

作业一：无损编码/压缩算法实验

梅佳伟 221050366

表 1: 图像矩阵

1	1	1	2
1	1	2	2
5	5	5	7
0	0	7	7

上表为一个灰度级在 $0 \sim 7$ 之间的二维图像。记： $a=1, b=2, c=5, d=7, f=0$ 。

先编写 Huffman 编码算法，再使用行程编码（分别使用课堂上讲解的二种排列方式），对图像进行压缩。计算原图 C1 和压缩后图像 C2 的尺寸，计算压缩比 $(C1/C2)$ ，并比较分析。

1 算法原理

Huffman 编码为统计一定文本中字符出现次数，并依此建树的算法。字符使用叶结点表示且出现次数较少的字符将排在树的远离根节点的位置，使用二叉树，左侧分支选 0，右侧选 1 即可完成编码。由于二叉树可旋转左右子节点，此编码方式并不唯一，需要额外存储树信息且在各字符分布概率相等的情况下效果最差。行程编码为统计临近重复个数的编码方式，可有效降低重复。

2 实验结果

Huffman 编码结果： $\{a : 11, b : 01, c : 101, d : 00, f : 100\}$

1. 直角 S 型展开

展开结果: 1, 1, 1, 1, 5, 5, 5, 2, 1, 2, 2, 7, 7, 7, 0, 0

行程编码结果: (4, 11), (3, 101), (1, 01), (1, 11), (2, 01), (3, 00), (2, 100)

压缩编码位数: $3 \times 7 + 2 + 3 + 2 + 2 + 2 + 2 + 3 = 37\text{bits}$

混合编码压缩率: $(3 \times 16) \div 37 = 1.2973$

2. 45° 角 S 型展开

展开结果: 1, 1, 1, 5, 1, 1, 2, 2, 5, 0, 0, 5, 2, 7, 7, 7

行程编码结果: (3, 11), (1, 101), (2, 11), (2, 01), (1, 101), (2, 100), (1, 101), (1, 01), (3, 00)

压缩编码位数: $2 \times 9 + 2 + 3 + 2 + 2 + 3 + 3 + 3 + 2 + 2 = 40\text{bits}$

混合编码压缩率: $(3 \times 16) \div 40 = 1.2000$

3 结果分析

可以发现直角 S 型展开方式压缩率在此实验中效果优于 45° 角 S 型展开。原因在于成矩形团状的像素具有相同像素值时，直角 S 型展开可有效将其串在临近位置，而 45° 角 S 型展开会穿越该直角边缘。45° 角 S 型展开在存在较多 45° 直线边缘的情况下效果会更好。

除此之外，我在独立使用哈夫曼编码压缩次图像矩阵时获得的压缩率也为 1.2973 左右。暗示也许图片尺寸再大一些时，这种混合使用的压缩效率会更高。