

# 浙江大学实验报告

专业：信息工程

姓名：徐晓刚

学号：3140102480

日期：2017 /5 /25

地点：

课程名称：数字图像处理实验 指导老师：项志宇 成绩：

实验名称：Otsu 阈值分割算法 实验类型：探究型 同组学生姓名：无

一、实验目的和要求（必填）

二、实验内容和原理（必填）

三、主要仪器设备（必填）

四、操作方法和实验步骤

五、实验数据记录和处理

六、实验结果与分析（必填）

七、讨论、心得

## 一、实验目的和要求

1. 掌握用 Otsu 方法的最佳全局阈值处理算法的原理以及实现过程
2. 尝试用该方法实现一幅图像的阈值分割，并且与 matlab 自带的阈值分割函数进行比较

## 二、实验内容和原理

### 1) 实验原理

#### 1. Otsu 确定全局阈值处理

Otsu 方法在类间方差最大的情况下是最佳的。令  $\{0,1,2,\dots,L-1\}$  表示一幅大小为  $M \times N$  像素的数字图像中的  $L$  个不同的灰度级， $n_i$  表示灰度级为  $i$  的像素数目。图像中像素总数  $MN$  为：

$$MN = n_0 + n_1 + \dots + n_{L-1}$$

归一化的直方图具有分量

$$p_i = \frac{n_i}{MN}, \sum_{i=0}^{L-1} p_i = 1$$

现在我们假设一个阈值  $T(k)=k$ ，并且使用它把输入图像阈值化处理为两类  $C_1$  和  $C_2$  使用该阈值，像素被分配到类  $C_1$  中的概率由以下的累积和给出：

$$P_1(k) = \sum_{i=0}^k p_i$$

类似的， $C_2$  类发生的概率为：

$$P_2(k) = \sum_{i=k+1}^{L-1} p_i$$

分配到类  $C_1$  的像素的平均灰度值为：

$$m_1(k) = \frac{1}{P_1(k)} \sum_{i=0}^k ip_i$$

分配到类  $C_2$  的像素的平均灰度值为：

$$m_2(k) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i$$

所以直至级  $k$  的累加均值由下式给出：

$$m(k) = \sum_{i=0}^k ip_i$$

而整个图像的平均灰度由下式给出：

$$m_G = \sum_{i=0}^{L-1} ip_i$$

为了评价级别  $k$  处的阈值的质量，我们使用归一化的无量纲矩阵：

$$\eta = \frac{\sigma_B^2}{\sigma_G^2}$$

其中

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 p_i$$

而  $\sigma_B^2$  为类间方差：

$$\begin{aligned} \sigma_B^2(k) &= P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 \\ &= \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]} \end{aligned}$$

我们可以看出，两个均值彼此间隔越远的话，类间方差越大，这表明类间方差是类之间的可分性度量。因为  $\sigma_G^2$  是一个常数，所以  $\eta$  也是一个可分性度量。所以我们的阈值就是最大化  $\sigma_B^2(k)$  的  $k$  值：

$$\sigma_B^2(k^*) = \max_{0 \leq k \leq L-1} \sigma_B^2(k)$$

Otsu 算法小结如下：

1. 计算输入图像的归一化直方图，用  $p_i$  表示直方图的分量；
2. 对于  $k = 0, 1, 2, \dots, L-1$  计算累计和  $P_1(k)$ ；
3. 对于  $k = 0, 1, 2, \dots, L-1$  计算累计均值  $m(k)$ ；
4. 计算全局灰度均值  $m_G$ ；
5. 对于  $k = 0, 1, 2, \dots, L-1$ ，计算类间方差  $\sigma_B^2$ ；
6. 得到 Otsu 的阈值  $k^*$ ，就是得到  $\sigma_B^2$  最大的  $k$  值，如果最大值不唯一，那么就应该各个最大值  $k$  进行平均取值；
7. 在  $k^*$  处得到可分性度量  $\eta$

## 2) 实验内容

用 Otsu 方法的最佳全局阈值处理算法的原理以及实现过程。

用该方法实现一幅图像的阈值分割，并且与 matlab 自带的阈值分割函数进行比较。

## 三、主要仪器设备

计算机，Matlab

## 四、操作方法和实验步骤

具体的代码实现采用 Matlab 编程环境

## 五、实验数据记录和处理

在这里我们使用的原始图像如下所示：



1. 首先我们需要编写 Otsu 函数，在这里可以直接按照上述实验原理中的小结部分，写出计算类间方差的程序，并且求取  $k^*$

程序如下所示：

```
function [dstimg, eta, result] = Otsu(srcimg)
%该函数是用来实现Otsu全局阈值分割的
%输入为一幅图像srcimg
%输出dstimg为阈值分割之后的图像，eta是可分性度量，result是最佳的阈值
if length(size(srcimg))==3
    srcimg = rgb2gray(srcimg);
end
%首先统计直方图，并且做归一化
Number=imhist(srcimg);
S = sum(Number);
Number = Number/S;

%统计累积和以及累积均值
P1 = zeros(length(Number), 1);
M1 = zeros(length(Number), 1);
for k = 1:length(Number)
    P = 0;
    M = 0;
    for i = 1:k
        P = P + Number(i);
        M = M + (i-1)*Number(i);
    end
    P1(k) = P;
    M1(k) = M;
end
%计算全局灰度均值
mg = M1(length(Number));
```

```

%计算类间方差和全局方差
sigmaB = zeros(length(Number), 1);
sigmaG = 0;
]for k = 1:length(Number)
    sigmaG = sigmaG + (k- 1 - mg)^2 * Number(i);
    sigmaB(k) = ((mg * P1(k) - M1(k))^2) / (P1(k) * (1-P1(k)));
end

%找出最大的k就是最后的最佳阈值
maxnum = max(sigmaB);
result = find(sigmaB == maxnum);
result = mean(result);

%计算可分性度量
eta = sigmaB(result) / sigmaG;

```

---

其中可以计算出最佳分割阈值，用这个 Otsu 程序计算出的阈值为 118

而我们调用 matlab 自带的函数：thresh = graythresh(im);

可以得到一个阈值的结果是 117，可见两者相差很近，所以可以认为我们的确定最佳阈值的程序编写是成功的。

## 2. 阈值分割以及结果：

下面的代码用来实现阈值分割：

```

%进行阈值分割
[m ,n] = size(srcimg);
dstimg = zeros(m, n);
]for i = 1:m
    for j = 1:n
        if srcimg(i,j) > result
            dstimg(i,j) = 1;
        else
            dstimg(i,j) = 0;
        end
    end
end
end

```

我们得到的结果图像如下所示：



而如果我们使用 matlab 自带的函数确定的阈值，最后得到的结果如下：



可见两者的结果十分接近，说明我们的算法的效果是正确的。

## 六、实验结果与分析

实验中通过使用 Otsu 算法来确定全局最佳阈值。这个方法是建立在最大化类间方差的基础之上的。我们知道类间方差越大，图像的可分性也就越好，而这个算法在类间方差最大的情况下是最佳的。而且这个算法有一个重要的特性，就是它完全在一幅图像的直方图上执行计算为基础，很容易得到一维阵列。

本次实验通过 matlab 编写代码实现了 Otsu 算法，并且将其效果与 matlab 自带的确定阈值的函数的结果进行比较。最后的结果显示我们的算法是正确的。

## 七、讨论、心得

本次实验的原理性是比较具有顺序的。主要需要我们先去掌握这个算法的合理性。在全局阈值的处理中，通过最大化类间方差，并且对图像进行阈值分割的操作。这个方法如果能够之后结合一些图像预处理的方法，想必会有更好的效果。另外 matlab 里面自带的确定灰度图像分割阈值的方法也是 Otsu，表明这个算法确实很有用。