**灰度图像的Huffman编解码**

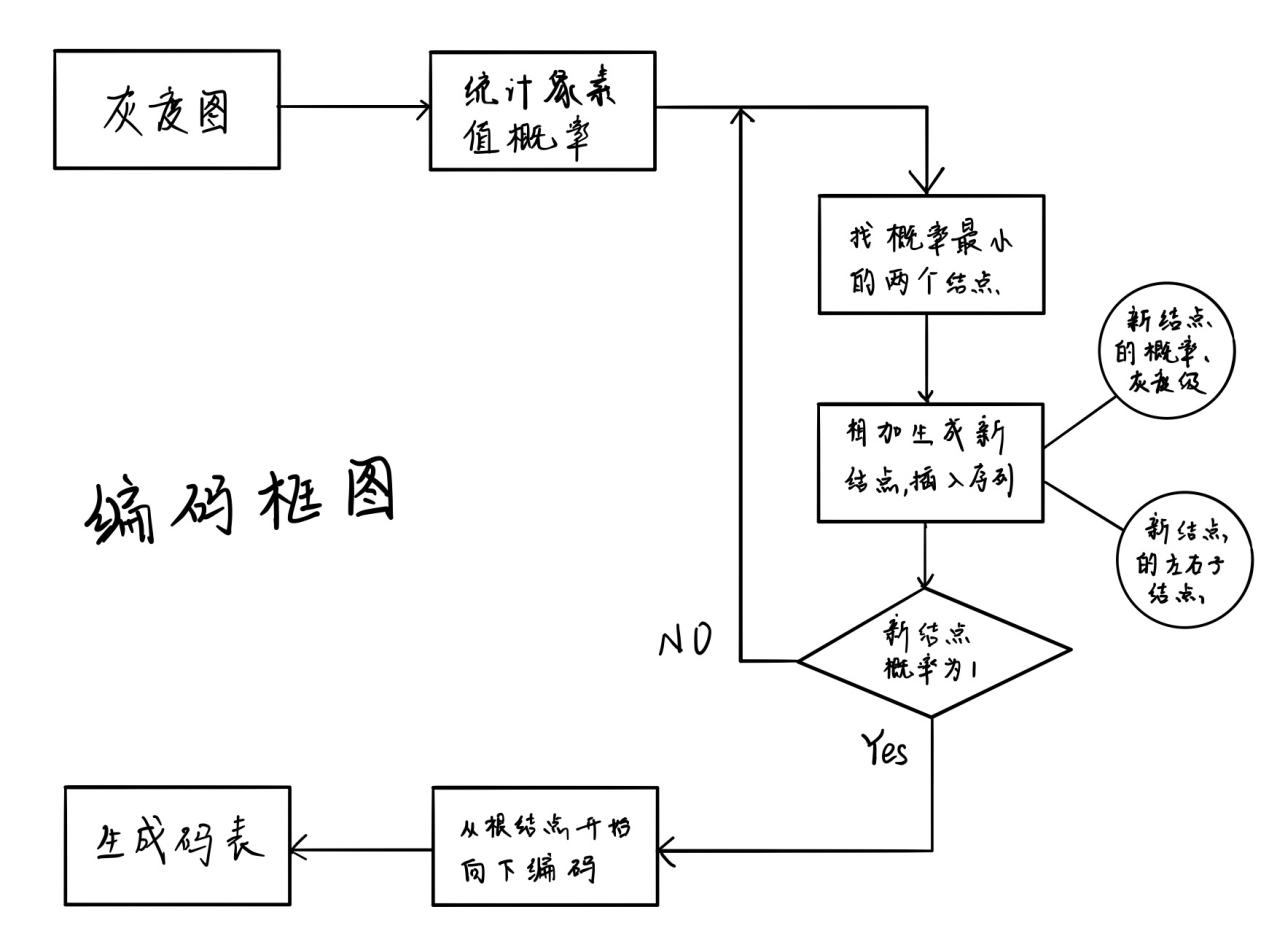
1. 作业目的

掌握Huffman编解码原理

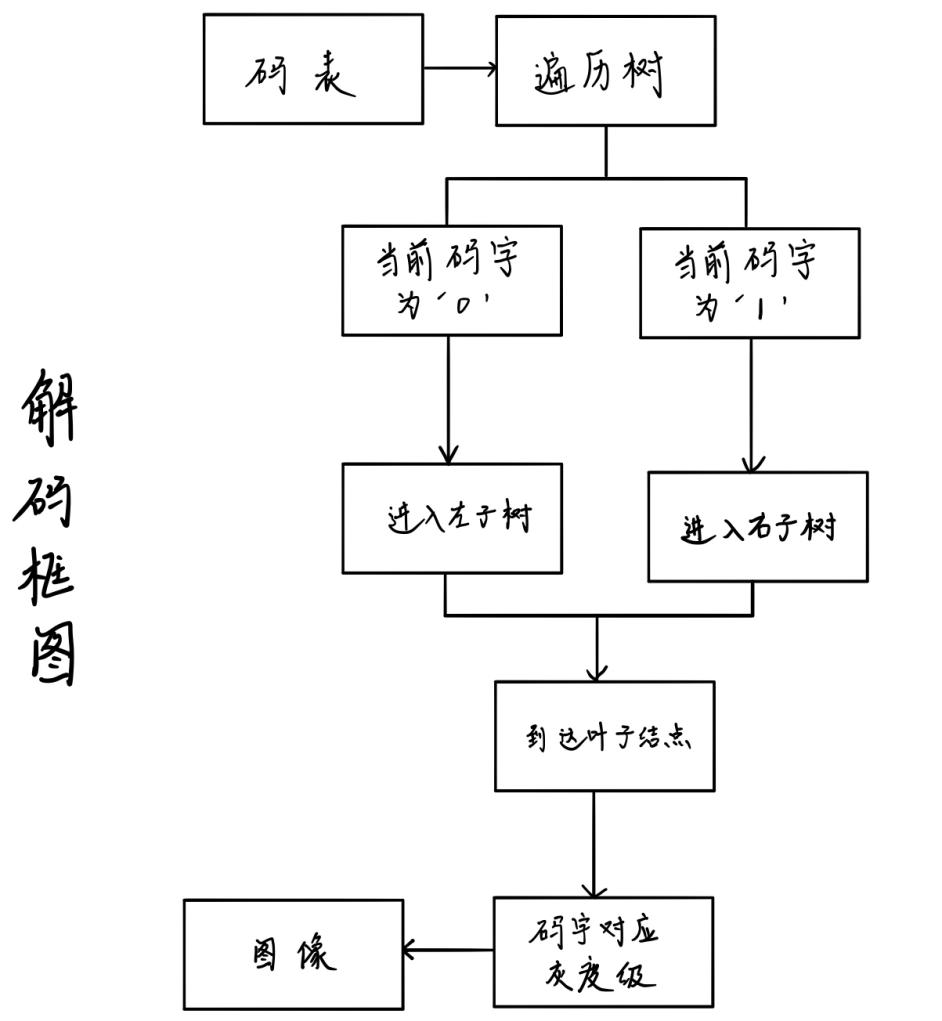
1. 作业内容

将一副给定的灰度图像进行Huffman编解码

1. 方法描述
   1. 编码
      1. 定义结构体：存储树节点的灰度级、出现概率、左右孩子节点指针、当前编码、是否已经加入到树中。
      2. 建树：每次需要找到剩余节点中概率最小的两个节点，将它们的概率相加形成新的节点加入进去，记录这个新节点的左右子节点为这两个点。循环这个过程，直到最终概率加到1。记录根节点的序号。
      3. 对建好的树进行编码，左节点为0，右节点为1。记录灰度值对应的码字。
      4. 通过每个点的像素值，输出对应的码字。于此同时计算熵和平均码长。



* 1. 解码
     1. 读取码表，即encode.txt文件
     2. 根据编码遍历树，遇到0进入左节点，遇到1进入右节点。
     3. 将编码对应的灰度级赋值给解码图像。



* 1. 计算编码效率
     1. 码率=熵/平均码长

1. 实验结果
   1. 编码表：

保存在encode.txt文件

* 1. 解码图像：

保存为decode.bmp图像



* 1. 码率：

熵为：4.910856 平均码长为：7.113792 码率为：0.690329

* 1. 程序源代码

在“工程”文件夹下

1. 附录

参考文献：

1. Huffman实现图像编码解码(<https://blog.csdn.net/YangHeng816/article/details/80538825>)
2. Huffman编码解码(<https://blog.csdn.net/q547550831/article/details/51589278>)
3. Huffman树——编解码(<https://www.cnblogs.com/vfdxvffd/p/11622261.html>)