



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ _____

КАФЕДРА _____ СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ (ИУ5) _____

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к домашнему заданию

по дисциплине: _____ Технологии мультимедиа _____

на тему: _____ Сегментация изображения методами SLIC, QuickShift _____

Студент ИУ5-61Б
(Группа)

(Подпись, дата) Алехин С. С.
(И.О.Фамилия)

Руководитель

(Подпись, дата) Афанасьев Г. И.
(И.О.Фамилия)

Оглавление

Введение	3
Идея	3
Исходные данные	5
Результат.....	5
Код программы	5
Вывод	6
Список литературы	6

Введение

Сегментация — это процесс разделения цифрового изображения на несколько сегментов (множество пикселей, также называемых суперпикселями)

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. Более точно, сегментация изображений — это процесс присвоения таких меток каждому пикселю изображения, что пиксели с одинаковыми метками имеют общие визуальные характеристики.

Идея

У нас есть изображения, нужно на них выделить суперпиксели методами SLIC и QuickShift. Рассмотрим, как работают методы

1. SLIC

Это алгоритм генерирует суперпиксели путем кластеризации пикселей на основе их сходства цвета и близости в плоскости изображения. Это делается в пятимерном пространстве $[labxy]$, где $[lab]$ — вектор цвета пикселей в пространстве CIELABcolor, а xy — положение пикселя. Нам нужно нормализовать пространственные расстояния, чтобы использовать евклидово расстояние в этом 5D пространстве, потому что максимально возможное расстояние между двумя цветами в пространстве CIELAB ограничено, в то время как пространственное расстояние в плоскости xy зависит от размера изображения. Поэтому для кластеризации пикселей в этом 5D-пространстве была введена новая мера расстояния, учитывающая размер суперпикселя, которая описана ниже.

Этот алгоритм принимает в качестве входного сигнала желаемое количество суперпикселей примерно одинакового размера K . В начале алгоритма центры K суперпиксельного кластера $S_k = [l_k, a_k, b_k, x_k, y_k]$ выбираются с $k = [1, K]$ через регулярные интервалы сетки S . Поскольку пространственная протяженность любого суперпикселя приблизительно равна S^2 (приблизительная площадь суперпикселя), можно смело предположить, что пиксели, связанные с этим центром кластера, лежат в пределах области $2S \times 2S$ вокруг суперпиксельного центра на плоскости xy . Нормализованная мера расстояния (D_s), используемая в 5D-пространстве, определяется как:

$$D_s = d_{lab} + (m/S) * dx_{xy} \dots \text{(eq 1)}$$

где $d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2}$, $dx_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}$ и D_s — сумма лабораторного расстояния (d_{lab}) и расстояния плоскости xy (dx_{xy}),

нормализованного интервалом сетки S . В D_s введена переменная m , позволяющая управлять компактностью суперпикселя. Чем больше значение m , тем компактнее кластер. Это значение может быть в диапазоне $[1, 20]$.

Этот алгоритм начинается с выборки K регулярно расположенных центров кластеров и перемещения их в места семян, соответствующие наименьшему положению градиента в районе 3×3 (это делается для того, чтобы избежать их размещения на краю и снизить шансы на выбор шумного пикселя). Градиенты изображения вычисляются как:

$$G(x,y) = \sqrt{I(x+1,y) - I(x-1,y)^2 + I(x,y+1) - I(x,y-1)^2}$$

где $I(x,y)$ - лабораторный вектор, соответствующий пикселю в позиции (x,y) , а \dots - норма L_2 . Это учитывает как информацию о цвете, так и об интенсивности. Каждый пиксель изображения связан с ближайшим центром кластера, область поиска которого перекрывает этот пиксель. После того, как все пиксели связаны с ближайшим центром кластера, новый центр вычисляется как средний лабораторный вектор всех пикселей, принадлежащих кластеру.

Процесс связывания пикселей с ближайшим центром кластера и пересчитывания центра кластера повторяется до сходимости. В конце этого процесса может остаться несколько бродячих меток, то есть несколько пикселей вблизи большого сегмента, имеющего ту же метку, но не связанного с ней. Связность может быть применена на последнем шаге алгоритма путем перемаркировки непересекающихся сегментов метками крупнейшего соседнего кластера.

2. QuickShift

QuickShift – это алгоритм быстрого поиска режима, похожий на средний сдвиг. Алгоритм сегментирует изображение RGB (или любое изображение с более чем одним каналом), идентифицируя кластеры пикселей в совместных пространственных и цветовых измерениях. Сегменты являются локальными (суперпикселями) и могут использоваться в качестве основы для дальнейшей обработки. Учитывая изображение, алгоритм вычисляет лес пикселей, ветви которых помечены значением расстояния. Это определяет иерархическую сегментацию изображения с сегментами, соответствующими поддеревьям. Полезные суперпиксели можно определить, разрезав ветви, метка расстояния которых выше заданного порога (порог может быть либо зафиксирован вручную, либо определен перекрестной валидацией).

Параметр, влияющий на алгоритм:

- Размер ядра. Плотность пикселей и ее режимы оцениваются с помощью оконного оценщика Parzen с гауссовым ядром

заданного размера. Чем больше размер, тем больше рассматриваемые окрестности пикселей.

- Максимальное расстояние. Это является максимальным расстоянием между двумя пикселями, которое алгоритм учитывает при построении леса. В принципе, это может быть бесконечность (так что дерево возвращается), но на практике гораздо быстрее рассматривать только относительно небольшие расстояния (максимальное расстояние может быть установлено на небольшое кратное размеру ядра).

Исходные данные

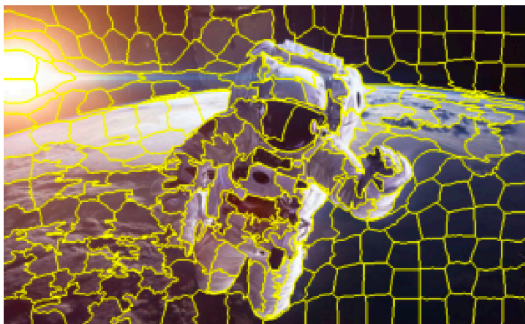


Пример фотографии, используемой для сегментации изображения

Результат

Визуализация обработанных изображений

SLIC



QuickShift



Код программы

```
import matplotlib.pyplot as plt
import numpy as np
```

```
from skimage.segmentation import felzenszwalb, slic, quickshift
from skimage.segmentation import mark_boundaries
```

```

from skimage import io

def seg_slic(img, imgname, ax, i):
    segments_slic = slic(img, n_segments=250, compactness=10, sigma=1, start_label=1)
    print(f'SLIC number of segments: {len(np.unique(segments_slic))}')

    ax[i].imshow(mark_boundaries(img, segments_slic))
    ax[i].set_title('SLIC')

    io.imsave(imgname, mark_boundaries(img, segments_slic))

def seg_quick(img, imgname, ax, i):
    segments_quick = quickshift(img, kernel_size=3, max_dist=6, ratio=0.5)
    print(f'QuickShift number of segments: {len(np.unique(segments_quick))}')

    ax[i].imshow(mark_boundaries(img, segments_quick))
    ax[i].set_title('QuickShift')

    io.imsave(imgname, mark_boundaries(img, segments_quick))

if __name__ == '__main__':
    img = io.imread('./test5.jpg')
    fig, ax = plt.subplots(1, 2)

    seg_slic(img, 'slic1.jpg', ax, 0)
    seg_quick(img, 'quick1.jpg', ax, 1)

    for a in ax.ravel():
        a.set_axis_off()

    plt.tight_layout()
    plt.show()

```

Вывод

В процессе выполнения домашнего задания мной были усвоены навыки построения разряженных моделей, также усвоена теория алгоритма, по которой строится разраженная модель.

Список литературы

1. https://scikit-image.org/docs/dev/auto_examples/segmentation/plot_segmentations.html
2. <https://scikit-image.org/docs/dev/api/skimimage.segmentation.html?highlight=segmentation#skimimage.segmentation.slic>
3. [https://ru.wikipedia.org/wiki/Сегментация_\(обработка_изображений\)](https://ru.wikipedia.org/wiki/Сегментация_(обработка_изображений))