

# 基于图的图像分割说明文档

## 一、 软件概述

用基于图的算法进行图像分割，软件包含两个模块：graph 和 utilities 分别对应为两个 python 文件 graph.py 和 utilities.py。其中 graph.py 是主调函数也是图相关算法的实现算法，utilities 包含并查集和高斯模糊的相关函数。

## 二、 软件环境、配置和引用库

编程语言：Python3.6

操作系统：Ubuntu18.04

代码编辑器：vscode

调用 Python 库：sys（用于读取图片文件）、numpy、matplotlib.pyplot（用于显示图片）

软件工程目录：

```
Image_segmentation
|-res 资源文件夹
|-img 图像文件夹
|-doc 模块说明文档文件夹
|-src 代码文件夹
|-graph.py 图相关算法的主要实现模块
|-utilities.py 并查集和高斯模糊的主要实现模块
|-实验报告
|-项目说明书
```

## 三、 模块说明文档

- graph.py

Help on module graph:

NAME

graph

DESCRIPTION

Author: 彭瑶

Date: 2020/1/20

Description: 图相关的函数，用于图像分割

CLASSES

builtins.object

Edge

class Edge(builtins.object)

| 边

| 表示相邻像素点的连接边

|

| Attributes:

| self.a: 像素点 1 的 id

```

|     self.b: 像素点 2 的 id
|     self.w: 该边的权重
|
| Methods defined here:
|
| __init__(self, a, b, w)
|     Initialize self. See help(type(self)) for accurate signature.
|
| -----
---
| Data descriptors defined here:
|
| __dict__
|     dictionary for instance variables (if defined)
|
| __weakref__
|     list of weak references to the object (if defined)

```

## FUNCTIONS

`gen_edges(img, diff)`

将图片中相邻的像素点用边连接，得到一个联通图

Args:

`img`: 待处理图片

`diff`: 用于衡量像素点之间的差别

Returns:

所有边的列表

`gen_img(ds, img)`

由像素点组成的并查集 `ds` 来还原分割后的图片

Args:

`ds`: 像素点组成的并查集

`img`: 原图片

Returns:

返回还原好的图片

`main()`

`seg_img(img, k, min_pixels, radius, segma, diff=<function <lambda> at 0x7f3780ff3a60>, threshold=<function <lambda> at 0x7f3780ff3ae8>)`

图像分割的主调函数

Args:

img: 待分割的图像  
k: k 值，配合内聚函数使用  
min\_pixels: 最小的像素点聚类个数  
diff: 用于衡量像素点之间的差异程度  
threshold: 衡量像素点聚类的内聚度函数

Returns:

返回分割好的图像

FILE

/home/onlyrobot/E/Project/DigitalImageProcessing/image\_segmentation/src/graph.py

● utilities.py

Help on module utilities:

NAME

utilities

DESCRIPTION

Author: 彭瑶

Date: 2019/1/20

Description: 图像分割的实用函数

CLASSES

builtins.object

DisjointSet

class DisjointSet(builtins.object)

| 并查集

|

| Attributes:

| self.array: 并查集的关键数组

| self.sizes: 每个元素所属的集合元素个数

| self.values: 每个元素所属的集合的取值

|

| Methods defined here:

|

| \_\_getitem\_\_(self, key)

| 得到元素 key 所属集合的取值

|

| \_\_init\_\_(self, n, value=None)

```

|     Initialize self. See help(type(self)) for accurate signature.
|
|     __setitem__(self, key, value)
|         设定元素 key 所属集合的取值
|
|     find(self, a)
|         从并查集中寻找元素所属的集合
|
|     get_size(self, a)
|         得到元素 a 所属集合的大小
|
|     merge(self, a, b, value=None)
|         合并两个元素，并可选择指定合并之后的集合取值
|
|     Args:
|         a: 元素 a
|         b: 元素 b
|         value: 合并后他们所属集合的取值
|
| -----
---
| Data descriptors defined here:
|
|     __dict__
|         dictionary for instance variables (if defined)
|
|     __weakref__
|         list of weak references to the object (if defined)

```

## FUNCTIONS

guassian\_blur(img, radius, sigma)  
高斯模糊函数

Args:  
radius: 高斯核的半径  
sigma: 高斯分布的 sigma  
img: 待模糊图像

main()

## FILE

/home/onlyrobot/E/Project/DigitalImageProcessing/image\_segmentation/src/utilities.py

#### 四、 操作说明

主调函数位于 grapy.py 模块的 main 函数：

```
def main():  
    # 显示原图  
    img = plt.imread(sys.path[0] + '/../res/img/origin.png')  
    plt.subplot(121)  
    plt.imshow(img)  
    # 显示分割好的图（参数分别表示待分割图像、k 值、最小分割区域大小、高斯模糊
```

核半径及高斯函数的 sigma

```
    segged_img = seg_img(img, 1, 5000, 2, 1)  
    plt.subplot(122)  
    plt.imshow(segged_img)
```

```
    plt.show()
```

通过修改待分割图片的路径可以对其他图片进行分割，seg\_img 的参数介绍如下：

```
def seg_img(img, k, min_pixels, radius, segma, diff=lambda img, xy, xxyy:  
    np.sqrt(np.sum((0. + img[xy] - img[xxyy]) ** 2)),  
    threshold=lambda k, size: k / size):  
    '''图像分割的主调函数
```

Args:

img: 待分割的图像  
k: k 值，配合内聚函数使用  
min\_pixels: 最小的像素点聚类个数  
radius: 高斯模糊高斯核的半径大小  
segma: 高斯模糊的高斯函数的 segma 值  
diff: 用于衡量像素点之间的差异程度  
threshold: 衡量像素点聚类的内聚度函数

Returns:

返回分割好的图像

'''