

浙江大学实验报告

专业：_自动化（电气）_
姓名：____潘盛琪____
学号：____3170105737____
日期：____3.21____
地点：生工食品学院机房

课程名称：计算机图像处理与机器视觉 指导老师：饶秀勤 成绩：_____

实验名称：阈值分割 实验类型：设计型

一、实验目的和要求

理解图像的内容，提取出我们感兴趣的对象

二、计算机配置与软件处理平台

硬件：

Windows 版本_____

Windows 10 家庭中文版

© 2018 Microsoft Corporation. 保留所有权利。

系统_____

处理器: AMD Ryzen 7 PRO 2700U w/ Radeon Vega Mobile Gfx 2.20 GHz

已安装的内存(RAM): 8.00 GB (6.93 GB 可用)

系统类型: 64 位操作系统, 基于 x64 的处理器

笔和触控: 没有可用于此显示器的笔或触控输入

软件：基于 matlab

三、算法描述

1. 迭代式阈值选择

3.3 迭代式阈值选择

1. 选一初始阈值，如:灰度均值 T_h
2. 利用阈值把图像分割成两组， R_1 和 R_2
3. 计算区域 R_1 和 R_2 的均值 μ_1 、 μ_2
4. 选择新的阈值 $T = (\mu_1 + \mu_2)/2$
5. 重复2-4步，直到 μ_1 和 μ_2 的均值不再变化为止 $(T_h - T) < 1$

阈值的改进策略是这一方法的关键

2. 大津法

3.6大津法（Otsu法或最大类间方差法）

最大类间方差法是由Otsu 于1979 年提出的，是基于整幅图像的统计特性实现阈值的自动选取的，是全局二值化最杰出的代表。

Otsu 算法的基本思想是用某一假定的灰度值 t 将图像的灰度分成两组，当两组的类间方差最大时，此灰度值 t 就是图像二值化的最佳阈值。

设图像有 M 个灰度级： $0 \sim M-1$ ，取 $0 < T_h < M-1$ ，将图像分成两组 G_0 和 G_1 ， G_0 组为灰度值在 $0 \sim T_h$ 的像素点，像素量为 n_0 ；其余的为 G_1 组，像素量为 n_1 ，用 N 表示图像像素总数 ($N = n_0 + n_1$)， $Pnum(k)$ 表示灰度值为 k 的像素的个数。

$$\text{整幅图像的质量矩: } sum = \sum_{k=0}^{255} (k \times Pnum(k))$$

$$\text{图像成分 } G_0 \text{ 的质量矩: } sum0 = \sum_{k=0}^{T_h} (k \times Pnum(k))$$

$$\text{平均灰度值: } u_0 = sum0 / n_0 \quad u_1 = (sum - sum0) / n_1$$

$$\text{间类方差为: } g(k) = n_0 \times n_1 (u_0 - u_1)^2$$

最佳阈值为: $T = \operatorname{argmax}(g(k))$ $k = 0 \sim 255$, 使得间类方差最大时所对应的 k 值即为最佳阈值 T

四、结果与讨论

4.1 输入图像

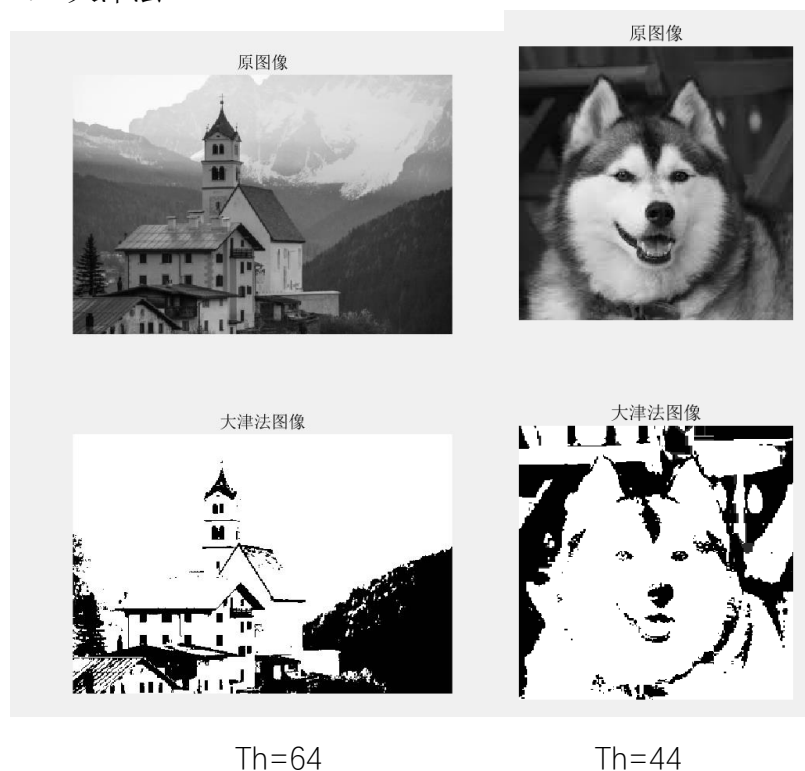


4.2 输出图像

1. 迭代式阈值选择



2. 大津法



4.3 讨论

可以看到在第一例中，这两种阈值选择算法的效果基本相同，计算所得的阈值仅相差 0.6。而在第二例中，两种阈值选择算法的效果相差比较明显

五、结论

迭代式阈值分割的关键在于阈值的改进策略

当物体目标与背景灰度差不明显时，使用大津法会出现无法忍受的大块黑色区域，甚至会丢失整幅图像的信息。

六、源程序

1. 迭代式阈值

%%迭代式阈值选择（二值分割）

```
img = imread('test.png');
```

```
img = rgb2gray(img);
```

```
Th = mean(img(:));
```

%将灰度均值设为初始阈值

```
newTh = 0;
```

```
i = 1;
```

```
while (Th - newTh) > 1
```

```
    pic1 = img;
```

```
    pic1(pic1 > Th) = 0;
```

%大于阈值置零方便求均值

```
    miu1 = mean(pic1(:));
```

%小于阈值的像素的灰度均值

```
    pic2 = img;
```

```

    pic2(pic2 < Th) = 0;           %小于阈值置零方便求均值
    miu2 = mean(pic2( : ));       %大于阈值的像素的灰度均值
    if i ~= 1
        Th = newTh;
    end
    i = i + 1;
    newTh = (miu1 + miu2) / 2;
end
imgout = img;
imgout(imgout < Th) = 1;
imgout(imgout > Th) = 256;
%%显示图像
subplot(2, 1, 1);
imshow(img);
title('原图像');
subplot(2, 1, 2);
imshow(uint8(imgout));
title('迭代阈值法图像');
2. 大津法
%%大津法（二值分割）
img = imread('test.png');
img = rgb2gray(img);
bestg_k = 0;
bestTh = mean(img( : ));        %设定灰度均值为初始阈值
for Th = 1 : 256
    %小于阈值的参数
    pic1 = img;
    pic1(pic1 > Th) = 0;
    miu1 = mean(pic1( : ));
    n1 = sum(sum(pic1 ~= 0));
    %大于阈值的参数
    pic2 = img;
    pic2(pic2 < Th) = 0;
    miu2 = mean(pic2( : ));
    n2 = sum(sum(pic2 ~= 0));
    %计算间类方差
    g_k = n1 * n2 * (miu1 - miu2) ^ 2;
    %当取到目前对大的阈值时，更新最佳阈值
    if g_k > bestg_k
        bestTh = Th;
        bestg_k = g_k;
    end
end
imgout = img;

```

```
imgout(imgout < bestTh) = 1;  
imgout(imgout > bestTh) = 256;  
%%显示图像  
subplot(2, 1, 1);  
imshow(img);  
title('原图像');  
subplot(2, 1, 2);  
imshow(uint8(imgout));  
title('大津法图像');
```