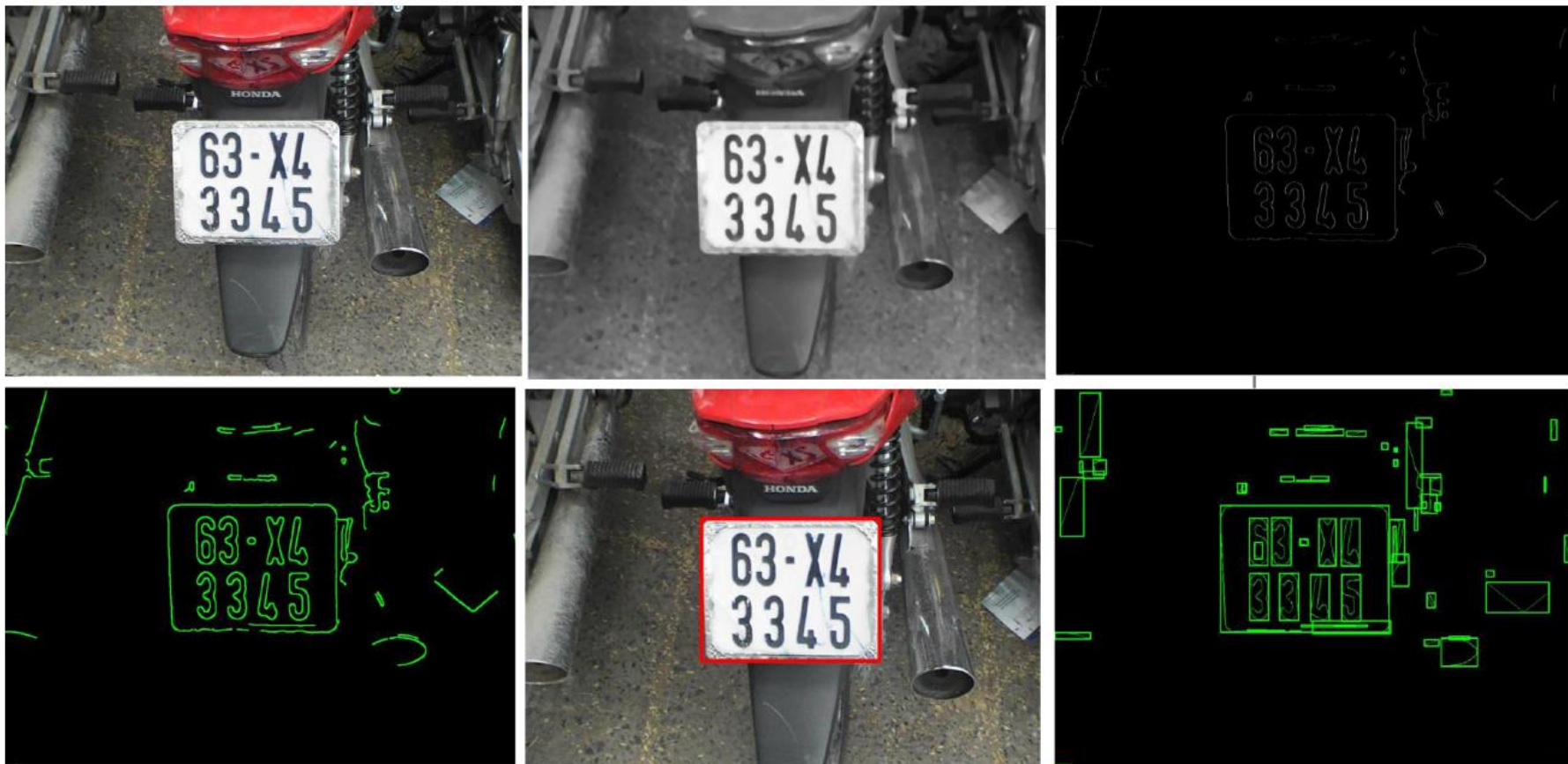


Выделение контуров объектов

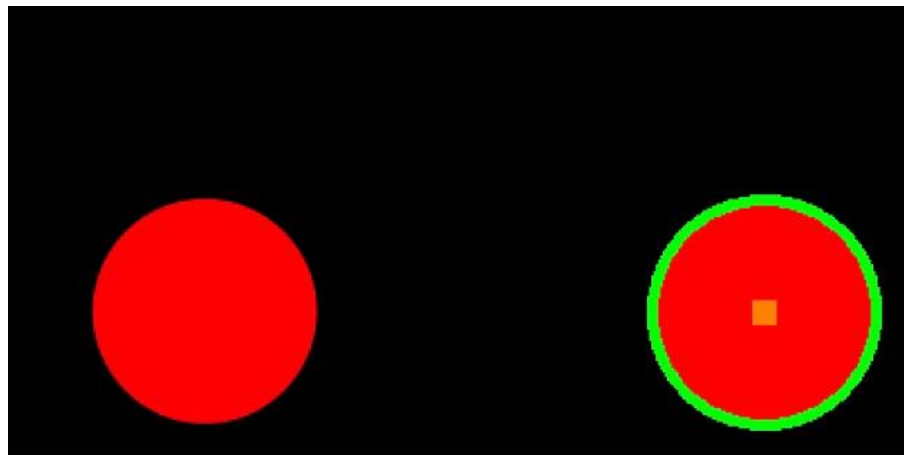
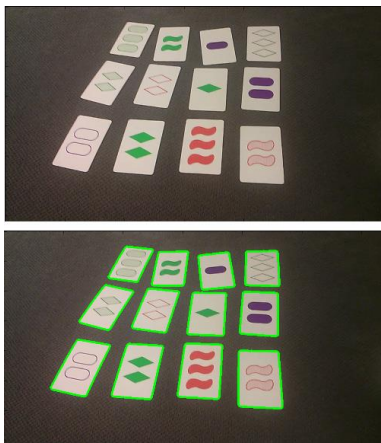


Алгоритм Кэнни (Canny Edge detector) был разработан в 1986 году Джоном Кэнни, и остается до сих пор одним из лучших методов обнаружения границ изображений.

Задача 6-1. Написать программу, выделяющую границы объектов методом Кэнни

Реализовать выделение контуров объектов по изображению границ, полученному методом Кэнни для произвольного изображения, а также их визуализацию поверх исходного изображения.

- Детектор границ Кэнни (cv::Canny)
- Обнаружение и отрисовка контуров (cv:: findContours, cv::drawContours):



Задачи 6-2, 6-3. Написать программу, которая автоматически определяет для одного объекта, выделенного методом Кэнни:

6-2. Размер объекта (маленький, средний, большой) относительно общего размера изображения

6-3. Форму объекта (квадрат, прямоугольник, треугольник, круг, овал)

Фильтрация изображений

Удаление шума
noisy lena



median filter



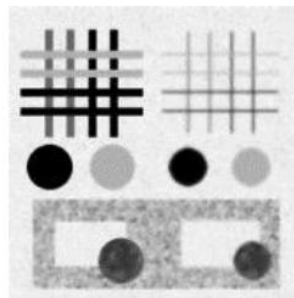
Gaussian filter



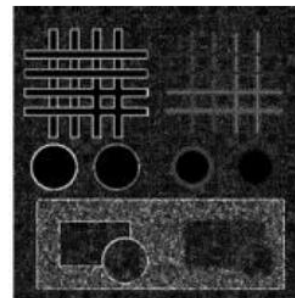
Wiener filter



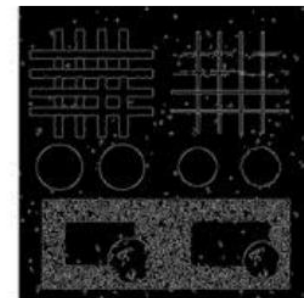
Выделение границ



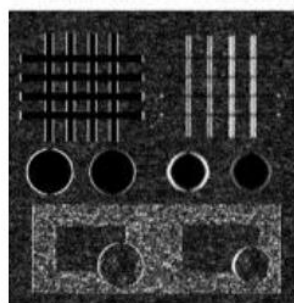
Original



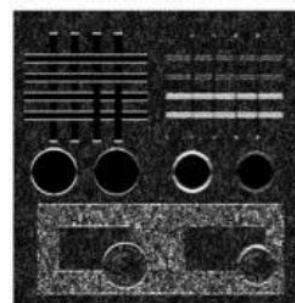
Laplacian



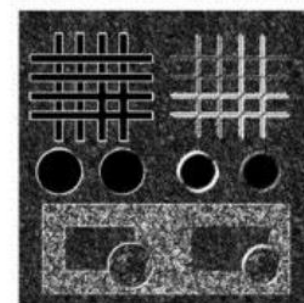
Canny



Sobel X



Sobel Y



Sobel X+Y

Задача 7. Продемонстрировать возможности линейной фильтрации изображений на основе: Гауссова фильтра, медианного, фильтра с произвольным ядром, оператора Собеля, фильтра Лапласа. Продемонстрировать удаление равномерного шума и Гауссова.

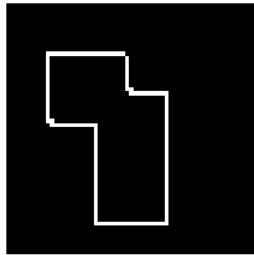
- Гауссов(cv::blur) и медианный фильтр (cv::medianBlur).
- Фильтрация произвольной маской (cv::filter2D).
- Выделение границ объектов (cv::Sobel).
- Фильтр Лапласа. Исследовать влияние входных параметров на результат фильтрации, провести сравнение с фильтром Собеля.
- cv::randn – нормальное распределение, cv::randu – равномерное распределение.

Извлечение элементов из изображений, преобразование Хафа

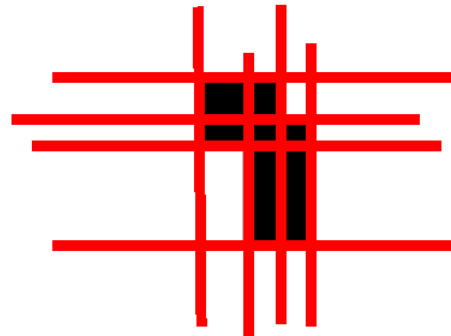
Преобразование Хафа



Объект

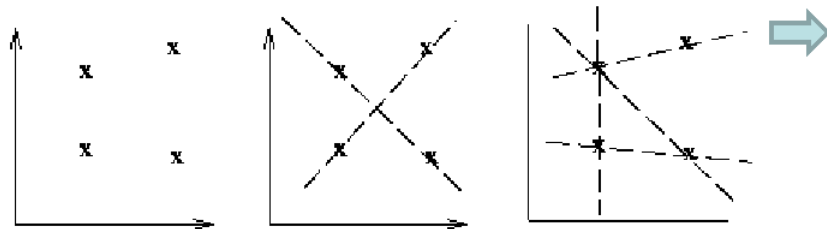


Выделение границ,
Бинаризация

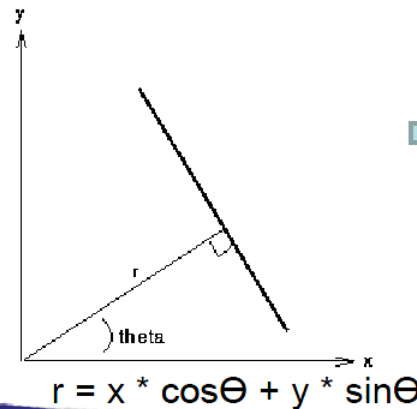


Извлечение линий

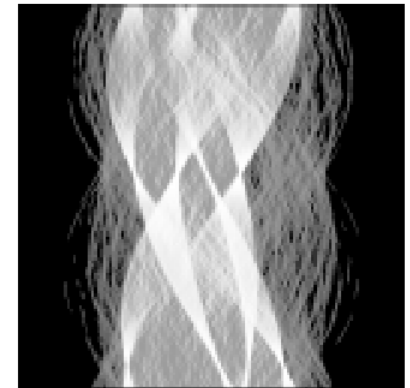
Пространство XY
 $ax + by + c = 0$



Параметризация. Переход в
пространство Хафа



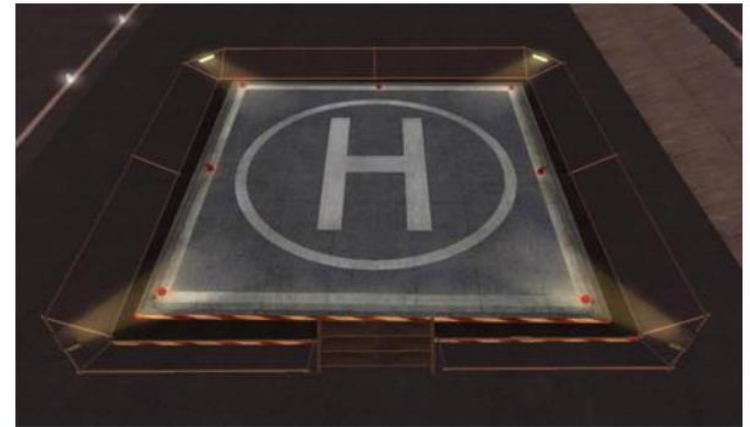
Пространство $r\theta$.
Множеству прямых
проходящих через точку
 x_0-y_0 , соответствует
синусоида



Преобразование Хафа (cv::HoughLines, cv::HoughLinesP)



Преобразование Хафа для кругов (`cv::HoughCircles`)



Задача 8. Реализовать поиск линий либо окружностей (на выбор) на характерных изображениях, исследовать эффективность при различных параметрах.

Сопоставление изображений и локальные особенности

1. Построение панорам



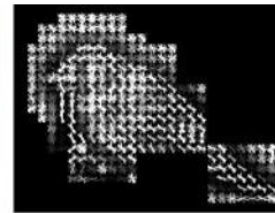
2. Совмещения спутниковых снимков, повышение разрешения



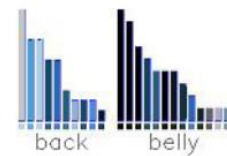
3. Обнаружение и распознавание объектов



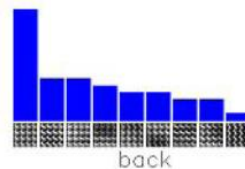
Image



HOG

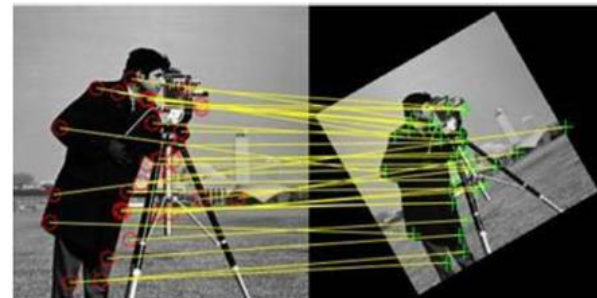
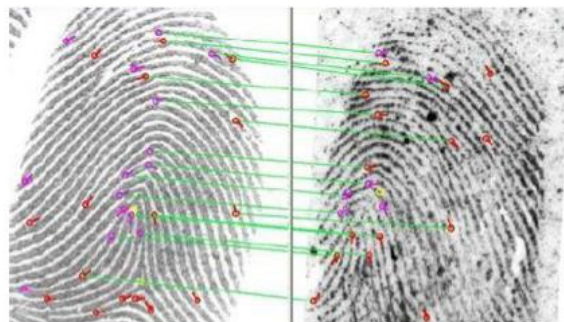
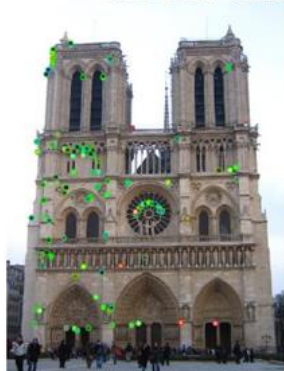


Color Histograms

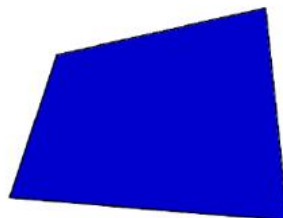
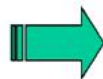


Bag of Words

Сопоставление по локальным особенностям (feature matching)



Зная соответствие
характерных точек двух
изображений можно
определить
пространственное
преобразование –
преобразование гомографии

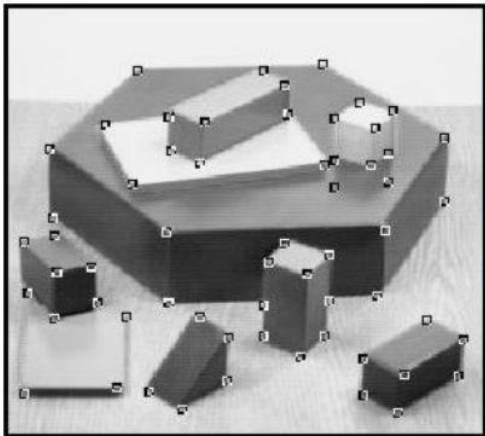


$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Задача сопоставления изображений по
локальным точкам:

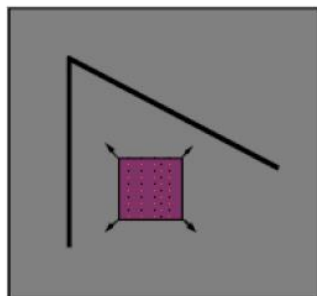
- выделение особых точек;
- формирование векторов признаков
точек;
- сопоставление точек в пространстве
признаков;

Локальные особенности (features)

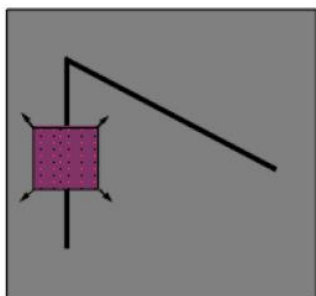


- *Повторяемость* – особая точка находится в одном и том же месте сцены или объекта изображения, несмотря на изменения точки обзора и освещённости.
- *Отличительность / информативность* – окрестности особых точек должны иметь большие отличия друг от друга, так, чтобы возможно было выделить и сопоставить особые точки.
- *Локальность* – особая точка должна занимать небольшую область изображения, чтобы уменьшить вероятность чувствительности к геометрическим и фотометрическим искажениям между двумя изображениями, снятых в различных точках обзора.
- *Компактность* – число обнаруженных особых точек должно быть достаточно большим, однако при этом их число значительно меньше точек изображения.
- *Точность* – обнаруженные особые точки должны точно локализовываться, как в исходном изображении, так и взятом в другом масштабе.
- *Эффективность (efficiency)* – время обнаружения особых точек на изображении должно быть допустимым в критичных по времени приложениях.

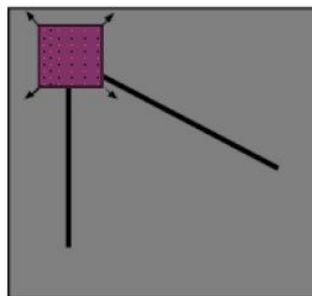
Сопоставление изображений на основе локальных особенностей



Монотонный
регион (blob) – в
любом
направлении
малые изменения



Грань(edge) –
вдоль края
изменения
минимальны

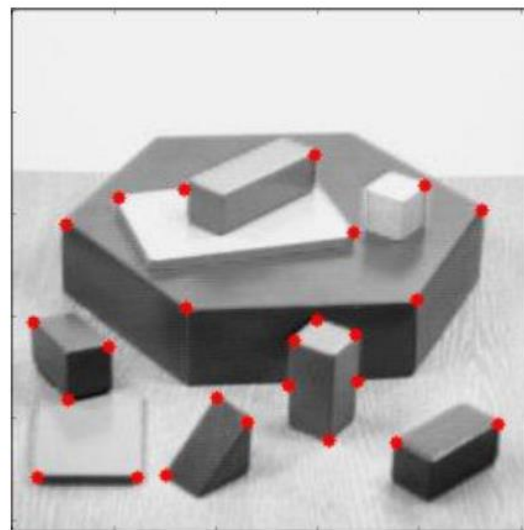


Угол (corner) –
пересечение
нескольких
граней

Углы



В окрестности угла у градиента два
доминирующих направления



Задача 9. Использовать детектор углов Харриса и детектор особых точек Ши-Томаси, исследовать влияние преобразований изображений (аффинные и перспективные преобразования, изменение яркости, контраста), а также параметров алгоритмов на результаты обнаружения локальных особенностей

- 1. Harris corner detector
(cv::cornerHarris),
- 2. Shi-Tomachi
(cv::goodFeaturesToTrack)
- 2. Преобразования
изображений
(cv::warpAffine,
cv::warpPerspective)



Задача 10. Реализовать поиск соответствия особых точек на двух изображений (feature matching), исследовать работу в реальных условиях.



?

Сегментация изображений

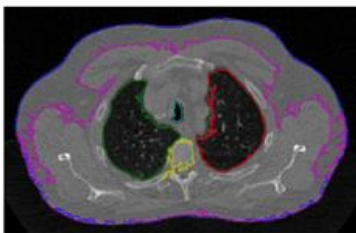
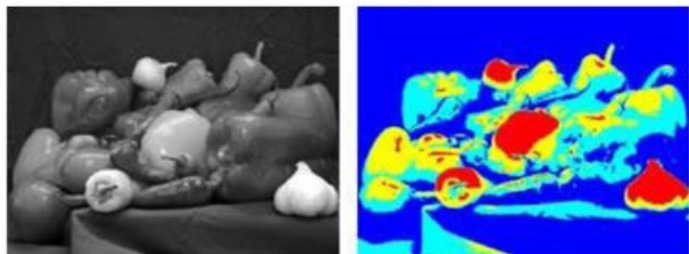
В компьютерном зрении, сегментация — это процесс разделения изображения на несколько сегментов (множество пикселей). Все пиксели в сегменте похожи по некоторой характеристике или вычисленному свойству, например, по цвету, яркости или текстуре.

Методы:

1. Пороговое преобразование
 2. Выделение границ
 3. Методы, основанные на кластеризации (например, k-means)
 4. Методы разрастания областей
 5. Метод водораздела (watershed)
 6. Метод разреза графа (graph-cut)
 7. Методы с использованием модели (active contours, deformable shapes, etc.)
 8. Машинное обучение
- ...

Сегментация изображений

1. Multi-level thresholding

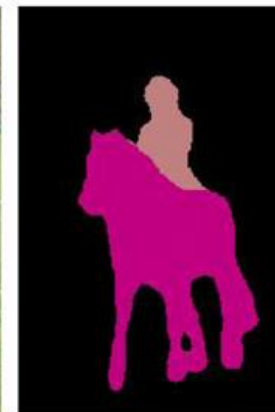
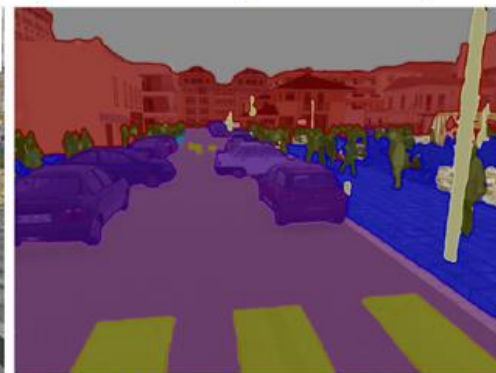


2. Использование критерия связанности точек в сегменте.

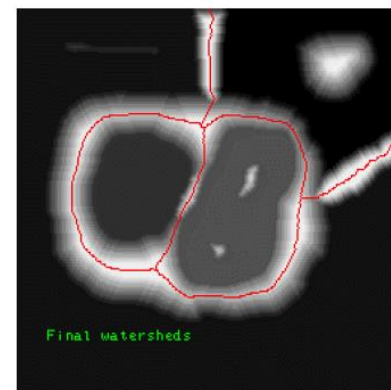
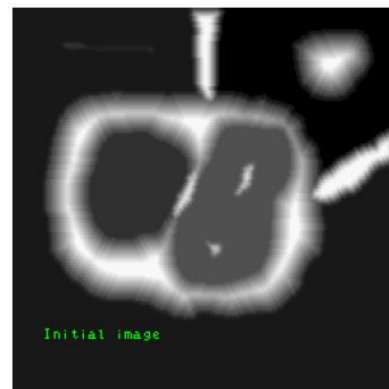
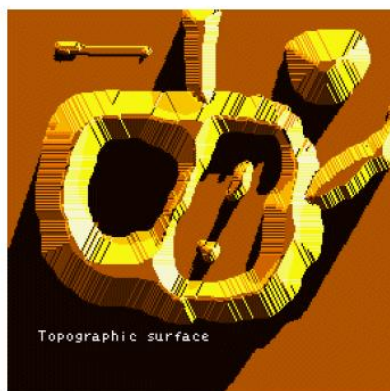
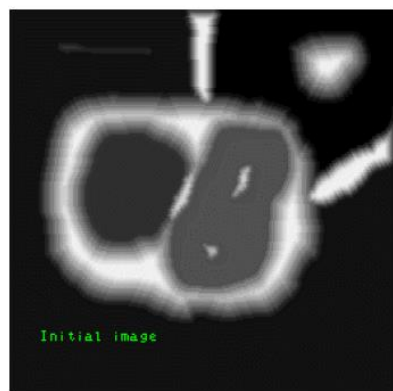
3. Background-foreground segmentation



Высокоуровневая сегментация (semantic segmentation)



Сегментация методом водораздела (Watershed)

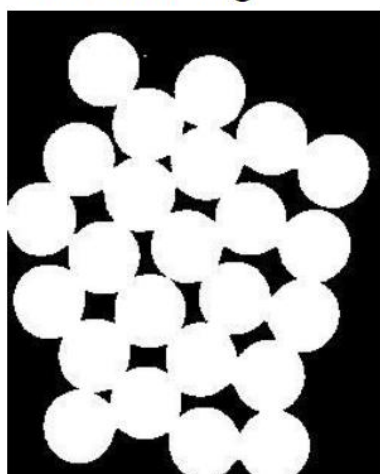


Cv::watershed

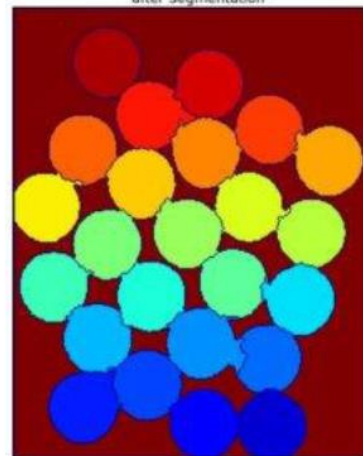
Initial image



Thresholding



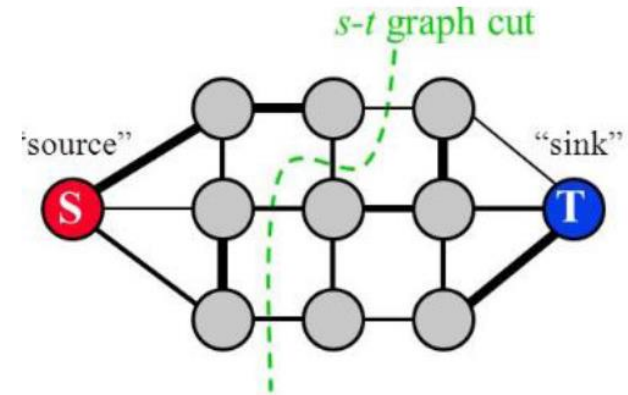
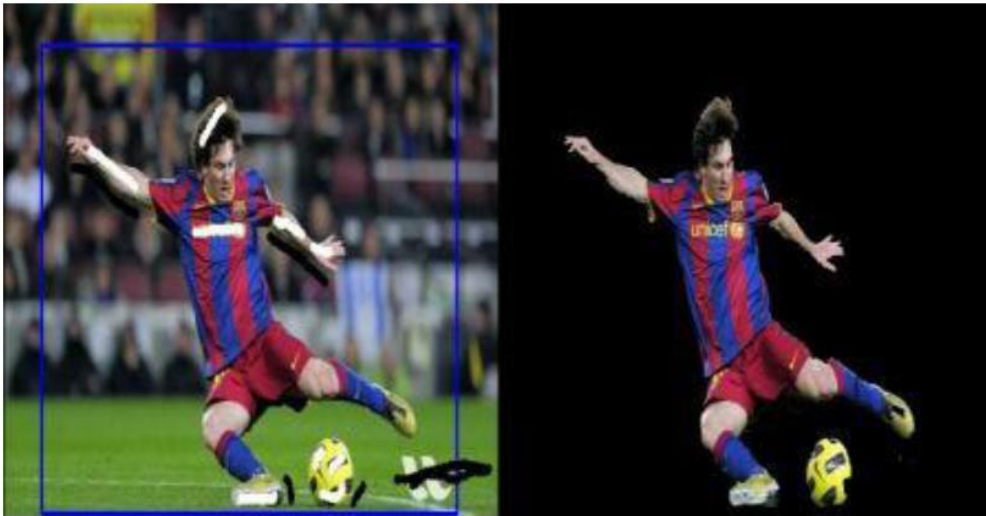
Marker image
after Segmentation



Result



Метод разреза графа (Graph-cut)



Cv::grabCut

0. Предподготовка: Белая кисть— передний план. Вне прямоугольника точно фон.
1. Далее запускается вероятностная оценка принадлежности точки к переднему плану или заднему.
2. Строится граф (каждая точка изображения - вершина графа), добавляются две вершины (sink-source), точки графа связываются с sink или source ребрами с весом пропорциональным вероятности их принадлежности классу, друг с другом согласно их схожести
3. Запускается алгоритм наименьшего разреза графа (min-cut). Делит весь граф на две части (передний – задний план)

Задача 11. Реализовать сегментацию изображений с помощью алгоритма watershed либо graphcut (на выбор), провести тестирование на различных примерах при разных параметрах.

Все задачи на одной странице:

Задача 6-1. Написать программу, выделяющую границы объектов методом Кэнни.

Задача 6-2. Автоматически определить размер объекта (маленький, средний, большой) относительно общего размера изображения.

Задача 6-3. Автоматически определить форму объекта (квадрат, прямоугольник, треугольник, круг, овал).

Задача 7. Продемонстрировать возможности линейной фильтрации изображений на основе: Гауссова фильтра, медианного, фильтра с произвольным ядром, оператора Собеля, фильтра Лапласа. Продемонстрировать удаление равномерного шума и Гауссова.

Задача 8. Реализовать поиск линий либо окружностей (на выбор) на характерных изображениях, исследовать эффективность при различных параметрах.

Задача 9. Использовать детектор углов Харриса и детектор особых точек Ши-Томаси, исследовать влияние преобразований изображений (аффинные и перспективные преобразования, изменение яркости, контраста), а также параметров алгоритмов на результаты обнаружения локальных особенностей.

Задача 10. Реализовать поиск соответствия особых точек на двух изображений (feature matching), исследовать работу в реальных условиях.

Задача 11. Реализовать сегментацию изображений с помощью алгоритма watershed либо graphcut (на выбор), провести тестирование на различных примерах при разных параметрах.

Задача 12. Дополнительная задача на выбор (на следующей странице)

Дополнительная задача 12 (любая из 4-х на выбор)

1. Реализовать алгоритм MNIST распознавания рукописных цифр, продемонстрировать несколько ошибочных примеров
2. Реализовать детектирование и распознавание номерных знаков автомобилей по фотографии (алгоритм ALPR или любой другой)
3. Реализовать алгоритм распознавания лиц на основе метода Виола-Джонса
4. Написать программу, определяющую возраст человека по его фотографии (любой метод)