精确保留第一行第一列原始数据，将整幅图像一起编码

Image size:24x32

Method 7, compression ratio = 1.5680 恢复图像质量很好

Method 6, compression ratio = 1.5770

Method 1, compression ratio = 1.7864

Image size:50x80

Method 7, compression ratio = 1.5657

Method 6, compression ratio = 1.5651

Method 1, compression ratio = 1.8198

Image size:100x160

Method 7, compression ratio = 1.5475

Method 6, compression ratio = 1.6035

Method 1, compression ratio = 1.7592

Image size:240x320

Method 7, compression ratio = 2.1327 恢复图像质量不可接受

Method 1, compression ratio = 2.2022

--------------------------------------以上内容保存在backup\_Apr8th.rar中

原有程序按亮度分布范围划分为100份小区间进行量化，量化步长达到了1.6~2.0之间。现在改进了原有量化过程，量化步长改成标准的1，并进行了量化误差补偿。

Image size:240x320

Method 7, compression ratio = 1.7849 未补偿，恢复图像质量尚可，亮度稍暗

Image size:240x320

Method 7, compression ratio = 补偿后，恢复图像右下角亮度过高

Image size:240x320

Method 7, compression ratio = 1.7551 利用ceil函数进行本身就有问题，会带来系统性的量化误差。应该改成round函数，这样就消除了系统误差。恢复图像效果非常好

Image size:240x320

Method 7, compression ratio = 2.2201, PSNR =36.1792 round函数，量化步长2图像效果很好

Image size:240x320

Method 7, compression ratio = 2.6500, PSNR = 36.2541 round函数，量化步长3 图像有沿右下方向的微弱失真色彩带

Image size:240x320

Method 7, compression ratio = 3.0113 round函数，量化步长4 图像有沿右下方向的明显失真色彩带。但是当计算图片尺寸变小时，失真减少，PSNR明显提高

--------------------------------------以上内容保存在backup\_Apr12th.rar中

进行动态huffman编码的情况：

170.bmp, size:80\*80, step1: time=2.7344 compressionratio=1.6257

170.bmp, size:40\*40, step1: time=0.6719 compressionratio=1.6345

170.bmp, size:80\*80, step2: time=2.7031 compressionratio=2.03

170.bmp, size:40\*40, step2: time=0.6563 compressionratio=2.0428

170.bmp, size:80\*80, step4: time=2.5781 compressionratio=2.6774

170.bmp, size:40\*40, step4: time=0.6719 compressionratio=2.6896

采用经验码本的情况：

170.bmp, size:80\*80, step1: time=3.2031 compressionratio=1.5889

170.bmp, size:40\*40, step1: time=0.6406 compressionratio=1.6267

170.bmp, size:80\*80, step2: time=2.9375 compressionratio=1.9732

170.bmp, size:40\*40, step2: time=0.6406 compressionratio=2.0173

170.bmp, size:80\*80, step4: time=2.9688 compressionratio=2.5307

170.bmp, size:40\*40, step4: time=0.5469 compressionratio=2.5852

每次运行需要事先载入码本

虽然采用经验码本少了很多计算步骤，但是可能是由于我算法比较粗糙，没有优化，利用经验码本的情况下计算速度仍然没有很大起色，图像较大时甚至速度变慢。不过采用经验码本压缩率没有明显降低。

--------------------------------------以上内容保存在backup\_Apr15th.rar中

完成了分步读取压缩和分步解压图像的程序。对于240x320的图像，程序顺序读取40x40图像块。图像边缘部分的量化步长为4，较中间的部分量化步长为2， 中心部分量化步长为1。计算结束时给出整体图像压缩比，中心PSNR，整体PSNR以及原始图像和恢复图像的对比。

--------------------------------------以上内容保存在backup\_Apr20th.rar中

在独立程序中完成了从RGB图片格式到BAYER格式的转换，对BAYER格式图片各色彩分量的分离，低通滤波；逆滤波，图像重构，最后通过色彩插值恢复成RGB格式

--------------------------------------以上内容保存在backup\_Apr26th.rar中

完成了对编码解码程序的改造，使其处理的图像由原先的RGB格式变成了BAYER格式。具体增加步骤包括：RGB转BAYER格式，BAYER格式图像各色彩分量的分离，低通滤波，逆滤波，图像重构，色彩插值。

为了减少内存占用，将图像第一行第一列改为固定码长编码，节省一半的码本空间或者读取次数（不需要读取codebook\_A1了）

由于王sir说医生要求不要降低图像质量，所以图像所有区域都采用量化步长为1的近无损压缩，码本只需要唯一的一个codebook\_Errorquant\_step1

纠正了原先编码的小bug。如果图像中某像素值在码本中找不到，原先会出现图像整体无法编码的问题，现在采用与该像素最相近的编码来近似。

快速解码算法：码本改成二叉树结构，解码搜索采用二叉搜索树方式，可以大幅度提高解码速度。已完成，但是处理sample3.bmp时出错。

原先估计由于编码时采用的是线性存储的码本，解码时采用线性码本则正确，解码时采用二叉树形式存储的码本则错误，且两种解码算法经检验无误，则估计是两种形式的码本存在某种程度上的不一致。

--------------------------------------以上内容保存在backup\_May1st.rar中

完成了快速解码算法。上周出现的算法bug是因为线性排列的码本和树形结构码本不一致。原因查明了，是因为很多不同的像素在统计时出现次数一样，对其进行排序操作会有很多偶然性。而先前的程序中编写线性码本和编写树形码本的部分是分开进行的，所以导致了很多出现概率相同的像素编码不一样。

完成了对bayer格式图像预处理后各像素分布概率的统计，重新编排了码本。

--------------------------------------以上内容保存在backup\_May6th.rar中

将程序中某些固定参数改成手动，图像大小，图像分析单元大小均为手动可调。目前分析单元大小该为6x40

改正了PSNR的计算方法，计算得到的PSNR约为45.5dB（量化步长为1）或43.5dB（量化步长为2），比文献中给出的无损理论计算值46dB低约0.5dB。

--------------------------------------以上内容保存在backup\_May8th.rar中

统计数据分布编制码本时将不再将正负值统一编码，而是统计所有值的绝对值。这样改动后数据量减少约一半，码本所需存储空间减少一半，代价是编码和解码时，每个像素得多1bit用于标记正负。导致压缩率有少许降低（在压缩图片3，步长为2时，压缩率由1.8降低到1.7；压缩图片2，步长为2时，从2.24降低到2.04）。

--------------------------------------以上内容保存在backup\_May10th.rar中

将编码过程中使用的码本改成差值huffman形式，即每个huffman编码都是前一个编码与当前编码的有差别部分，大约能减少码本尺寸一半左右。

--------------------------------------以上内容保存在backup\_May11th.rar中

码本长度是33，改为32.

--------------------------------------以上内容保存在backup\_May18th.rar中

进行总体性能测试：

NEAR=2图像1：PSNR=43.9 CR=2.2 SCB: 576 bit

NEAR=2图像2：PSNR=43.5 CR=2.0 SCB: 576 bit

NEAR=2图像3：PSNR=43.4 CR=1.7 SCB: 576 bit

NEAR=3图像1：PSNR=42.3 CR=2.3 SCB: 308 bit

NEAR=3图像2：PSNR=41.7 CR=2.2 SCB: 308 bit

NEAR=3图像3：PSNR=41.5 CR=1.9 SCB: 308 bit

对于每个计算单位(6x40x1,共n像素)，计算复杂度为：O(n)

线性预测编码：3n

量化：n

调用coding: 最少n，最多32n，绝大多数情况下比n略大

加上符号标记：n