**实验报告**

装 订 线

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**

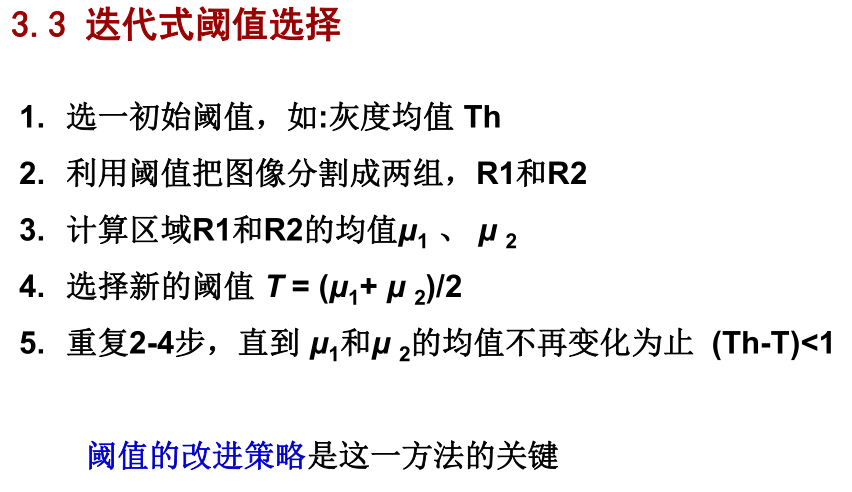
硬件：



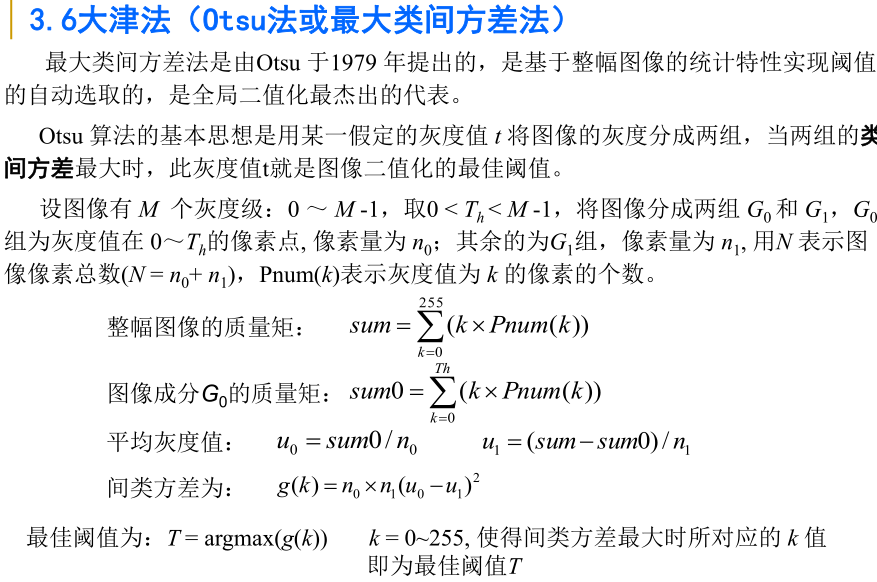
软件：基于matlab

**三、算法描述**

1. 迭代式阈值选择



1. 大津法



**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

Th=63.4091 Th=46.2959

1. 大津法



Th=64 Th=44

**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');

1. 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');