

JPEG 静图像压缩实验

PB20061372 朱云沁 Dec. 8, 2023

1 实验目的

- 了解数字图像压缩的一类方法.
- 掌握 2D 快速 DCT 变换的算法.

2 实验原理

JPEG 标准是面向连续色调静止图像的图像压缩标准. 它定义了多种类型的工作模式, 其中最基本的是基于 8×8 块的 DCT 变换的顺序编码. 就是将一帧图像分为 8×8 的数据块单元, 按照从左至右、自上而下的顺序对“块流”, 其编码、解码的框图如下:

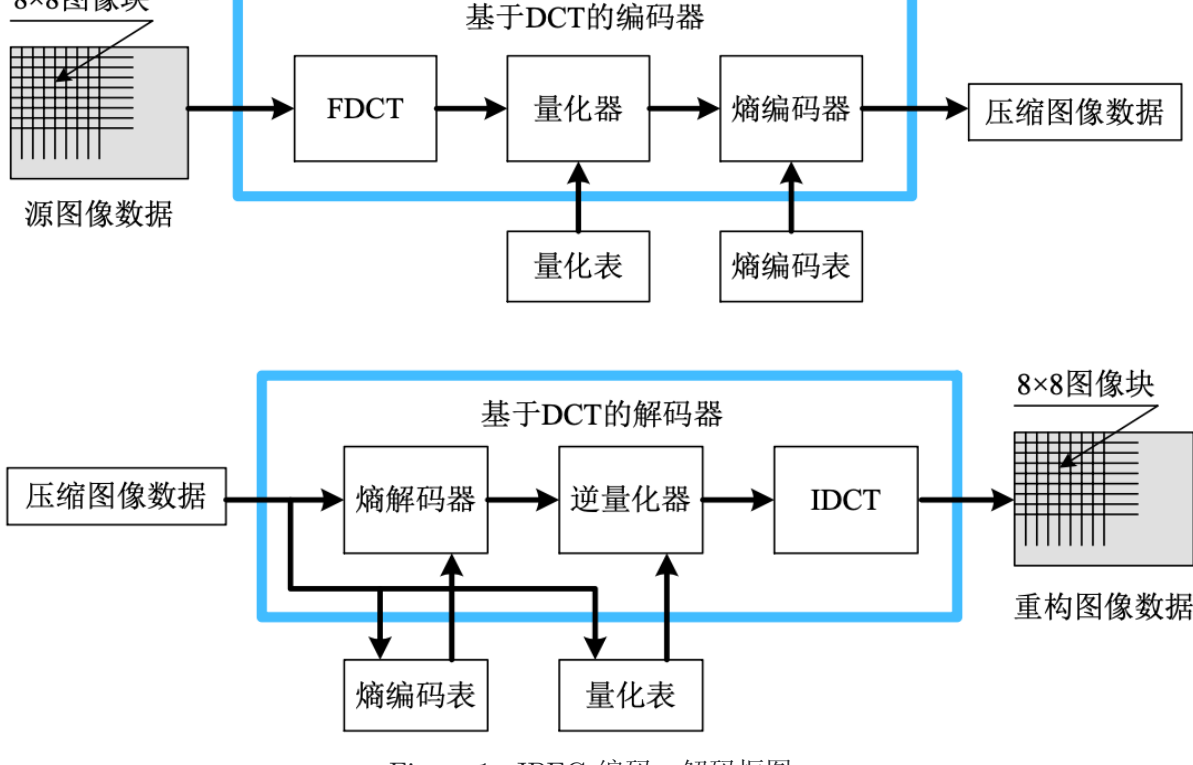


Figure 1. JPEG 编码、解码框图.

2.1 DCT

对于一个 8×8 的图像数据块, 表示为一个矩阵 f , 计算相应的 DCT 系数 F 的公式如下:

$$F_{uv} = \frac{1}{4}C(u)C(v)\sum_{i=0}^7\sum_{j=0}^7f_{ij}\cdot\cos\left(\frac{(2i+1)u\pi}{16}\right)\cdot\cos\left(\frac{(2j+1)v\pi}{16}\right), \quad u,v=0,1,\cdots,7$$

其中, $C(u)$ 和 $C(v)$ 是缩放因子. $C(0) = \frac{1}{\sqrt{2}}$. 对于 $u > 0$, $C(u) = 1$. 为了实现 2D 快速 DCT 变换, 使用分步的策略, 在水平和垂直方向上分别应用一维 DCT 变换.

出于惯例, 在 DCT 之前, 将 $[0, 255]$ 范围内的各像素值减去 128.

2.2 量化

本实验中, AC、DC 量化的表达式分别为:

$$G_{00} = \text{round}\left(\frac{F_{00}}{Q_{00}}\right), \quad G_{ij} = \text{round}\left(\frac{F_{ij} \times 8}{Q_{ij} \times \mathbf{g_scale}}\right), \quad (i,j) \neq (0,0)$$

其中, 矩阵 Q 是量化表, G 是量化后的 DCT 系数. $\mathbf{g_scale}$ 是参数. 本实验考虑亮度和色度分量的量化表如下:

$$Q_Y = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}, \quad Q_C = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$

2.3 熵编码

对量化后的 DC、AC 系数分别进行熵编码.

- DC 系数:** 差分 + Huffman

- 以 0 为初始值, 对相邻块的 DC 系数作差分, 得到 ΔG_{00} .
- 将差分值改写为 (size, amplitude) 的形式, 其中 size 表示 amplitude 的二进制长度.
- 按照 DC 系数表对 size 采用 Huffman 编码, 按照正负值表将 amplitude 表示为二进制数.

- AC 系数:** 行程 + Huffman

- 以块为单位, 对 AC 系数进行 Zigzag 扫描, 记下非零值的大小和其连续零的个数.
- 将结果改写为 (run-length, size) (amplitude) 的形式, 其中 run-length 表示非零值前连续零的个数. 以 EOB 标识一个块的扫描结果. 以 EOB 标识一个块的扫描结果.
- 按照 AC 系数表对 (run-length, size) 采用 Huffman 编码, 按照正负值表编码 amplitude.

- 字节填充: 在连续出现 0xFF 时, 在其后插入 0x00, 防止 0xFF 被解码为标记符.

3 实验内容及结果

按照如上框图实现一帧图像的压缩编码.

3.1 实现细节

实验在 M1 芯片 macOS 下进行, 使用 Python 3.10.10 编写, NumPy 版本为 1.23.2. 代码结构:

- ./tables.py: Zigzag 索引和所有码表.
- ./main.ipynb: JPEG 编码实现及实验数据分析.
- ./demo/: 包含 C++ 示例程序和输入输出. 示例程序使用 Apple Clang 15.0.0 编译.

3.2 JPEG 编码程序

压缩前, 总字节数为 65536. 选定量化表为 Q_Y , 参数 $\mathbf{g_scale} = 8$. 经过 C++ 示例程序 JPEG 编码后, 总位数为 44487, 总字节数为 5561. 经过 NumPy 实现的 JPEG 编码后, 总位数为 44460, 总字节数为 5558. 原图像、C++ 示例程序压缩图像、NumPy 实现压缩图像比较如下:



Figure 2. 原图像、C++ 示例程序压缩图像、NumPy 实现压缩图像.

C++ 示例程序与 NumPy 实现的压缩图像存在细微差别, 如下图:

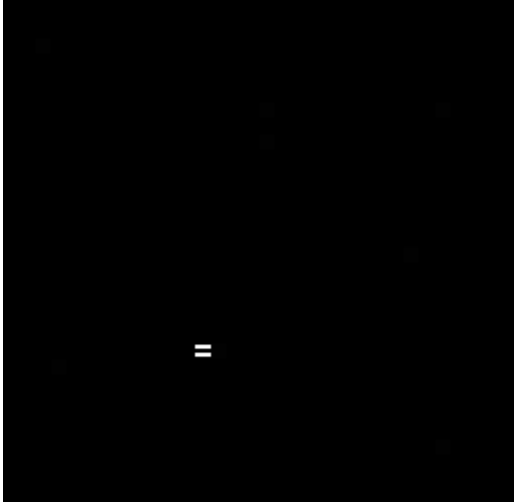


Figure 3. C++ 示例程序与 NumPy 实现的压缩图像的差值的绝对值.

经对照, 发现系 DCT 及量化步骤双精度浮点数舍入误差所致. 两者的平均绝对误差 $\text{MAE} = 0.14$, 均方根误差 $\text{RMSE} = 0.19$, 均可忽略不计, 因此判断 NumPy 实现程序正确.

4 思考题

- 计算图像的压缩比, 并比较原图像的效果.

定义压缩比为原图像总字节数与压缩后图像总字节数的比值. 为了度量压缩后的图像质量, 本实验使用 MAE、RMSE、PSNR 三种指标. 其中, PSNR 表示压缩后图像与原图像的峰值信噪比, 其定义为:

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}}$$

结果如下表所示

	Ratio	MAE	RMSE	PSNR
C++ 示例程序	11.7849	106.3914	4.6069	34.8627 dB
NumPy 实现	11.7913	105.9499	4.6065	34.8635 dB

压缩效果如图 2 所示. 相较原图像, JPEG 压缩图像整体感知效果仍然较好, 但出现锯齿状的边缘和模糊的细节. NumPy 实现各指标均优于 C++ 示例程序, 可能是由于计算精度更高.

- 改变 $\mathbf{g_scale}$ 和量化矩阵中的元素, 比较压缩比和恢复图像的效果.

改变 $\mathbf{g_scale} \in [1, 30]$, 量化矩阵分别设置为 Q_Y 和 Q_C , 比较恢复图像和各项指标, 如图:



Figure 4. 选定 Q_Y , 不同 $\mathbf{g_scale}$ 的恢复图像 (从左至右: $\mathbf{g_scale} = 1, 5, 10, 15, 20, 25, 30$).



Figure 5. 选定 Q_C , 不同 $\mathbf{g_scale}$ 的恢复图像 (从左至右: $\mathbf{g_scale} = 1, 5, 10, 15, 20, 25, 30$).

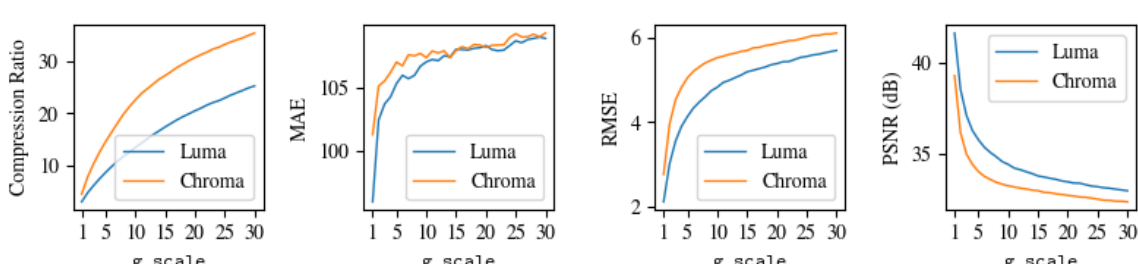


Figure 6. $\mathbf{g_scale} \in \{1, 5, 10, 15, 20, 25, 30\}$, 量化矩阵为 Q_Y (Luma) 和 Q_C (Chroma) 时的压缩指标.

可见, 随着 $\mathbf{g_scale}$ 的增大, AC 系数的量化精度降低, 量化后的图像质量下降, 但压缩比提高. 色度量化矩阵 Q_C 的恢复图像质量较差, 是由于考虑到色度分量的变化对人眼的敏感度较低, 量化表中的各元素值较大.

- DCT 变换, 量化后, 数据的值域是多少? 有没有可能超出码表所给幅值的范围?

经过平移, 图像像素值 $f_{ij} \in [-128, 127]$, 计算得 DCT 系数满足

$$F_{00} = \frac{1}{8} \sum_{i=0}^7 \sum_{j=0}^7 f_{ij} \in [-1024, 1016],$$

$$F_{uv} \in [-1020, 1020], \quad (u,v) \neq (0,0)$$

查得量化表范围为 $Q_{00} \in \{16, 17\}$, $Q_{uv} \in [10, 121]$, 又规定 $\mathbf{g_scale} \in [1, 30]$, 因此

$$G_{00} = \text{round}\left(\frac{F_{00}}{Q_{00}}\right) \in [-64, 64],$$

$$G_{ij} \in \left[\text{round}\left(\frac{-102 \times 8}{\mathbf{g_scale}}\right), \text{round}\left(\frac{102 \times 8}{\mathbf{g_scale}}\right) \right], \quad (i,j) \neq (0,0)$$

码表所给 DC 位长至多为 11, 幅值范围为 $[-2047, 2047]$, 因此 DC 不可能超出. AC 位长至多为 10, 幅值范围为 $[-1023, 1023]$, 因此 AC 也不可能超出.