

图像分割：阈值分割

2019年10月30日 20:40

简介

阈值分割是一种基于区域的图像分割技术，原理是通过设定不同的特征阈值，把图像像素点分成若干类。阈值分割是一种传统的最常用的图像分割方法，因其实现简单、计算量小、性能稳定而成为图像分割技术中最基本和应用最广泛的分割技术。特别适用于目标和背景占据不同灰度级范围的图像。基于灰度阈值的分割方法，关键是如何合理地选择阈值。

原理

(1) 人工选择法

人工选择法是通过人眼的观察，应用人对图像的认识，在分析图像直方图的基础上，人工选择出合适的阈值。也可以在人工选出阈值后，根据分割效果，不断地交互操作，从而选择出最佳的阈值。

(2) 迭代式阈值选择法

迭代式阈值选择法的基本思想是：开始时选择一个阈值作为初始估计值，然后按照某种策略不断地改进这一估计值，直到满足给定的准则为止。在迭代过程中，关键之处在于选择什么样的阈值改进策略。好的阈值改进策略应该具备两个特征：一是能够快速收敛，二是在每一个迭代过程中，新产生的阈值优于上一次的阈值。其算法步骤如下：

- 选择图像灰度的初值作为初始阈值 T_0 。
- 利用阈值 T_i 将图像分割成两个区域—— R_1 和 R_2 ，用下式计算区域 R_1 和 R_2 的灰度均值 μ_1 和 μ_2 ：

$$\mu_1 = \frac{\sum_{j=0}^{T_i} jp_j}{\sum_{j=0}^{T_i} p_j} \quad \mu_2 = \frac{\sum_{j=T_i}^{L-1} jp_j}{\sum_{j=T_i}^{L-1} p_j}$$

其中， L 是图像的灰度总级数， p_j 是第 j 个灰度级在图像中出现的次数。

- 计算出 μ_1 和 μ_2 后，用下式计算出新的阈值 T_{i+1} ：

$$T_{i+1} = \frac{1}{2}(\mu_1 + \mu_2)$$

- 重复步骤b~c，直到 T_{i+1} 和 T_i 的差小于某个给定值。

(3) 最大类间方差阈值选择法 (Otsu)

最大类间方差阈值选择法又称为Otsu算法，该算法是在灰度直方图的基础上用最小二乘法推理出来的，具有统计意义上的最佳阈值分割。它的基本原理是以最佳阈值将图像的灰度直方图分割为两部分，使两部分之间的方差取得最大值，即分离性最大。

设X是一副具有L级灰度级的图像，其中第i级像素为 n_i 个，其中i的值在 $0 \sim L-1$ 之间，图像的总像素点个数为：

$$N = \sum_{i=0}^{L-1} n_i$$

第i级出现的概率为：

$$p_i = \frac{n_i}{N}$$

在Otsu算法中，以阈值k将所有的像素分为目标 C_0 和背景 C_1 两类。其中， C_0 类的像素灰度级为 $0 \sim k-1$ ， C_1 类的像素灰度级为 $k \sim L-1$ 。

图像的总平均灰度级为：

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

C_0 类像素所占面积比例为：

$$\omega_0 = \sum_{i=0}^{k-1} p_i$$

C_1 类像素所占面积比例为：

$$\omega_1 = \sum_{i=k}^{L-1} p_i = 1 - \omega_0$$

C_0 类像素的平均灰度为：

$$\mu_0 = \mu_0(k) / \omega_0$$

C_1 类像素的平均灰度为：

$$\mu_1 = \mu_1(k) / \omega_1$$

其中

$$\mu_0(k) = \sum_{i=0}^{k-1} ip_i$$
$$\mu_1(k) = \sum_{i=k}^{L-1} ip_i = \mu - \mu_0(k)$$

由上面几个式子可得：

$$\mu = \omega_0 \mu_0 + \omega_1 \mu_1$$

则类间方差公式为：

$$\sigma^2(k) = \omega_0(\mu_0 - \mu)^2 + \omega_1(\mu_1 - \mu)^2 = \omega_0 \omega_1 (\mu_0 - \mu_1)^2$$

令k从 $0 \sim L-1$ 变化取值，计算在不同k之下的类间方差 $\sigma^2(k)$ ，使得 $\sigma^2(k)$ 最大的k值就是所要求的最佳阈值。

MATLAB代码

```
clc;clear;
I = imread('cameraman.tif');
imshow(I);

%%
% (1) 人工选择法
T1 = 78;
[ht, t] = imhist(I);
figure();
stem(t, ht, 'LineStyle', '-', 'Marker', 'none');
xlim([min(t), max(t)]);
hold on;
stem(T1, ht(T1), 'Marker', 'o');
xticks(T1);
hold off;
I1 = I > T1;
figure();imshow(I1);

%%
% (2) 迭代式阈值选择法
[ht, t] = imhist(I);
T2_pre = round(mean([min(t), (max(t))]));
while 1
    u1 = sum(ht(t <= T2_pre) .* t(t <= T2_pre)) / sum(ht(t <= T2_pre));
    u2 = sum(ht(t > T2_pre) .* t(t > T2_pre)) / sum(ht(t > T2_pre));
    T2 = round(mean([u1, u2]));
    if abs(T2 - T2_pre) <= 1
        break;
    end
    T2_pre = T2;
end
I2 = I > T2;
figure();imshow(I2);

%%
% (3) 最大类间方差阈值选择法
%代码实现
[ht, t] = imhist(I);
N = sum(ht);
L = length(ht);
p = ht / N;
sigma2 = zeros([1, L]);
for i = 1:L
    k = t(i);
    omega0 = sum(p(t < k));
    omega1 = sum(p(t >= k));
    if omega0 == 0 || omega1 == 0
        sigma2(i) = 0;
    else
        mu0 = sum(p(t < k) .* t(t < k)) / omega0;
        mu1 = sum(p(t >= k) .* t(t >= k)) / omega1;
        sigma2(i) = omega0 * omega1 * (mu0 - mu1) ^ 2;
        disp([k, omega0, omega1, mu0, mu1, sigma2(i)]);
    end
end
end
[~, index] = max(sigma2);
T3 = t(index);
I3 = I > T3;
```

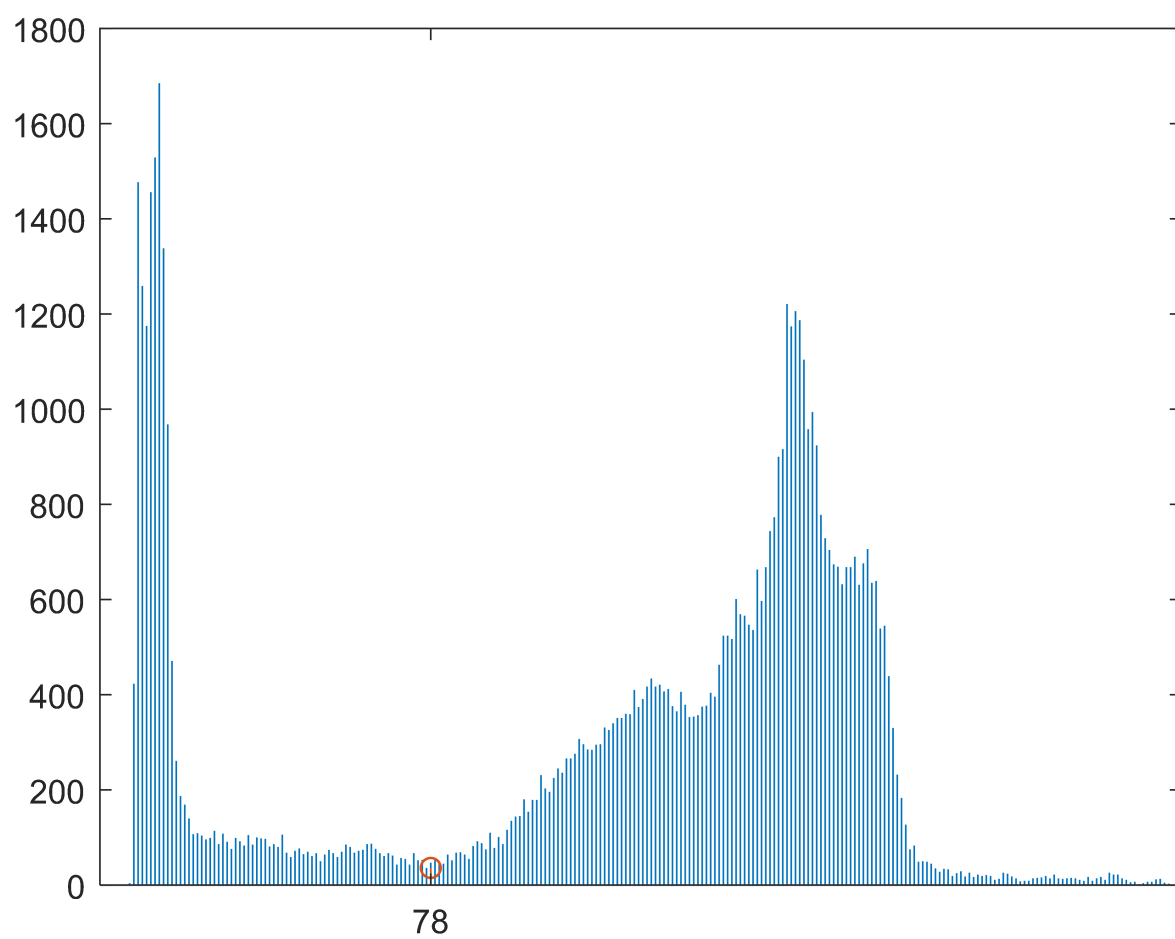
```
figure();imshow(I3);  
%graythresh函数实现  
T4 = graythresh(I) * 255;  
I4 = I > T4;  
figure();imshow(I4);
```

结果分析

图片用的经典的cameraman



首先是人工选择法，通过观察直方图有两个比较明显的峰，将阈值选为78





迭代式阈值选择法设定条件两次阈值的差小于或等于1，最后得出的阈值为89



最大类间方差阈值选择法中，代码实现得到的令类间方差最大的阈值为89，同迭代式阈值选择法一样，利用graythresh函数得到的阈值为88，下图为阈值为88时的图像分割结果

