

# 数字图像处理作业 1

## 直方图匹配

### 实验报告

姓名 刘扬

学号 171850524

邮箱 [541446436@qq.com](mailto:541446436@qq.com)

联系方式 188 3289 9861

(南京大学 计算机科学与技术系, 南京 210093)

#### 1. 实现细节

直方图匹配的核心思想是以均衡化直方图图像为桥梁, 建立输入图像与均衡化图像的映射函数  $s=T(r)$  和模板图像与均衡化图像的映射函数  $s=G(z)$ , 之后计算由均衡化图像到模板图像的反函数  $z= G^{-1}(s)$ , 便可以实现由输入图像到模板图像的映射.

实现流程:

- (1) 读取输入图像和模板图像; 对于 RGB 三种颜色分别进行以下处理;
- (2) 用灰度值出现的频率作为概率, 计算输入图像和模板图像到均衡化图像的变换函数, 以表格的形式存储;
- (3) 使用(2)中得到的表格, 计算由输入图像到模板图像的变换函数, 同样以表格形式存储;
- (4) 对于输入图像的每一个灰度值, 通过查表得到变换后的灰度值.

核心函数:

##### (1) 直方图均衡化函数

此函数以灰度值矩阵作为输入, 以均衡化直方图(表格形式存储)作为输出.

`imread` 函数读取图像作为三维数组, 对应着 RGB 三种颜色的灰度值矩阵, 矩阵的长和宽分别对应着图像的长和宽, 直方图均衡化函数的输入即来源于此.

对于灰度值矩阵, 遍历矩阵存储每个灰度值出现的次数, 存储在 256 维向量 `r_table` 中.

计算转换函数, 对于原直方图中的灰度值  $k$ , 统计 `r_table` 中灰度值小于等于  $k$  的灰度值的次数和, 除以灰度值矩阵的元素总数, 再乘以 255 得到变换后的灰度值  $T(k)$ , 此处灰度值以浮点数形式存储, 以避免四舍五入的误差.

在 `matlab` 中的实现代码:

```

function s_table = histogram_equalization(gray_values)
[height, width] = size(gray_values);
L = 256;
r_table = zeros(1, L);
for i = 1 : height
    for j = 1 : width
        value = gray_values(i, j) + 1;
        r_table(1, value) = r_table(1, value) + 1;
    end
end
s_table = zeros(1, L);
s_table(1,1) = r_table(1,1);
for i = 2 : L
    s_table(1, i) = s_table(1, i-1) + r_table(1, i);
end
parameter = (L - 1) / (height * width);
for i = 1 : L
    s_table(1, i) = s_table(1, i) * parameter;
    % round() will lead to big error, which is obvious in the image
end
end

```

## (2)由输入图像到模板图像的灰度变换函数

此函数以上一个函数的输出作为输入，输入图像和模板图像通过上一个函数会得到均衡化的直方图 table1, table2.

对于输入图像的灰度值  $k$ ，通过查表 table1 得到其在均衡化图像中对应的灰度值  $s$ ，此处的  $s$  是浮点数.

对于每一个  $s$ ，在 table2 中查表找到与它最接近的  $s'$ ，再得到  $s'$  对应的  $k'$ ， $k'$  就是最终转换得到的灰度值.

在 matlab 中的实现代码：

```

% function r2z makes a transition from image_a to image_b
function trans = r2z(table1, table2)
trans = zeros(1, 256);
for i = 1 : 256
    value = table1(1,i);
    delta = 256;
    index = 2;
    for j = 1 : 256
        if abs(table2(1, j) - value) < delta
            delta = abs(table2(1, j) - value);
            index = j;
        end
    end
    trans(1, i) = index - 1;
end
end

```

## 2. 结果

### 2.1 实验设置

使用软件 MATLAB R2019a 编程

### 2.2 实验结果

以下 3 组图片为一组，每组依次是输入图像，模板图像，输出图像











