

数字图像处理 综合作业二

自 45 柳荫 2014011858

对课堂演示的 JPEG 编解码算法进行如下改进：

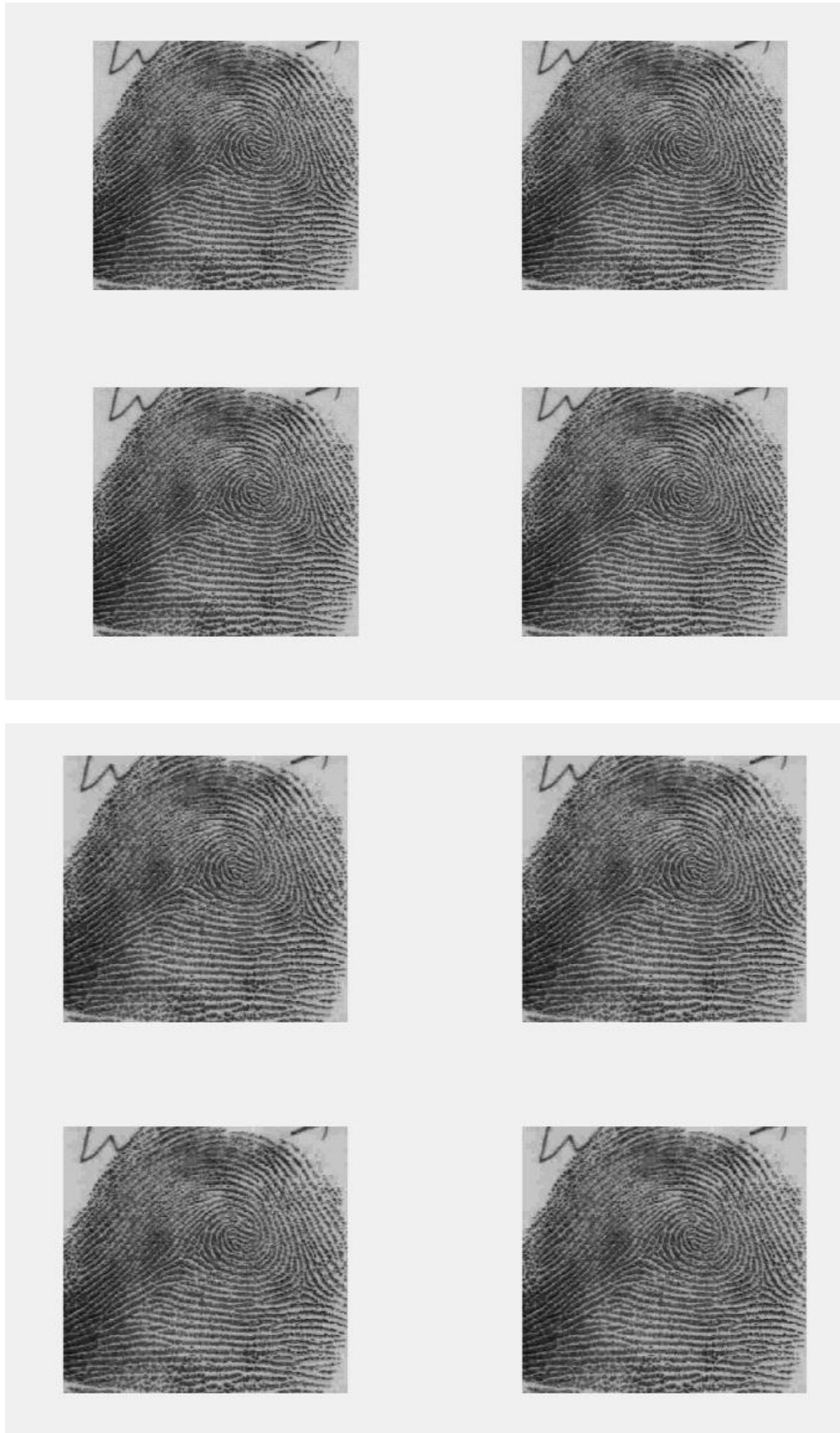
1. 利用 LPC (lossless predictive coding) 对所有 block 的直流成分进行编码，对预测误差进行霍夫曼编码；
2. 考虑断开的连续 0；
3. 改用固定的霍夫曼字典（利用给定图像库估计符号概率）。

在给定图像库上，测试不同改进版本的编解码算法，取 quality 为 1,5,10,20（用 quality 乘以 normalization 矩阵^②），分别计算压缩比、均方根误差 (rmse)。

解答思路与解答：

- 1, 对 LPC (无损预测编码) 而言，关键是像素的新信息被定义为实际值和预测值的差。存储的是预测误差 (compute error)。这里就设定当前数值大小 **预测** 为前一个数值大小。所以在预测误差矩阵等于 DC 分量减去右移一位后的 DC 分量。编码为 Huffman 编码，在所给示例程序中已有 huffman 编码，所以并不需要做更多改动。
详见程序 [im2jpeg_LPC](#)、[jpeg2im_LPC](#)。
- 2, 考虑断开的连续 0, 对于连续次数超过 1 次的 0, 设置两个参数存储，一个是标志 -1000, 意思是这里不存储数值，而是遇到了连续的 0；一个是连续的个数 count, 即从当前位置开始，有连续的多少个 0，这在程序注释中已写清楚。
Decode 就是 encode 的逆运算，当没有遇到 -1000 时，正常按照顺序恢复，当遇到 -1000 后，开始恢复 n 个 0，其中 n 为后一个参数的值。
详见程序 [im2jpeg_considerzeros](#)、[jpeg2im_considerzeros](#)。
- 3, 固定的霍夫曼字典与常规的方式基本相同，不同的是，训练结束后，开始计算累加后的 numel 和 prob, 将这一结果存储为固定值。之后再进行某一特定图像的压缩时，不需要存储最合于此图像的霍夫曼编码，从而减小了图片的数据量。
详见程序 [im2jpeg_fixeddict](#)、[jpeg2im_fixeddict](#)。

下图是第 1 张图 F0001 的结果，4 张图分别对应不同的 quality (1, 5, 10, 20)，每张图上的 4 个图按顺序又分别是用 lpc 编解码变换、考虑连续 0 的编解码变换、用固定 huffman 字典编解码变换、原始的编解码变换的图：





压缩比 cr 和方均根误差 rmse 被输出到一个 Excel 表格文件 a.xlsx 中，10 张图分别用 10 个 sheet 存储，从表格中可以看出，quality 越大，同等情况下 cr 越高，rmse 越大。另外，在 quality 比较低的情况下，4 种编解码算法中，固定 huffman 字典的 cr 比较高，而其他算法较低；在 quality 比较高的情况下，考虑连续 0 的算法的 cr 比较低，其他算法都较高。
数据详见 [a.xlsx](#)。

实验感悟：

这次实验相较于第一次综合作业更加抽象，由于设计多个算法文件，一些函数的意义不甚清楚的情况下，对其改进难度不小；实际上静下心来慢慢去搜索各个函数的意义就知道它的含义了，得花不少时间。

这次实验让我对图像压缩 jpeg 算法有了更多的理解，对 matlab 编程也有了更深的领悟。