



中山大學
SUN YAT-SEN UNIVERSITY

《数字图像处理》

实验报告

实验题目 灰度图像的缩放和灰度级量化

年 级 15 级

学 院 数据科学与计算机学院

学 号 15331044

学生姓名 陈再兴

2017 年 9 月 21 日

一、实验目的

- 1. Scaling，利用双线性插值根据给定的图像大小对特定的灰度图像进行缩放，并输出操作后的图像。
- 2. Quantization，根据给定的灰度级别对灰度图进行灰度值量化，并输出操作后的图像

二、实验环境

- 1. 操作系统：windows 10
- 2. 语言：c++
- 3. 编译器：visual studio 2015
- 4. 第三方库：OpenCV

三、实验原理

1. PNG 图片格式

- a) PNG 图片主要由 PNG 文件标识和各种数据块组成，而其中数据块又分为关键块（critical chunk）和辅助块（ancillary chunk）。每个数据块都要按照规定的格式排列数据。
- b) 数据块格式

名称	字节数	说明
Length (长度)	4 字节	指定数据块中数据域的长度，其长度不超过 $(2^{31}-1)$ 字节
Chunk Type Code (数据块类型码)	4 字节	数据块类型码由 ASCII 字母(A-Z 和 a-z)组成
Chunk Data (数据块数据)	可变长度	存储按照 Chunk Type Code 指定的数据

CRC (循环冗余检测)	4 字节	存储用来检测是否有错误的循环冗余码
--------------	------	-------------------

c)4 个关键数据块

- 文件头数据块 IHDR (header chunk)：包含有图像基本信息，作为一个数据块出现并只出现一次。
- 调色板数据块 PLTE (palette chunk)：必须放在图像数据块之前。
- 图像数据块 IDAT (image data chunk)：存储实际图像数据。
PNG 数据允许包含多个连续的图像数据块。
- 图像结束数据 IEND (image trailer chunk)：放在文件尾部，表示 PNG 数据流结束
- 其中 IDAT 储存了 PNG 图像的像素信息。

2.PNG 图片读入、处理和输出

- 从原理上说，可以利用 c++ 文件流将 PNG 图片以二进制文件方式逐字节将图像信息分数据块读入到特定的数据结构中。然后根据 PNG 设定的图像格式，找到 IDAT 数据块，并对其中的图像像素信息进行处理。处理完之后再将各个数据块的信息按 PNG 文件格式顺序输出到新的图像文件中即可。
- 实际上，在此项目中，是利用了第三方库 OpenCV 中的 API 来进行图片的读入和输出。这样做主要是因为，PNG 图片与 BMP 图像不同，PNG 图片的像素储存块（即 IDAT 中的数据块数据部分）并不是直接储存每个像素的像素值的。而是利用了特定的压缩算法将像素值进行压缩再进行储存的，要操作像素值必须先解压这个数据部分，才能进行操作。而这个解压缩的算法以及数据块数据和 CRC 块之间的关系我尚未弄明白，所以还是必须要借助第三方库来进行图像的读取和输出。

3.双线性插值算法

- a) 双线性插值是线性插值的进阶版本，线性插值只是在某个方向上进行插值，而双线性插值则是在 x 、 y 方向上都计算插值。
- b) 在图像处理中应用插值算法可以用来计算放缩后图像的像素值。要求新图像中某个像素的像素值，首先利用空间变换，求得该像素在原图像中的对应像素点坐标，然后找到这个对应像素周围最近的四个点的像素，利用这四个像素点的值和双线性插值公式就可以算出新图像中该像素点的像素值。

- c) 双线性插值公式，设 (x, y) 为新图像像素对应的原图像的像素坐标， (x_1, y_1) ， (x_2, y_1) ， (x_1, y_2) 和 (x_2, y_2) 为最邻近的四个像素点的坐标， Q_{11} ， Q_{21} ， Q_{12} ， Q_{22} 。

$$\begin{aligned} f(x, y) &\approx \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2) \\ &= \frac{y_2 - y}{y_2 - y_1} \left(\frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \right) + \frac{y - y_1}{y_2 - y_1} \left(\frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \right) \\ &= \frac{1}{(x_2 - x_1)(y_2 - y_1)} (f(Q_{11})(x_2 - x)(y_2 - y) + f(Q_{21})(x - x_1)(y_2 - y) + f(Q_{12})(x_2 - x)(y - y_1) + f(Q_{22})(x - x_1)(y - y_1)) \\ &= \frac{1}{(x_2 - x_1)(y_2 - y_1)} \begin{bmatrix} x_2 - x & x - x_1 \end{bmatrix} \begin{bmatrix} f(Q_{11}) & f(Q_{12}) \\ f(Q_{21}) & f(Q_{22}) \end{bmatrix} \begin{bmatrix} y_2 - y \\ y - y_1 \end{bmatrix}. \end{aligned}$$

- d) 原图像对应坐标的计算方法。首先根据原图像的大小和设定的新图像的大小计算出放大比例。因为缩放不一定是等比例缩放，所以长和宽的缩放比例应该分开计算。然后根据长宽缩放比例就可以将原图像对应像素点坐标求出来。（具体求法根据缩放比例的求法不同而略有不同，假设长的放缩比例是利用原图像长/新图像长，则原图像对应点的横坐标就应该用新图像像素点的坐标乘以放缩比例，如果是利用新图像长/原图像长，则原图像对应点的横坐标应该用新图像像素点的坐标除以缩放比例，纵坐标的计算同理。）

- e) 四个最邻近像素点的计算方法。一般计算获得的原图像像素坐标可能有以下几种情况，设 $x_1 < x_2, y_1 < y_2$ 。

·求得的 x, y 为小数，则 x_1, x_2 分别为 x 向下和向上取整的整数， y_1, y_2 分别为 y 向下和向上取整的整数。

· x 为整数， y 为小数，则 $x_1 = x, x_2 = x + 1$ ； y_1, y_2 分别为 y 向下和向上取整的整数。

- x 为小数, y 为整数, 与上面第二种情况类似
- x, y 均为整数, 则直接令新图像中 (x, y) 点的像素值等于原图像中 (x, y) 点的像素值
- 新图像坐标变换为原图像坐标就是以上四种情况, 需要特别注意的就是变换后的原图像坐标位于最高一行 (y 最大) 和最右一行 (x 最大) 的时候需要防止 x2 和 y2 越界。

4.图像量化

- a) 简单点来说把连续的数据变成离散的数据就称为量化。而这里的灰度图像量化其实则是把原有图像的灰度级别降低 (即降低量化级别)。令落在某个范围的所有值等于一个固定的值, 这样就能把灰度级降低。
- b) 具体的量化方法就是对原图像中每个像素点的像素进行范围判断, 然后根据这个点所处的范围赋予新图像中对应像素点一个灰度值。量化并不会改变图像的大小。
- c) 在这次实验中设定灰度的数值表示不变, 即 0 表示黑色, 255 表示的是白色。变化的是灰度数值的取值范围 S, 如灰度级别为 256 时, $S=\{0,1,2,3, \dots, 255\}$, 灰度级别为 128 时, 则 $S=\{1,3,5,\dots, 255\}$ 。
- d) 灰度值所处范围的计算方法。这个是根据给定的灰度级别, 然后用 256 除以这个灰度级别, 求出每个范围的大小, 然后设定各个范围内的值对应的新的灰度值。如灰度级别为 4, 则每个范围大小为 64, $S=\{63, 127, 191, 255\}$ 。当原图像的灰度值落在 (0,63) 时, 则新图像中对应的灰度值应为 63, 其他范围的值以此类推。

四、实验内容

1.Scaling

- a) Down-scale to 192*128 (width: 192, height: 128), 96*64, 48*32, 24*16 and 12*8, then manually paste your results on the report



192*128



96*64



48*32



24*16



12*8

- b) Down-scale to 300*200, then paste your result.



300*200

- c) Up-scale to 450*300, then paste your result.



450*300

- d) Scale to 500*200, then paste your result.



500*200

2.Quantization

- a) Reduce gray level resolution to 128; 32; 8; 4 and 2 levels, then paste your results respectively. Note that, in practice computers always represent “white” via the pixel value of 255, so you should also follow this rule. For example, when the gray level resolution is reduced to 4 levels, the resulting image should contain pixels of 0; 85; 170; 255, instead of 0; 1; 2; 3.



128



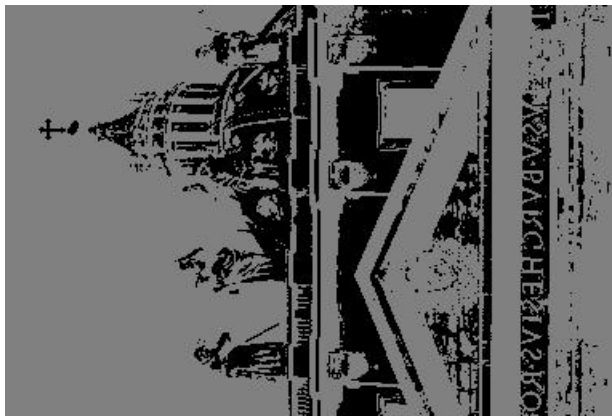
32



8



4



2

- b) Detailedly discuss how you implement the quantization operation, i.e., the “quantize” function, in less than 2 pages. Again, please focus on the algorithm part. Analyzing and discussing interesting experiment results are also welcomed, but please don't widely copy/paste your codes in the report

具体的量化方法就是对原图像中每个像素点的像素进行范围判断，然后根据这个点所处的范围赋予新图像中对应像素点一个灰度值。量化并不会改变图像的大小。

在这次实验中设定灰度的数值表示不变，即 0 表示黑色，255 表示的是白色。变化的是灰度数值的取值范围 S ，如灰度级别为 256 时， $S=\{0,1,2,3, \dots, 255\}$ ，灰度级别为 128 时，则 $S=\{1,3,5,\dots, 255\}$ 。

灰度值所处范围的计算方法。这个是根据给定的灰度级别，然后用 256 除以这个灰度级别，求出每个范围的大小，然后设定各个范围内的值对应的新的灰度值。如灰度级别为 4，则每个范围大小为 64， $S=\{63, 127,$

191,255}。当原图像的灰度值落在 (0,63) 时，则新图像中对应的灰度值应为 63，其他范围的值以此类推。

五、总结

1. 这次实验是第一次的数字图像处理实验，令我对数字图像处理的方法和图片格式有了一定程度上的了解。
2. 学习了第三方库 OpenCV 有关读取图片、操作图片和写入图片的 API 和数据结构