

北京交通大学数字图像处理实验报告

姓名：程维森

学号：21231264

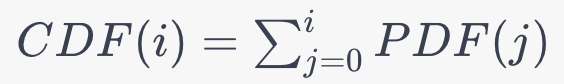
1. 直方图均衡化的数学推导

原始图像的累积分布函数（CDF）：

对于一个灰度图像，每个像素值的概率分布函数（PDF）可以用直方图表示。

直方图的累积分布函数（CDF）表示在某个像素值之前的概率总和。

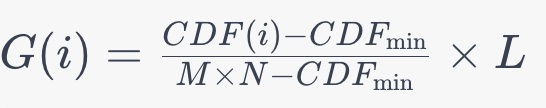
CDF 的计算公式为：



均衡化映射函数：

直方图均衡化的目标是将原始图像的像素值映射到一个新的像素值，使得新的像素值具有均匀分布。

均衡化映射函数 G(i) 的计算公式为：



其中，

M×N 是图像的总像素数，L 是像素的最大灰度级别

映射函数的**应用**：

将均衡化映射函数应用到原始图像的每个像素值，得到均衡化后的像素值。

新的像素值取整数部分，即

new\_pixel\_value = 

通过这个过程，原始图像的像素值被映射到一个具有均匀分布的新像素值集合。这样做的结果是增强了图像的对比度，使得图像中的细节更为突出

1. 直方图均衡化的代码解释

以下是代码的主要逻辑：

parse\_bmp 函数：

打开 BMP 文件并读取文件头部（前 54 个字节），提取图像的宽度、高度和每像素位数等信息。

计算每行像素的字节大小，考虑到可能的填充。

跳过文件头，移动到像素数据的起始位置。

逐行读取像素数据，考虑行填充。

calculate\_histogram 函数：

遍历像素数据，统计每个像素值的出现次数，得到图像的直方图。

equalize\_histogram 函数：

计算图像的累积分布函数（CDF）和直方图均衡化映射。

应用直方图均衡化映射到原始像素数据，更新像素值。

plot\_histogram 函数：

使用 matplotlib 绘制直方图图像，显示处理前后像素值的分布情况。

save\_bmp 函数：

将处理后的图像数据保存为 BMP 文件。

接下来的代码部分执行了以下操作：

使用 parse\_bmp 解析 BMP 文件，获取图像信息和像素数据。

通过 calculate\_histogram 计算原始图像的直方图。

使用 equalize\_histogram 进行直方图均衡化。

通过 calculate\_histogram 计算均衡化后图像的直方图。

使用 save\_bmp 将均衡化后的像素数据保存为 BMP 文件。

使用 matplotlib 绘制并保存原始直方图和均衡化后直方图的条形图

实验代码在附件中以“ipynb”的python格式展现

1. 实验结果

