

22336216_陶宇卓

实验五

要求

- a. 编写一个计算机程序/方法，能够用任意(即指定的)大小为3x3的结构元素进行二进制膨胀和腐蚀。
- b. 使用(a)的结果实现公式(9.18)所示的形态边界提取，对图 `image.jpg` 提取字符边界。

原理

- **膨胀**：通过将结构元素的中心移动到图像的每个像素位置，并将结构元素覆盖的区域中的任何一个像素值为1，则将该位置的像素值设为1。
- **腐蚀**：通过将结构元素的中心移动到图像的每个像素位置，并将结构元素覆盖的区域中的所有像素值都为1，则将该位置的像素值设为1。

形态边界提取公式表示通过从原始图像中减去腐蚀后的图像来提取边界。

具体实现

```

import numpy as np
from skimage.io import imread, imsave
from skimage.color import rgb2gray
from skimage.util import img_as_ubyte
import matplotlib.pyplot as plt

def load_image(filepath):
    """加载图像并确保是二值化灰度图"""
    image = imread(filepath)
    if image.ndim == 3: # RGB 图像
        image = rgb2gray(image)
    return image > 0.5

def binary_dilation_3x3(image):
    """使用3x3结构元素进行二进制膨胀"""
    structuring_element = np.ones((3, 3), dtype=bool)
    padded_image = np.pad(image, pad_width=1, mode='constant', constant_values=0)
    result = np.zeros_like(image, dtype=bool)
    for i in range(1, padded_image.shape[0] - 1):
        for j in range(1, padded_image.shape[1] - 1):
            if np.any(padded_image[i-1:i+2, j-1:j+2] & structuring_element):
                result[i-1, j-1] = 1
    return result

def binary_erosion_3x3(image):
    """使用3x3结构元素进行二进制腐蚀"""
    structuring_element = np.ones((3, 3), dtype=bool)
    padded_image = np.pad(image, pad_width=1, mode='constant', constant_values=0)
    result = np.zeros_like(image, dtype=bool)
    for i in range(1, padded_image.shape[0] - 1):
        for j in range(1, padded_image.shape[1] - 1):
            if np.all(padded_image[i-1:i+2, j-1:j+2] & structuring_element):
                result[i-1, j-1] = 1
    return result

def morphological_boundary_extraction(image):
    """形态边界提取"""
    eroded_image = binary_erosion_3x3(image)
    boundary = image & ~eroded_image
    return boundary

# 加载图像
image = load_image('image.jpg')

# 提取字符边界

```

```

boundary = morphological_boundary_extraction(image)

# 保存结果
imsave('boundary_result.png', img_as_ubyte(boundary))

# 可视化结果
fig, axes = plt.subplots(1, 2, figsize=(10, 5))

axes[0].imshow(image, cmap='gray')
axes[0].set_title("Original Image")
axes[0].axis('off')

axes[1].imshow(boundary, cmap='gray')
axes[1].set_title("Boundary Extraction Result")
axes[1].axis('off')

plt.tight_layout()
plt.show()

```

结果分析



通过上述代码，我们实现了使用3x3结构元素进行二进制膨胀和腐蚀，并通过形态边界提取公式提取了图像 `image.jpg` 中的字符边界。结果显示在 `boundary_result.png` 中，原始图像和边界提取结果并排显示在 `Figure_1.png` 中。

总结

本实验通过实现二进制膨胀和腐蚀操作，并结合形态边界提取公式，成功提取了图像中的字符边界。该方法可以有效地用于图像处理中的边界检测和形态学分析。