

均值移动图像分割

实验目的

试自行拍摄一幅教学楼外立面图像，并使用均值移动的方法对该图像进行分割。

实验环境

- OpenCV
- NumPy
- Matplotlib

实验原理

均值移动 (Mean Shift) 是一种无监督的聚类算法，常用于图像分割。其核心思想是通过在特征空间中对数据点进行密度估计和迭代更新，寻找密度较高区域的中心（模式）。在图像分割中，均值移动方法将图像的像素值（包括颜色信息和空间坐标）映射到高维特征空间，通过不断更新每个像素所在的特征向量，使得特征向量逐渐靠近相应的密度峰值，进而完成分割。

数学模型

均值移动算法可以通过核密度估计的方式进行建模。假设在特征空间中有一个点 x_i ，其密度函数 $f(x)$ 通过核函数 $\phi(x)$ 表示为：

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^n \phi\left(\frac{x - x_i}{h}\right)$$

其中：

- n 是样本点的数量，
- x_i 是特征空间中的数据点，
- h 是带宽参数，控制核函数的窗口大小，
- d 是特征空间的维度。

均值移动的核心步骤是计算均值移动向量，其表达式为：

$$m(x) = \frac{\sum_{i=1}^n x_i \phi\left(\frac{x - x_i}{h}\right)}{\sum_{i=1}^n \phi\left(\frac{x - x_i}{h}\right)} - x$$

该向量指示了当前点 x 应该如何更新，以便移动到密度更高的区域。均值移动的目标是通过不断迭代，使数据点 x 移动到密度函数的峰值。

迭代求解

均值移动算法的迭代过程如下：

1. 选择一个初始点 x_0 ，并定义带宽参数 h 。
2. 计算均值移动向量 $m(x)$ 。
3. 更新数据点位置：

$$x_{t+1} = x_t + m(x_t)$$

4. 重复步骤 2 和 3，直到收敛条件满足，即当移动向量的范数 $|m(x)|$ 小于某个阈值时停止迭代。

通过这种迭代过程，所有数据点最终会聚集到一个密度峰值，从而实现特征空间中的聚类。在图像分割中，这意味着每个像素都被分配到对应的颜色簇，从而形成分割后的区域。

OpenCV

`cv2.pyrMeanShiftFiltering` 是 OpenCV 中实现均值移动（Mean Shift）算法的一种方法。其参数为

```
cv2.pyrMeanShiftFiltering(src, sp, sr, maxLevel=1, termcrit=None)
```

- `src`：输入图像，以矩阵的形式。
- `sp`：空间窗口的半径 h_s ，控制图像的平滑程度。值越大，空间范围内的像素影响越大。
- `sr`：颜色窗口的半径 h_c ，控制颜色范围的平滑程度。值越大，颜色空间的平滑程度越高。
- `maxLevel`：金字塔最大层数。图像金字塔是一种用于降低图像分辨率的技术，较大的金字塔层数能够在更低分辨率下进行平滑，计算效率更高。这用于控制样本点的数量 n 。
- `termcrit`：迭代终止条件，用于控制算法何时停止。可以根据最大迭代次数或误差阈值来终止。

染色

通过对连通分量染色，使得分割结果的可视化更清晰。

实验流程

设置空间半径、颜色半径为50和30。

```
In [1]: import cv2
import numpy as np
import matplotlib.pyplot as plt
from utils.seg_viz import label_segments, color_labels

# 读取图像
image = cv2.imread('example.jpg')
```

```

image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

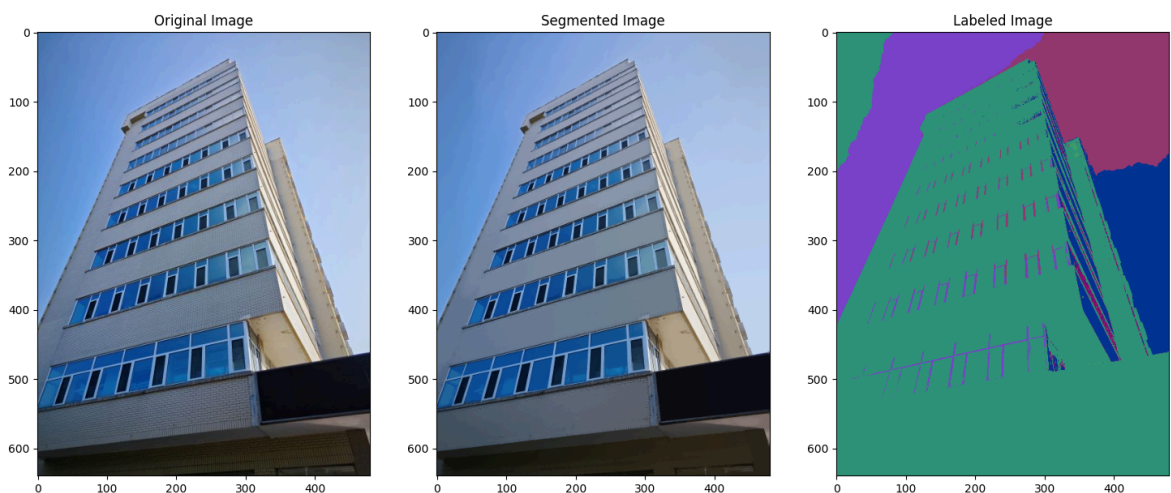
# 均值移动参数
spatial_radius = 50 # 空间半径
color_radius = 10 # 颜色半径
max_pyramid_level = 1

# 应用均值移动分割
segmented_image = cv2.pyrMeanShiftFiltering(image, spatial_radius, color_radius,

# 标记分割区域
num_labels, labels = label_segments(segmented_image, 5**3)
colored_labels = color_labels(labels)

# 显示结果
plt.figure(figsize=(18, 8))
plt.subplot(1, 3, 1)
plt.imshow(image)
plt.title('Original Image')
plt.subplot(1, 3, 2)
plt.imshow(segmented_image)
plt.title('Segmented Image')
plt.subplot(1, 3, 3)
plt.imshow(colored_labels)
plt.title('Labeled Image')
plt.show()

```



可以看到，均值移动算法使得图像平滑了。经过均值移动处理后，教学楼外墙瓷砖缝隙被几乎消除，但仍然存在；由于玻璃的透明和反光，导致其难以被完整分割。经过阈值处理后，教学楼被分割出来，并且阴面和阳面被分别分割；窗户玻璃没有被单独分割，但窗框被分割了；由于高层的玻璃反光，这对分割效果有一定影响，容易和天空混淆；由于天空面积较大，且左侧无云右侧有云，所以被分割成了四部分；教学楼的瓷砖缝隙被良好消除。总的来说分割效果良好，但教学楼砖缝、玻璃反光透光以及天空的非均匀色彩是难点，效果不是很好。