

https://docs.opencv.org/3.4/d2/dbd/tutorial_distance_transform.html

Алгоритмы сегментации

Сегментация

Сегментация — это процесс разделения цифрового изображения на несколько сегментов.

Сегменты — неперекрывающиеся области, однородные по некоторому признаку.

Цель сегментации заключается в упрощении и/или изменении представления изображения, чтобы его было проще и легче анализировать.

Сегментация изображений обычно используется для того, чтобы выделить объекты и границы (линии, кривые, и т. д.) на изображениях.



- Медицинские изображения: обнаружение опухолей и других патологий, определение объёмов тканей, диагностика, изучение анатомической структуры
- Выделение объектов на спутниковых снимках
- Распознавание лиц
- Системы управления дорожным движением
- Обнаружение стоп-сигналов
- Машинное зрение
- Распараллеливание информационных потоков при передаче изображений высокого разрешения

Области применения

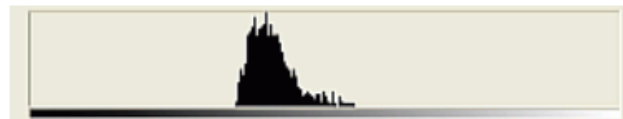
Критерии «похожести» пикселей

- По расстоянию
- По яркости
- По цвету
- По текстуре

Сегментация это выделение похожих пикселей.

Критерий однородности

- Гистограмма содержит не больше 1 значительного пика



- Отклонение любого пикселя от средней яркости $< T_{avg}$

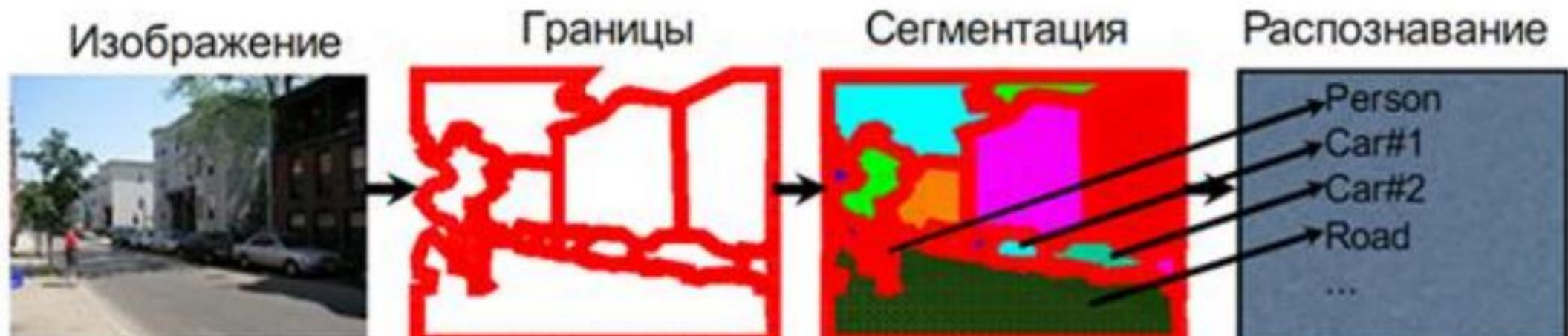
$$\forall p \in S \quad \left| I(p) - \frac{1}{N} \sum_{q \in S} I(q) \right| < T_{avg}$$


- Разница между соседними пикселями $< T_{diff}$

$$\forall p \in S, \forall q \in N(p) \quad |I(p) - I(q)| < T_{diff}$$

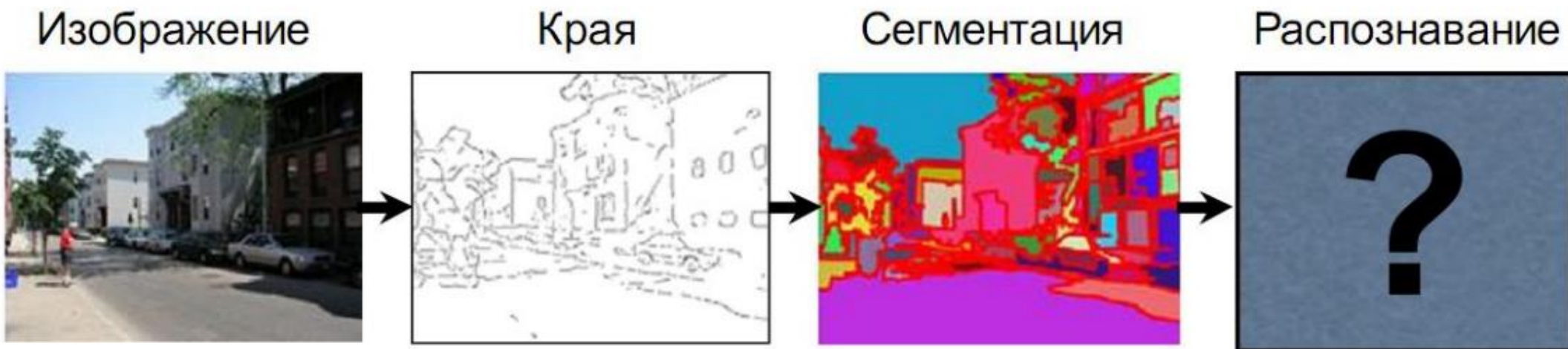
- «Слабая» граница между регионами (только для слияния)

Сегментация через края



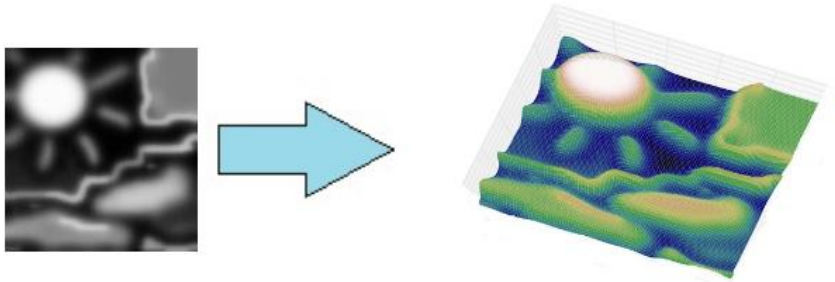
Теория 

Практика 

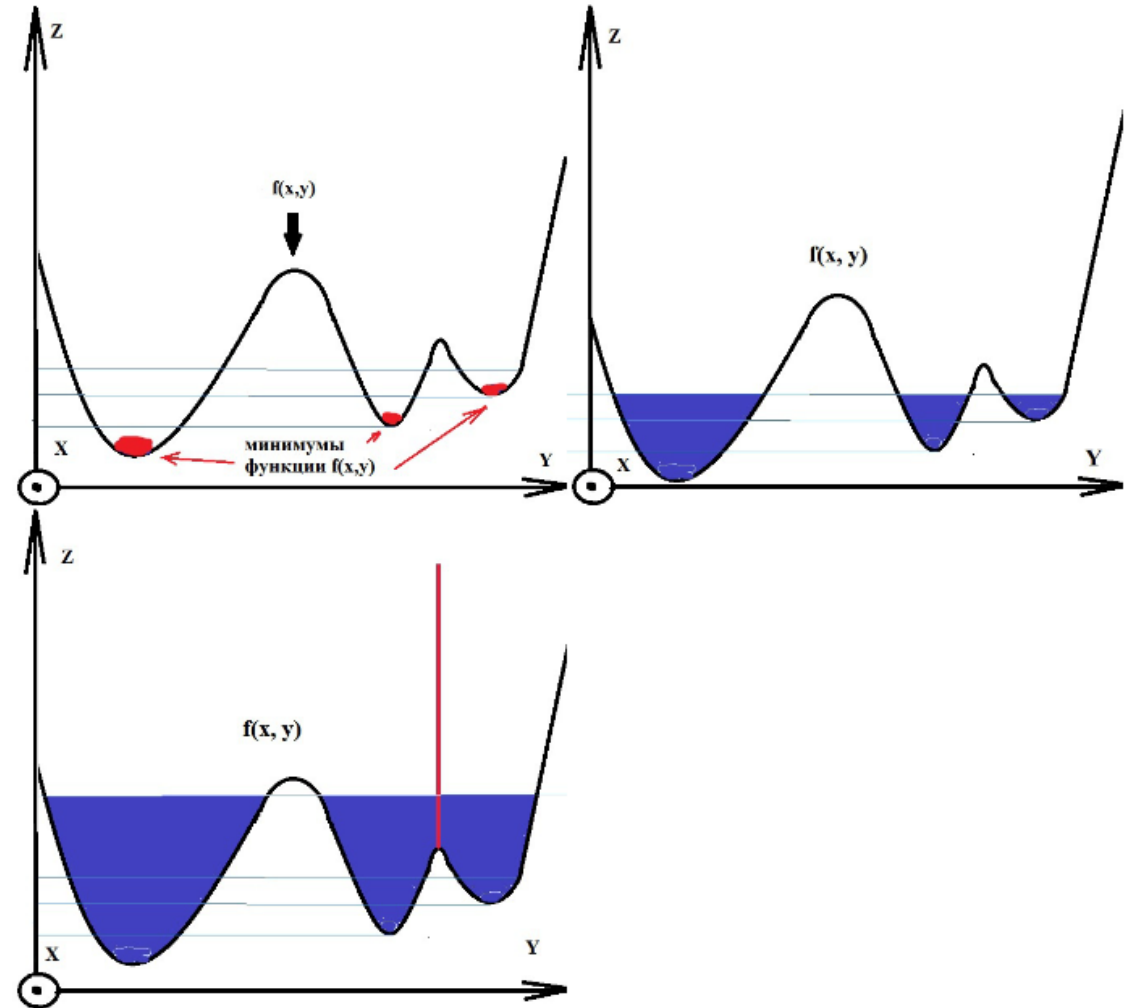


Алгоритм сегментации по водоразделам (WaterShed)

Значением функции может быть интенсивность или модуль градиента. Для наибольшего контраста можно взять градиент от изображения. Если по оси OZ откладывать абсолютное значение градиента, то в местах перепада интенсивности образуются хребты, а в однородных регионах – равнины.



После нахождения минимумов функции f , идет процесс заполнения “водой”, который начинается с глобального минимума. Как только уровень воды достигает значения очередного локального минимума, начинается его заполнение водой. Когда два региона начинают сливаться, строится перегородка, чтобы предотвратить объединение областей. Вода продолжит подниматься до тех пор, пока регионы не будут отделяться только искусственно построенными перегородками (см. рис. справа).



Однако на практике сегментации по водоразделам приводит к значительной чрезмерной сегментации из-за шума или локальных неоднородностей в градиентном изображении.



Разработано множество методов, позволяющих избежать чрезмерной сегментации, используя различные маркеры или работая с иерархией контуров сегментации.

На следующем слайде можно увидеть некоторые примеры работы алгоритмов сегментации водоразделом, подробнее про них можно посмотреть по ссылке:

<https://people.cmm.minesparis.psl.eu/users/beucher/wtshed.html>

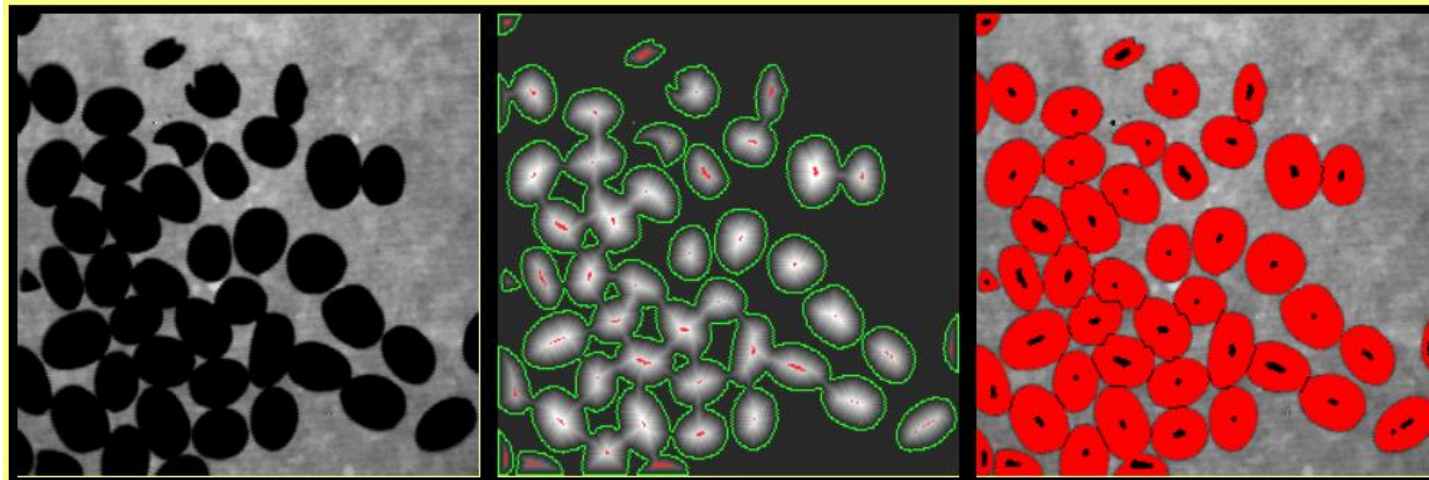
Сегментация дороги

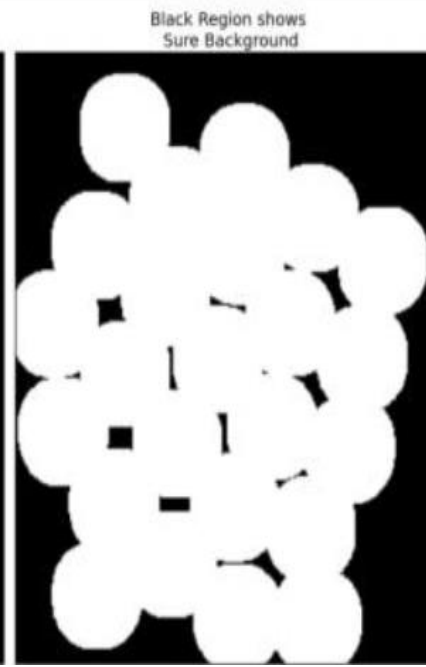
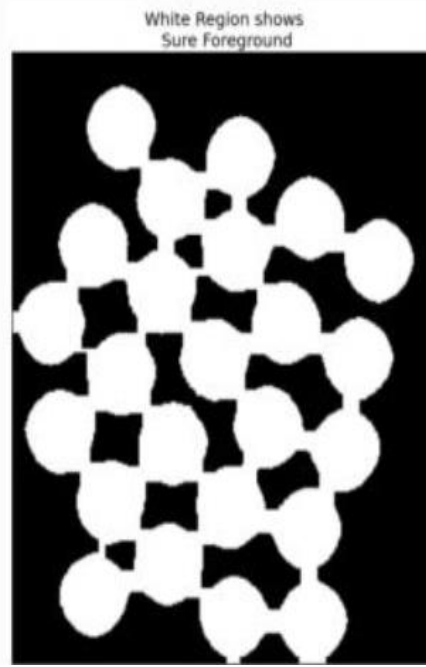
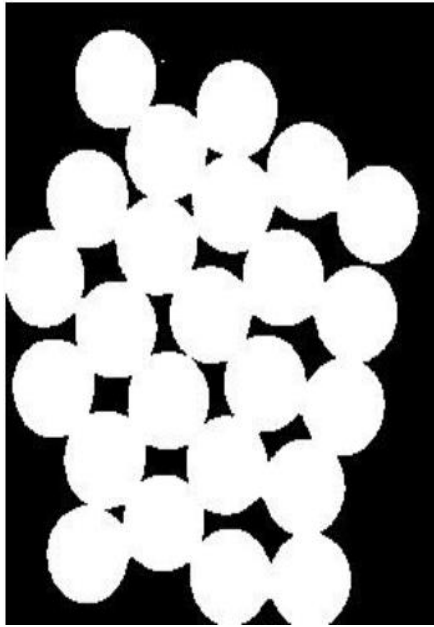
В этом примере маркеры были введены вручную.



Разделение кофейных зерен

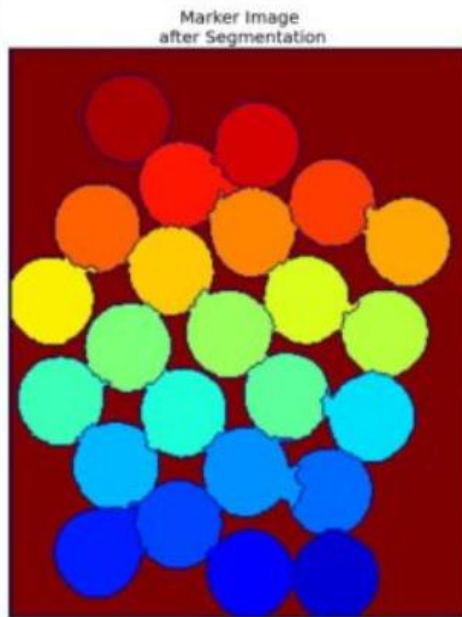
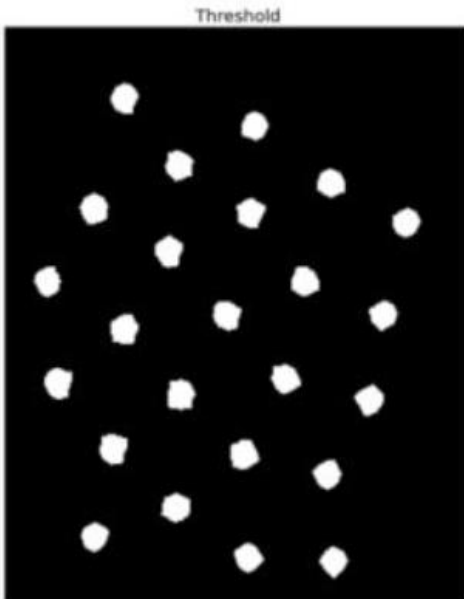
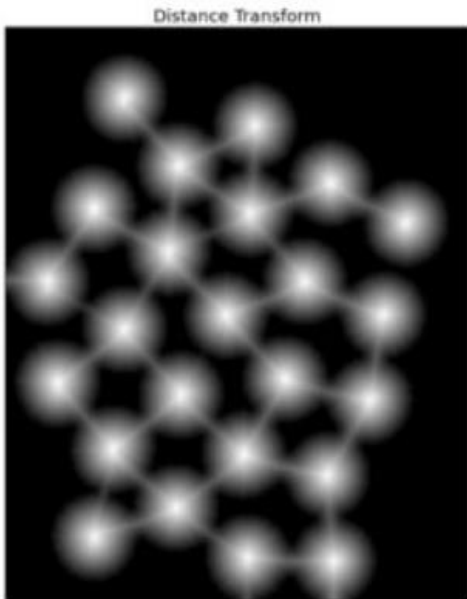
В этом случае в качестве критерия используется функция расстояния исходного изображения.





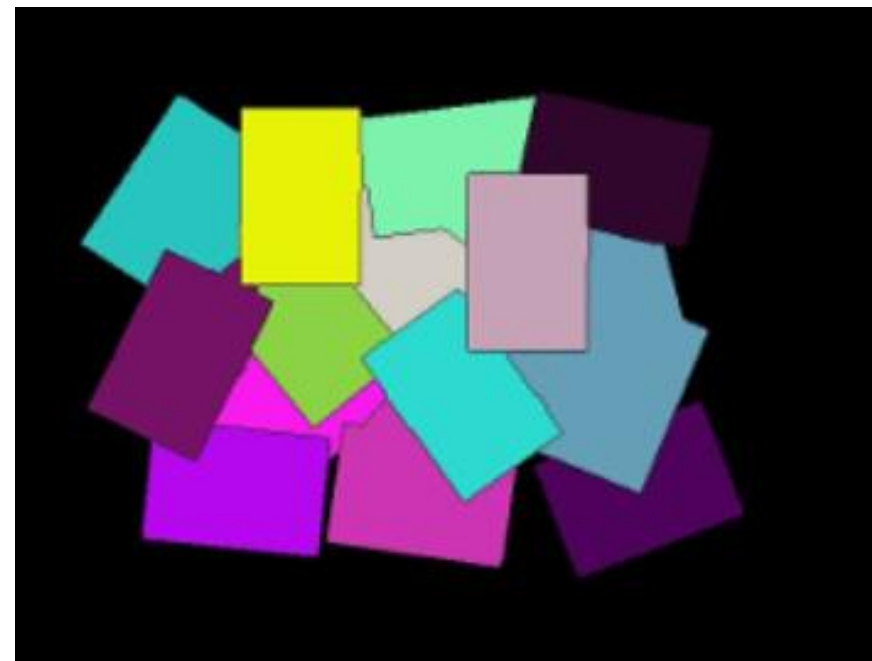
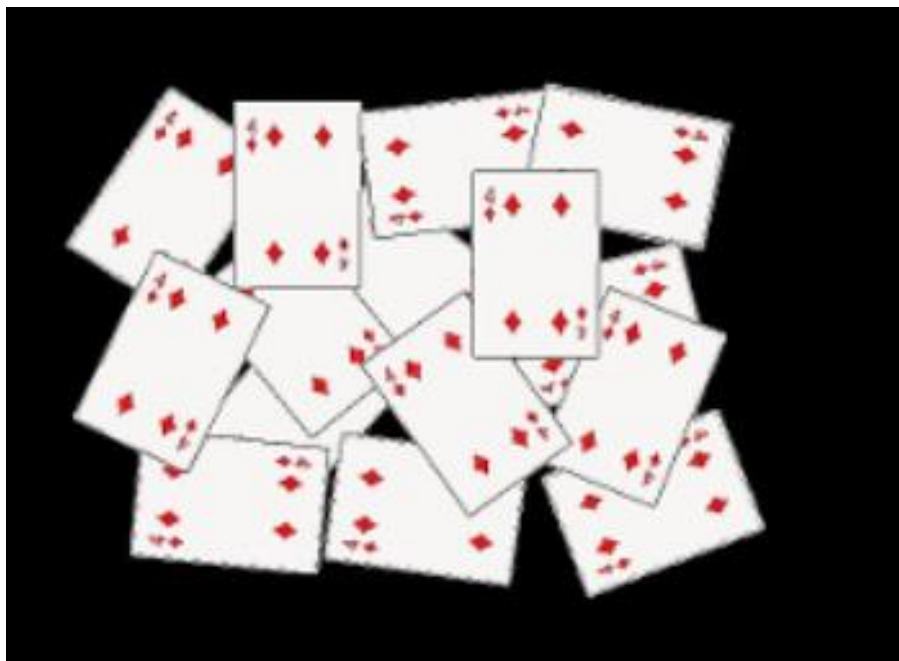
Поэтапное описание этого примера с сегментацией водоразделом по функции расстояния можно найти по ссылке:

https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_watershed/py_watershed.html



Еще один пример сегментации изображения с помощью дистанционного преобразования и алгоритма водораздела

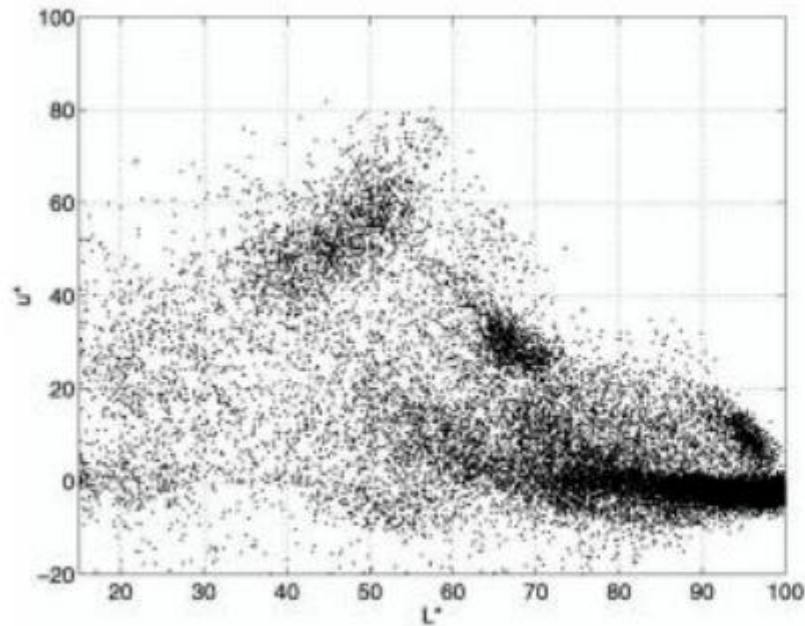
https://docs.opencv.org/3.4/d2/dbd/tutorial_distance_transform.html



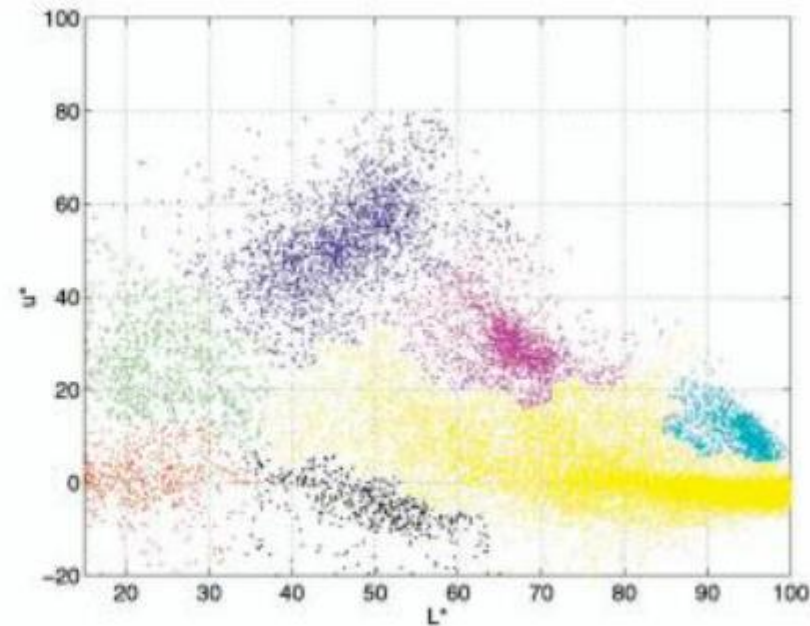
Алгоритм сегментации MeanShift

MeanShift группирует объекты с близкими признаками. Пиксели со схожими признаками объединяются в один сегмент, на выходе получаем изображение с однородными областями.

Например, в качестве координат **в пространстве признаков** можно выбрать координаты пикселя (x, y) и компоненты **RGB** пикселя. Изобразив пиксели в пространстве признаков, можно заметить сгущения в определенных местах.



(a)

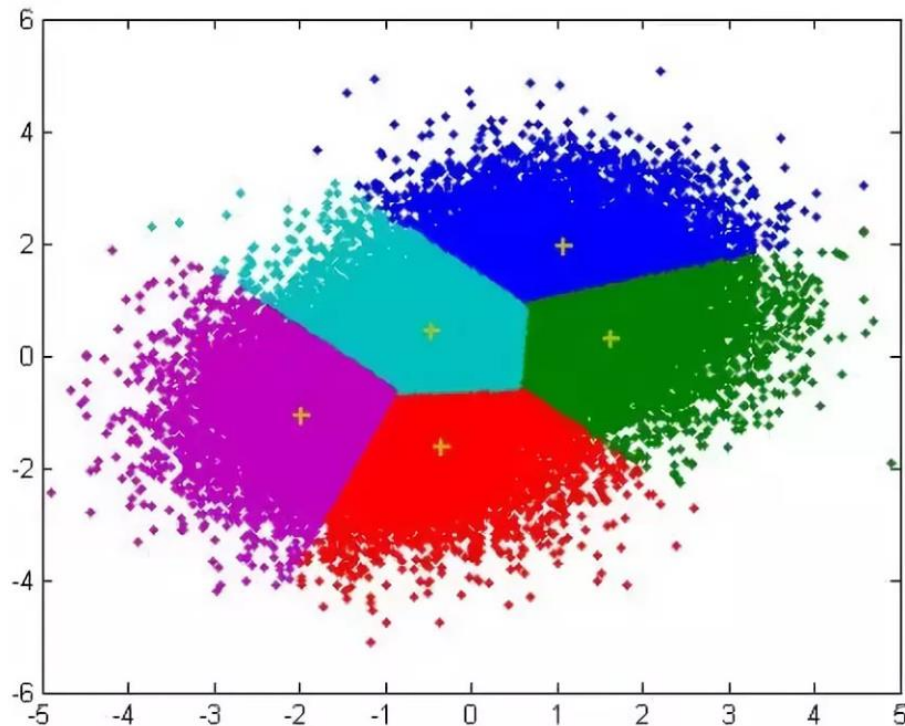


(b)

Метод центров тяжести (k-means)

Для решения задачи кластеризации в библиотеки OpenCV реализован метод центров тяжести (k-means).

Метод центров тяжести разбивает выборку на заданное количество кластеров путем выбора их центров. Поиск центров кластеров производится из соображений минимизации суммарного расстояния от каждой точки до ближайшего центра с помощью метода локальной оптимизации.



Пример сегментации изображения



Функция кластеризации kmeans()

```
import numpy as np
import cv2 as cv

img = cv.imread('home.jpg')
Z = img.reshape((-1,3))

# convert to np.float32
Z = np.float32(Z)

# define criteria, number of clusters(K) and apply kmeans()
criteria = (cv.TERM_CRITERIA_EPS + cv.TERM_CRITERIA_MAX_ITER, 10, 1.0)
K = 8
ret,label,center=cv.kmeans(Z,K,None,criteria,10,cv.KMEANS_RANDOM_CENTERS)

# Now convert back into uint8, and make original image
center = np.uint8(center)
res = center[label.flatten()]
res2 = res.reshape((img.shape))

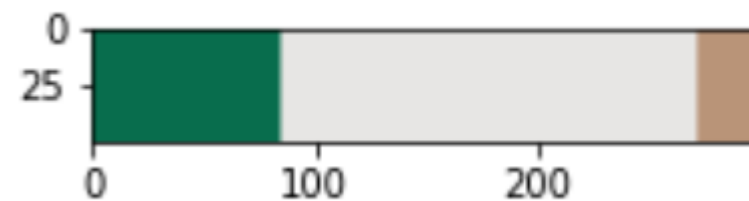
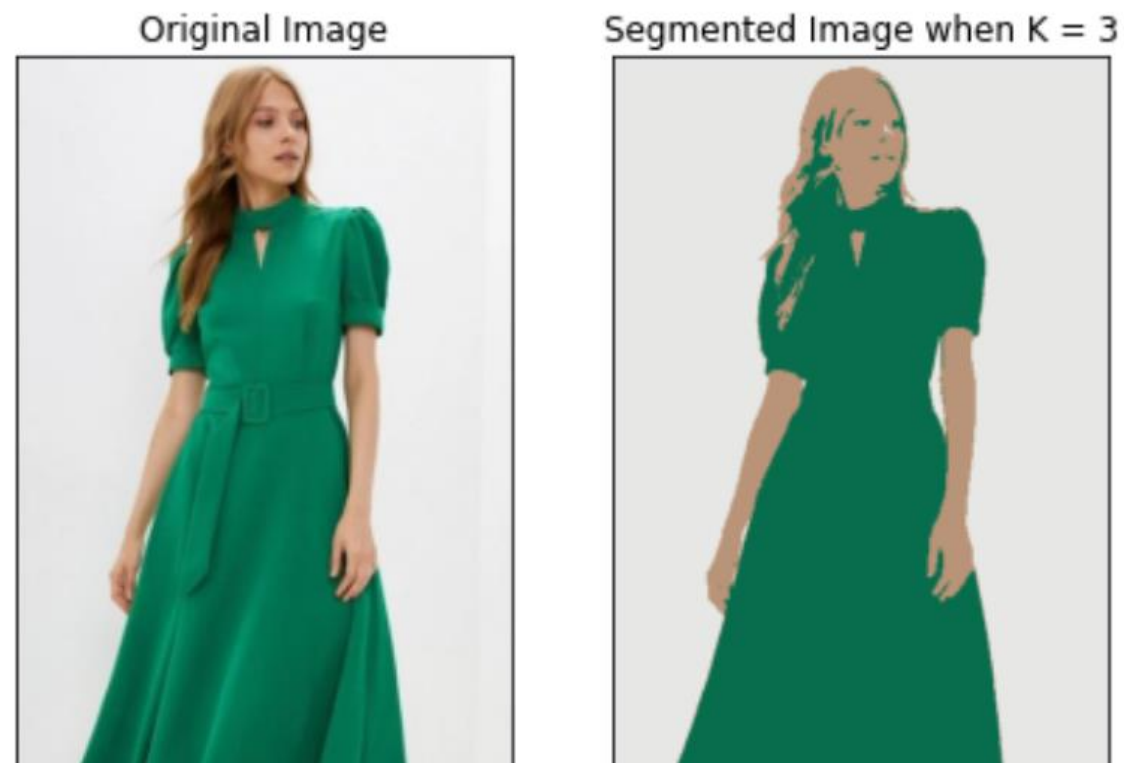
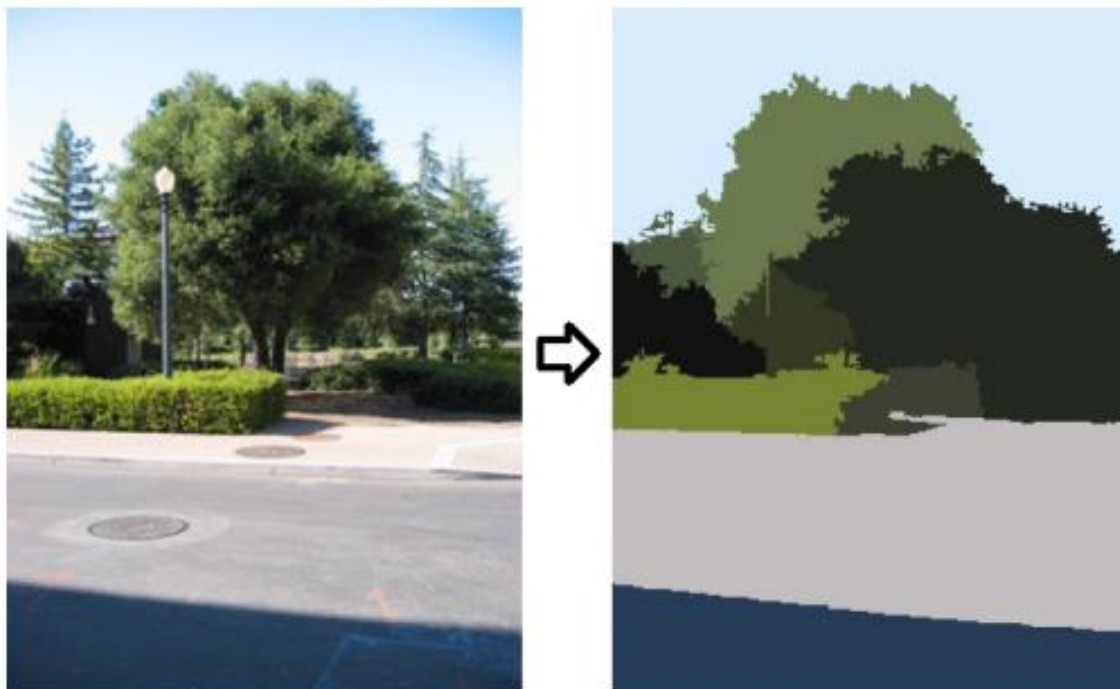
cv.imshow('res2',res2)
cv.waitKey(0)
cv.destroyAllWindows()
```

Квантование цвета — это процесс уменьшения количества цветов в изображении. Одной из причин для этого является уменьшение объема памяти. Иногда у некоторых устройств могут быть ограничения, например, они могут воспроизводить только ограниченное количество цветов. В этих случаях также выполняется цветовое квантование (см.пример)

https://docs.opencv.org/3.4/d1/d5c/tutorial_py_kmeans_opencv.html



Результаты сегментации

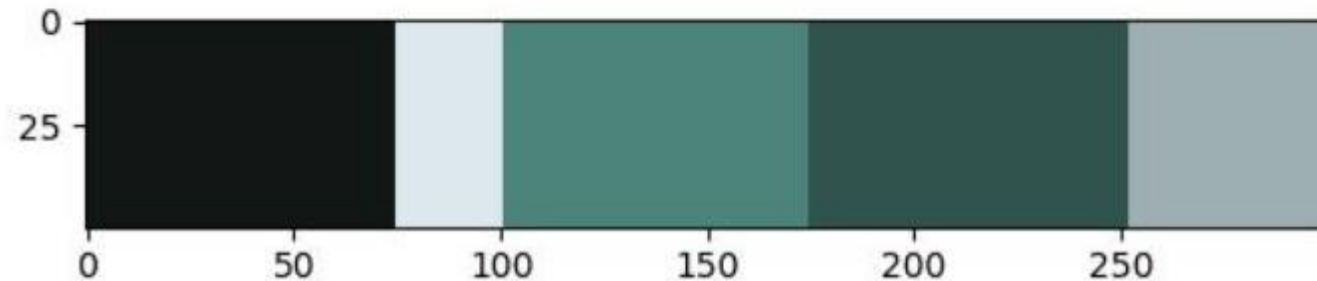


Определение доминирующих цветов методом кластеризации kmeans()



Пример реализации на Python

<https://russianblogs.com/article/1056533793/>



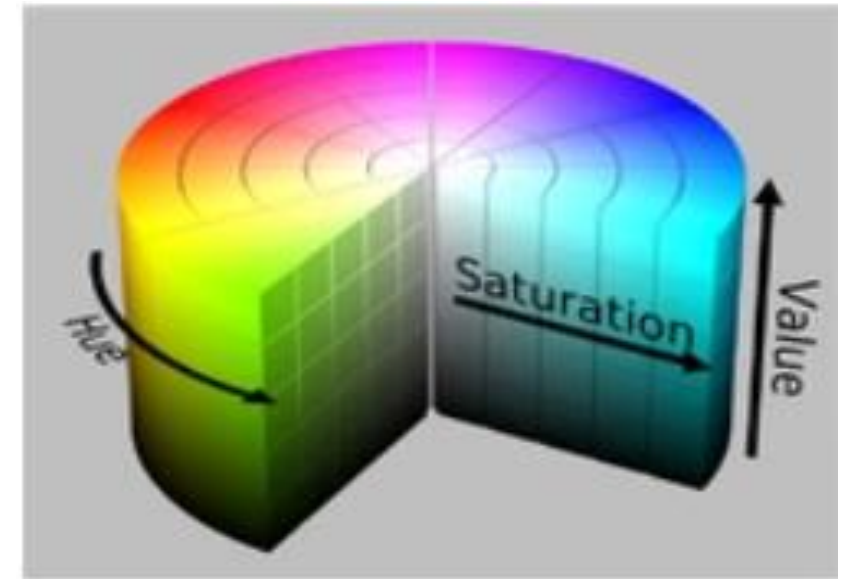
Выделение объектов по цветам

Как можно определить площадь леса на аэро снимке?



Самым простым критерием при выделении объектов классификации на изображении является цвет. Поэтому задачу разметки изображений можно свести к задаче **выделения объектов по цветам.**

По умолчанию в OpenCV цветное изображение хранится в палитре BGR. В RGB цвет пикселя определяется насыщенностью красным, зеленым и голубым. Таким образом, выбор диапазона оттенков одного и того же цвета становится не самой простой задачей. При использовании цветовой модели HSV сделать это гораздо проще. Эта цветовая схема определяется тремя компонентами (рисунок 1): Hue - цветовой тон, Saturation – насыщенность и Value - яркость. **Поскольку канал тона моделирует тип цвета, он очень полезен в задачах обработки изображений, которым необходимо сегментировать объекты на основе их цвета.**



```
cv.inRange(src, lowerb, upperb[, dst]) ->dst
```

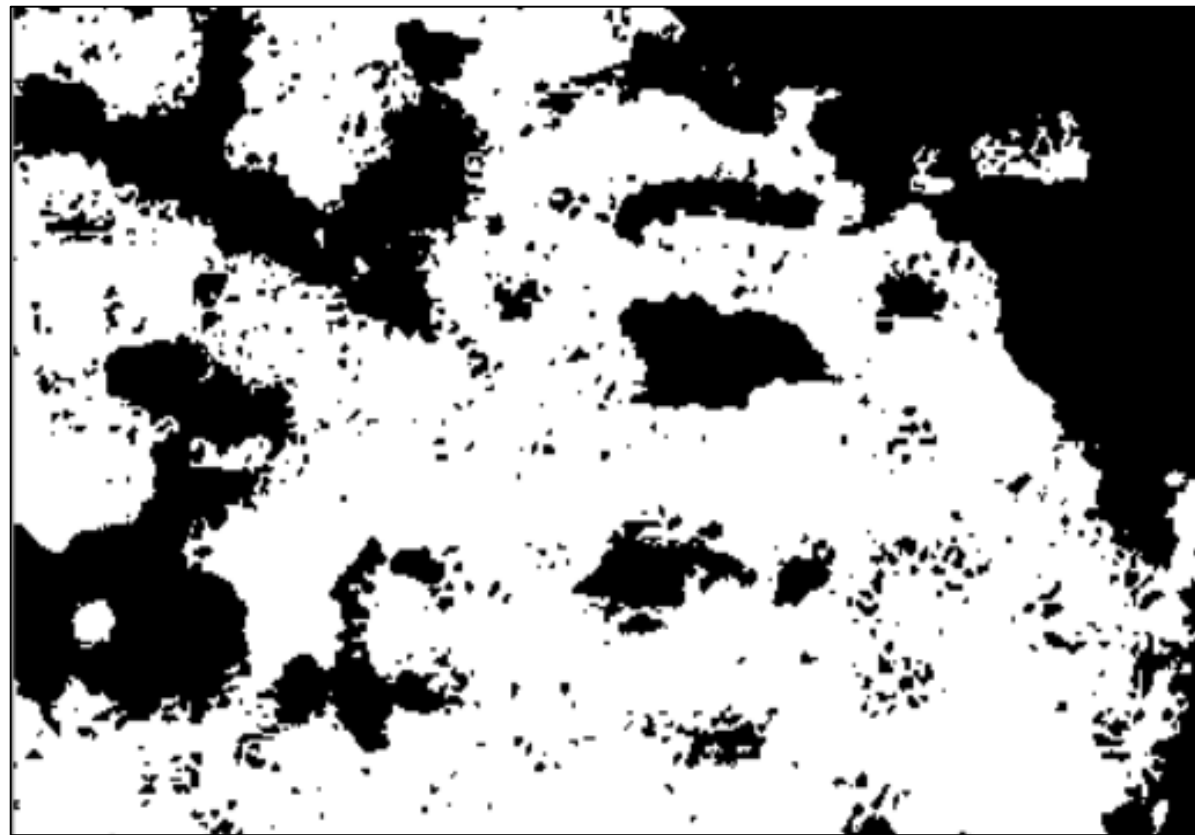
```
frame_HSV = cv.cvtColor(frame, cv.COLOR_BGR2HSV)
frame_threshold = cv.inRange(frame_HSV, (low_H, low_S, low_V), (high_H, high_S, high_V))
```

Convert RGB

<https://toolstud.io/color/rgb.php?rgbhex=AA6600&convert=rgbhex>

В рассматриваемом для зеленого цвета лесного массива приняты следующие ограничения

```
# диапазон изменения тона  
low_H = 50;  
high_H = 100;  
  
# диапазон изменения насыщенности  
low_S = 20;  
high_S = 256;  
  
# диапазон изменения яркости  
low_V = 10;  
high_V = 100;
```



Как можно улучшить?

После применения морфологических операций находим контуры, сортируем по длине и удаляем самые короткие. Затем можно найти площадь контуров и получить площадь лесных массивов на данной территории.

