**网络多媒体 课程设计**

|  |  |
| --- | --- |
| 成  绩 |  |

题 目：**JPEG图像编解码**

**主讲教师： 沈沛意**

**学院班级： 1613012 云**

**学 号： 16130120190**

**姓 名： 任晓路**

**日 期： 2019.6.10**

一、编码原理

JPEG是第一个国际图像压缩标准，能够在提供良好的压缩性能的同时，具有比较好的重建质量，被广泛应用于图像、视频处理领域。

1. 算法框架：

① 使用离散余弦变换DCT把空间域表示的图变换成频率域表示的图。

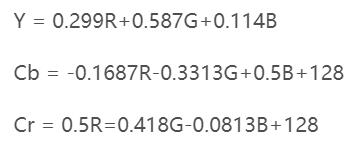
② 使用加权函数对DCT系数进行量化，这个加权函数对于人的视觉系统是最佳的。

③ 使用霍夫曼可变字长编码器对量化系数进行编码。

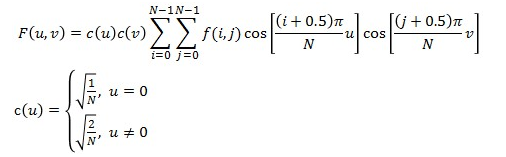
④ 译码或者叫做解压缩的过程与压缩编码过程正好相反。

2. 算法流程：

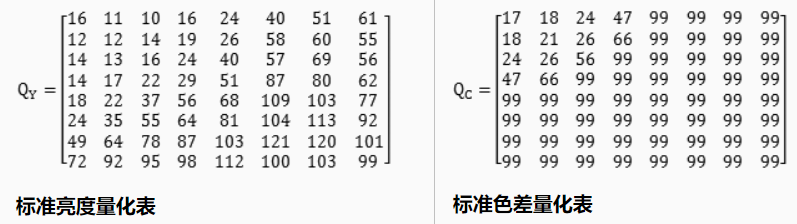
①变域：由于JPEG图像采用YUV颜色空间，而BMP图像一般为RGB颜色空间，故需要在压缩前对BMP图像进行颜色空间转换。公式如下：



②变换：把转化为YUV颜色空间后得到的三维矩阵（前两维为二维色彩矩阵，最后一维为三种颜色分量）分割为三个对应颜色分量的二维矩阵，进一步分为多个8\*8的块f(i,j)进行二维DCT变换，生成多个DCT变换系数矩阵F(u,v)。通过将色彩分量转化频率空间，达到方便后续舍入处理的目的。二维DCT变换公式如下：



③量化：通过减少JPEG压缩图像所需要的位数达到压缩目的。它用每个频率除以一个整数，然后取整得到包含很多0的8\*8压缩矩阵。量化是JPEG中产生信息丢失的主要原因，因为牵扯到大数化小和四舍五入必然导致后续的解码丢失信息。所用的符合人类心理学最佳量化矩阵表如下：



量化公式为：

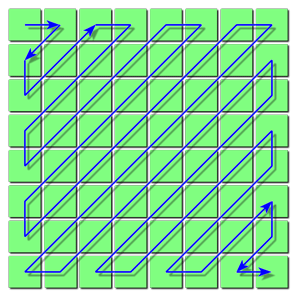


在量化完之后大量的有规律排序的数值0集中在右下角，非常有利于进一步压缩。

④编码：

a)对DC分量的编码：直流分量是一个块中的主要成分，做完DCT变换一定是数值最大的那个，将它单独抽出做DPCM编码会有比较好的效果。

b)对AC分量的编码：先通过Z形扫描将二维矩阵转化为一维，毕竟数据传输也是流式一维传输，然后对其中长连0进行压缩表示。以下为Z形编码的示例：

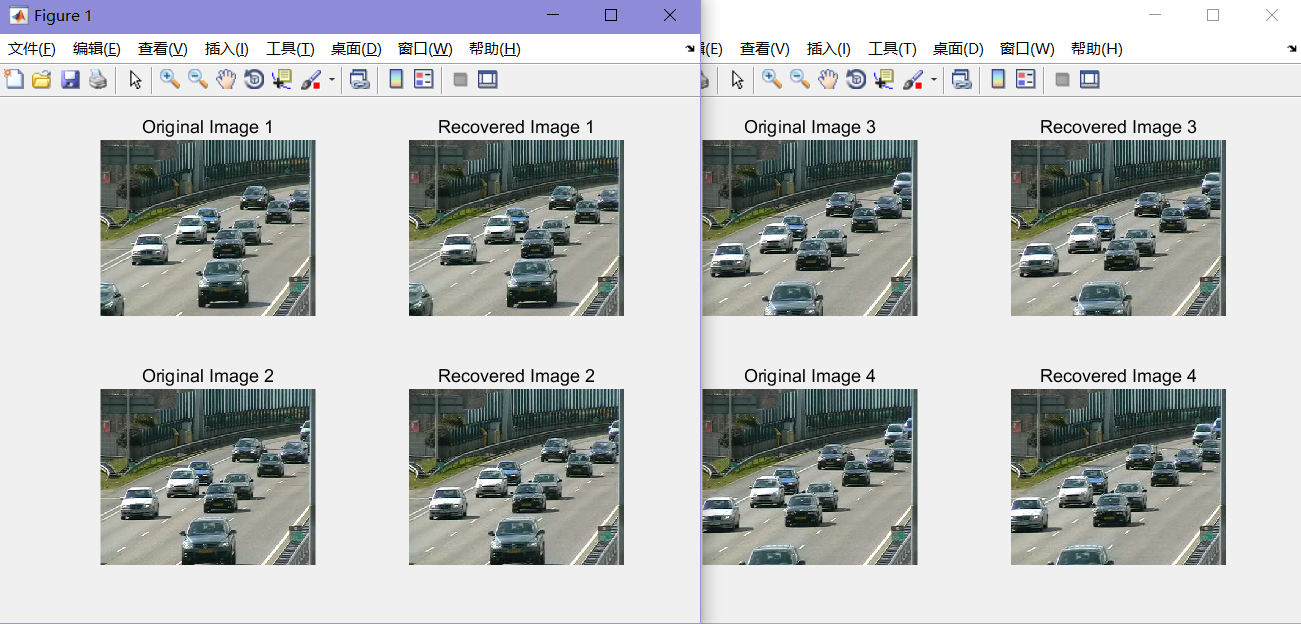


c)采用Huffman编码技术进一步压缩编码过的DC和AC分量，使得传输数据量达到最小。

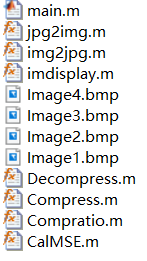
⑤解码：解码过程与编码过程互逆。

二、项目分析：

1.项目介绍：



该项目是一个对一组连续图像进行JPEG编解码的项目，通过对压缩过程中产生的中间数据进行记载和数据运算，从而直观地分析其压缩率、均方误差和时间等性能问题，深入了解JPEG压缩原理。

2.项目架构：

本项目由matlab实现，所有的代码如下：

①main：负调用各个处理流程，输出最后结果。

②Compress：负责实现JPEG压缩算法。

③img2jpg：由Compress调用，负责z形数据读取，加入EOF

④Decompress:负责实现JPEG解压缩算法。

⑤jpg2img:由Decompress调用，负责重排矩阵，逆DCT变化。

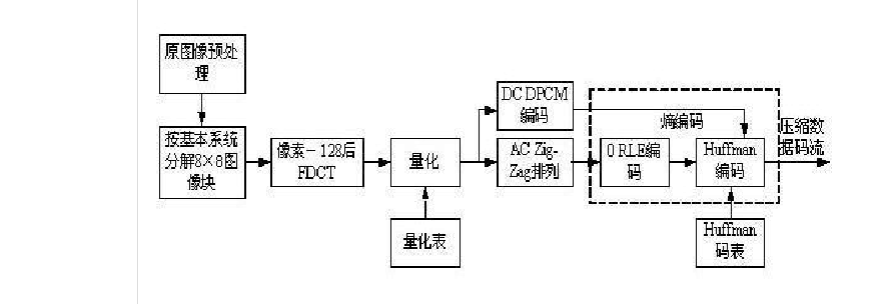
⑥Compratio:负责计算压缩率

⑦CalMSE：负责计算均方差，评估数据损失

⑧imdisplay：负责展示压缩结果

⑨Image1234：四张原数据图片

3.算法流程图：

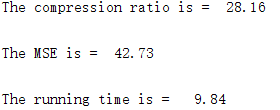


算法大体同第一部分所述，细小变化为由于本项目不牵扯数据传输，故没有使用哈夫曼编码将数据进一步压缩，而是z形编码后直接就开始进行反变换，即图中的虚线框部分并没有编码实现。

这里的图像只是压缩部分的流程图，解压缩流程图与这部分呈镜像关系，故不再赘述。

4.编解码性能分析：

正如源代码架构分析处所述，有两个m文件专门负责计算压缩率和均方误差。同时在代码中还嵌入了计算起始时间的代码，最后得到的结果如下：



①可以看出压缩率达到了28.16，这个数据是由原数据长度除以z形读取矩阵后再压缩的数据长度得到的结果。

②均方误差42.73，通过计算欧式距离得到的数据，对于原图像352x288的分辨率来说这个值不是很大。

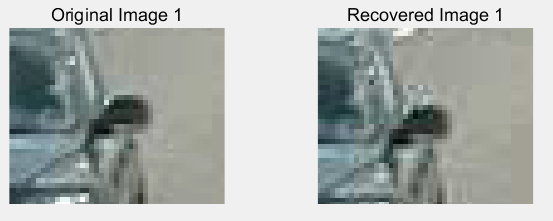
③运行时间四张图片平均每张2.5秒左右，较一般的压缩软件较慢。但由于是运行在matlab上，也会有一定的性能损失，处于可以接受的范围之内。

三、噪声分析：

1.量化噪声：

①正常压缩率：

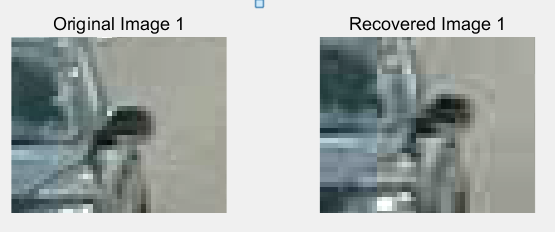
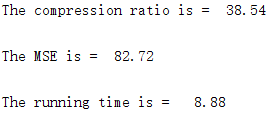
量化噪声是JPEG图片压缩中最主要的问题，前面我们已经通过均方误差分析过量化误差。我们可以通过查看图像感性地分析噪声：



可以看出基本差异不明显，噪声影响很小。

②二倍压缩率：

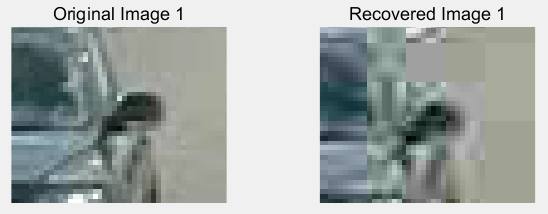
通过肉眼比对我发现在上一步那么高的压缩率下量化噪声依然不明显，所以我决定增大压缩率再做对比，查看噪点变化：



通过将亮度和色差矩阵数值翻倍后的结果如上图。压缩率并没有如预想的翻倍，而是有一种边际效应递减的趋势。压缩和解压缩时间反倒减少，预计是因为解压缩所需的运算量因压缩率变大、舍入值变多而变少导致的。

同时，通过放大对比原图像和压缩图像，在压缩后的图像中开始出现明显的方块，这是因为压缩率过大和算法特性导致相邻的8\*8矩阵之间产生了亮度和色差的信息割裂。

③四倍压缩率：



在此压缩率下我们可以发现图像基本已经损失了大量细节信息，整体效果十分糟糕。

2.其他噪声：

其他噪声对于JPEG压缩算法的影响很小，但不能说完全没有。比如原图像本身的量化噪声（矢量图形除外），数据传输过程中导致的编码差错等等。它们在这里不是主要因素，故不作详细分析。

3.结论：

JPEG压缩算法在有损压缩的情况下能提供很高的压缩比同时还能保持不错的输出效果，但是这个压缩比也是有极限的，过多的丢失数据最明显的问题就是图像中出现方块。即便如此，它依旧在现在的计算机图像格式中占据举足轻重的作用，能够胜任大部分图像处理需求。