

CODING 平台使用说明

牛犇犇 2019.11

niu_benben@foxmail.com

CODING 平台为《编码引论》信源编码部分实验平台，主要实现图象压缩过程中的两部分功能：量化（Quantization）与变长编码（VLC）。下面对实验平台的基本操作进行介绍。

一、量化

- 1、OPEN FILE: 打开一张图片。图片格式要求为 **bmp** 格式的**灰度**图象。如平台中提供的 lena_128_bw.bmp。文件打开后，右侧左上角会显示当前图片，最下方会显示当前图片按照每像素 8bit 计算所需的比特数：input image bit。
- 2、BLOCK POLICY OPTION: 选取图象分块大小。其中 8x8 block 表示将图象按照 8*8 块大小进行分割，full picture 表示不进行分块。图象分块大小用作后续操作：
 - 1) 图象量化后的 PSNR 显示；2) 使用条带头的变长编码（Slice-VLC）。
- 3、QUANTIZATION OPTION: 选择不同的图象量化方式：
 - 1) uniform: 均匀量化。输入[1-255]之间的一个整数作为均匀量化步长。量化重建值为两个相邻边界值的中值。
 - 2) H.261: H.261 标准量化。输入[1-100]之间的一个整数作为量化参数 QP（Quantization Parameter）。QP 越大，量化图象质量越差，压缩率越高。量化操作是在 8x8DCT 变换之后。此处的 DCT 变换采用了整数运算，以加快运算速度。QP 较大时图象会出现明显的块效应。
 - 3) custom: 自定义量化。输入参数为量化数组，选取[1-255]之间的多个整数作为量化边界值，按从小到大的顺序输入，数字之间用空格分隔。量化重建值为量化区间内的均值。
自定义数组举例：
中部细化 5 35 65 88 104 120 136 152 168 185 215 245
8 值量化 40 76 108 134 158 189 224
- 4、start quantization: 点击按钮，开始量化操作，结束后会有量化成功的提示。量化后图象显示在右侧右上角。同时在左下角显示 processed image bit 和 PSNR 信息。量化前需要打开原始图片、选定块划分大小和选定量化参数，否则将进行相应错误提示。
- 5、processed image bit: 量化图象经过 Huffman 编码之后需要的比特数。程序内部集成了 Huffman 编码过程，可以计算出量化后图象经过编码之后的比特数。

可用作量化性能的比较。

- 6、PSNR: 量化图象与原始图象间的 PSNR。将鼠标移到量化后的图象上, 会按照 (2) 中所选的块划分方式, 计算出当前所指块的 PSNR, 显示在窗口左下方, 当块划分方式选择“full picture”时, 给出的是全图的 PSNR。
- 7、Zoom: 点击 start quantization 完成量化后, 可通过鼠标滚轮滑动对量化图象进行缩放。注意不要进行过度放大, 以免造成内存不足, 程序意外停止。
- 8、SAVE PROC FILE: 保存量化后图象。

二、变长编码 (VLC)

本部分完成对量化图象的变长编解码过程。用户需要自定义设计变长码表 table, 实现像素灰度的量化重建值对编码码字的一一映射。对于量化图象, 根据码表将每个像素灰度编码为二进制串, 最终得到量化图象的变长编码码流。同理, 对于得到的编码码流, 通过对变长码表的遍历查找, 解码出不同的码字对应的像素灰度值。最终完成解码过程, 得到重建图象。

下面对于平台的基本输入输出及 VLC 操作进行介绍:

1、Input: 输入码表

编解码码表由用户自定义设置。码表分为两种: 单符号 (one symbol) 码表和双符号 (two connected symbols) 码表, 分别对应两种符号映射方式。两种码表文件的示例如 table.txt 和 table2.txt。其文件格式有如下要求:

- 1) 码表文件为文本格式的 txt 文件。
- 2) 每行对应一种符号映射方式。前面写像素灰度值[1-255], 后面写对应的二进制编码码字, 若为双符号映射, 则有两个灰度值。各数值及码字之间以空格或者 Tab 分隔。
- 3) 文件的最后一行是一个单独的码字, 成为逃逸码。当像素的量化灰度值在上面的映射中无法找到时, 进行逃逸码编码。其编码格式为:

逃逸码 + 灰度值 8bit 表示 (+ 灰度值 8bit 表示, 对于双符号码表)

码表示例:

1) one symbol (table.txt)

40 1001

70 1100

01111

2) two connected symbols (table2.txt)

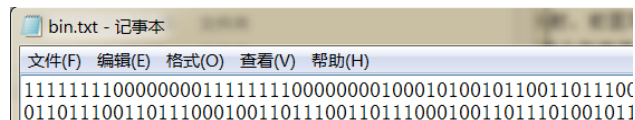
20 40 00001

60 100 01001

11110

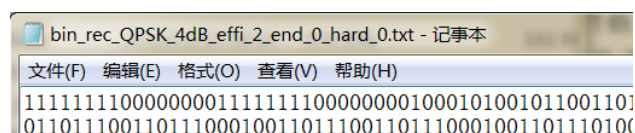
2、Output: 输出 VLC 编码码流

VLC 编码完成后得到的编码码流为 bin.txt，为文本格式的 txt 文件，在当前目录下输出。此码流将进行信道编解码过程。文件示例：



3、Input: 输入 VLC 编码码流

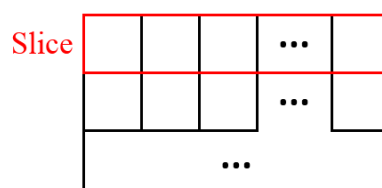
输入编码码流 bin.txt，进行信道编码，得到新的码流文件。此码流将作为 VLC 解码部分的输入，进行后续的变长解码过程。由于信道传输条件的限制，经过信道编码得到的码流很可能会出现误码，两个码流文件不再一致。文件示例：



4、条带

经过信道传输之后的编码码流可能会出现多个误码。常规的编解码过程中，编码内容仅为量化图象的各个像素灰度。一旦码流出现误码，解码过程中可能会出现连续的错误，最终的重建图象质量会大大降低。为了避免此种情况的出现，实验平台使用基于条带的 VLC 编解码过程。

依据（一）中量化部分的图象分块大小 BLOCK POLICY OPTION，条带可定义为一行相同块大小的分块。如下图所示。对应的条带 VLC 编解码过程如下：



1) 在 VLC 编码端，每一个条带起始位置，需要添加一个条带起始码 slice_start_code (24bit 长度)，以进行条带起始和条带位置的判断，后续分块进行正常的编码，直到下一个条带。

2) 在 VLC 解码端，需要对可能出现的条带起始码进行检测，如果检测到起始码，则进一步确定当前条带的位置，及该条带下的重建像素的位置。

条带 VLC 编解码可以在较少误码的情况下对重建图象进行修正，避免出现过度失真。但是，当码流出现较多误码时，此种编解码方式作用有限，解码过程会出现很多异常情况，重建图象质量很差。

5、基本操作

1) Slice Start Code

用户可自行设置条带起始码的前 24 位前缀，以便调整码流结构，避免重复条带头出现，一般使用默认设置即可。

2) input VLC table: 输入 VLC 码表

勾选单符号码表或者双符号码表，点击按钮，进行编码过程，若编码成功则提示成功信息，并输出码流大小，否则提示错误信息。编码前需要进行图象的量化，否则会进行相应错误提示。编码成功后会在当前目录下生成编码码流 bin.txt。

注意：为了考虑码表大小对于压缩性能的影响，将码表传输代价计入总的比特数中。采用简单的方式来衡量码表大小，即符号与码字的二进制表示位数总和。

3) NoiseLevel

用来模拟信道传输带来的误码效应。

Level = 0 时，无误码出现，解码端可完整解码。

Level = [1-5] 时，可用来模拟不同信噪比下信道质量，级别越高，误码越严重。此处的模拟仅仅是对码流进行若干次比特翻转得到，不能完全模拟实际信道的随机性，故仅在第一次作业中用来考察条带对于误码传播的作用。在完成信道编码部分的作业后，可进行实际的信源和信道编码联合调试，具体分析不同信道编码参数下产生的误码影响。

4) Iterations

用来模拟多次信道传输下的码流解码结果。

Iterations = 1 时，进行一次信道传输与码流解码，会提示相应的解码信息。

Iteration 大于 1 时，不再提示解码信息，多次解码得到的重建图象与原始图象的 PSNR 会输出到对应 txt 文件中。

5) input encoded stream: 输入 VLC 编码码流

点击按钮，进行信道编码及 VLC 解码过程。

若解码端得到的重建图象与量化图象完全相同，则提示“All correct!!!”，并输出码流大小，及重建图象与原始图象的 PSNR；

若解码端得到的重建图象与量化图象不完全相同，则提示“Reconstruction success but not match!”，并输出重建图象与原始图象的 PSNR。

否则，则认为解码失败，提示相应的异常信息。

解码图象显示在右侧左下角，量化图象与重建图象的差值显示在右侧右下角，用作差错识别。同时，量化与重建的差值图象将在当前目录下输出，

diff_proc_rec.bmp。

6) SAVE REC FILE: 保存重建图象。

6、异常信息

编码端:

1) Please change another image with even pixs width!

进行双符号编码时需要保证图象宽度为偶数。

2) Please use another VLC table!

编码码流中条带起始码的个数应该与设计的条带数目相同，但实际编码过程中各种码字的组合可能产生额外的条带起始码，这会导致解码端出现错误。这种情况下会提示错误信息，请调整条带起始码重新进行编码。

3) Wrong Slice Start Code!

条带起始码的前缀为定长 24 位 01 比特，请保证输入前缀的合法性。

4) Wrong NoiseLevel!

模拟的信道干扰程度为[0-5]，请选择合适的 Level。

5) Wrong Iterations!

循环次数为[1-100]，请选择合适的 Iterations。

解码端:

1) No enough bits !

解码未完成时已经遍历到码流末尾，已经解码出的象素个数少于量化图象，解码失败。

2) Too many bits !

解码未完成时已经得到了量化图象的所有象素数，仍有剩余码流未被处理，解码失败。

3) Reconstruction failed!

其他错误解码情况下的提示信息。