Codec平台使用说明

魏紫威 2021.10

E-mail: ziwei\_wei@foxmail.com

Codec平台用于《编码引论》课程中信源编码部分的实验教学工作。该平台使用MATLAB语言编程实现，主要包含图像压缩过程中量化（Quantization）操作以及变长编码（Variable Length Coding, VLC）方式编码。以下部分将分别介绍Codec平台的使用。

1. 量化

参数选择：

blockOption：取值范围为。每一个取值分别表示用于计算PSNR的子块尺寸和量化、变长编码中Slice的高度。取值0代表划分子块尺寸为整幅图，Slice高度为整幅图像的高；取值为1代表Slice高度为4；取值为2代表划Slice高度为8；取值为3代表划Slice高度为16；取值为4代表划分Slice高度为32；取值为5代表划分Slice高度为64。Slice高度在VLC中使用。

i\_quant：取值范围为。每一个取值分别表示平台所使用的量化方式。其中，取值为0表示均匀量化（uniform quantization）；取值为1表示使用H.261标准中的量化方式；取值为2表示自定义量化，通过自己及设定量化范围，对图像的每一个像素值进行量化。

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。具体实现方式详见程序。
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。变换以及反变换操作在平台中默认尺寸为8x8。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。值得注意的是，在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。对于输入的量化数组[A,B]，首先在首尾分别加上0和255数值，构成最后的量化数组[0,A,B,255]。对于其中每一个区间A~B，采用左闭右开[A,B)的方式进行区间内像素统计。计算均值时采用向下取整(floor)的方式。自定义数组如下：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. Huffman编码

量化得到的图象经过Huffman编码之后得到所需要的比特数。程序内部已集成Huffman编码函数，（注：但不可见，要求自己实现相关程序。）可以计算量化后图象经过Huffman编码后的比特数，可用作量化性能的比较。

1. 变长编码（VLC）

本实验平台集成对量化图象的变长编解码过程。用户需自定义设计变长码表table，实现象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，根据码表将每个象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象素灰度值。最终完成解码过程，得到重建图象。

参数选择：

vlcRadio：取值为0或者1，分别表示使用单符号变长编码以及双符号变长编码。

onesymbol\_codebook\_file：单符号变长编码的码表路径及文件名。默认为工作目录下的table.txt文件。

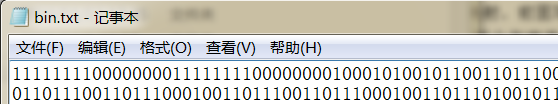
twosymbol\_codebook\_file：双符号变长编码的码表路径及文件名。默认为工作目录下的table2.txt文件。

码表文件的每行对应一种符号映射方式。前面写象素灰度值[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。两种码表文件的最后一行是一个单独的码字，称为逃逸码。当象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：

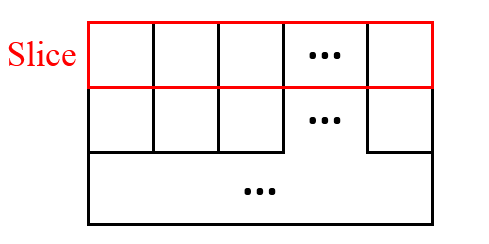
逃逸码 + 灰度值8bit表示 （+ 灰度值8bit表示，对于双符号码表）

slice\_start\_code：每一个slice头的编码，表示一个slice的开始。默认为一个24bit的码流。

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：



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象素灰度。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blockOption参数，确定Slice的高度。Slice的宽度为图象的宽度。



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）及条带位置索引(8bit长度)，以进行条带起始和条带位置的判断，后续进行正常的编码，直到下一个条带。

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，及该条带下的重建象素的位置。

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。至此，变长编码中的编码为三种：1）slice\_start\_code 2）码表中的码字 3）逃逸码 + 灰度值8bit。

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，即符号值的二进制表示的位数与码字的二进制长度之和。计算传输的码流总比特数时，使用bin.txt中的比特数加上单/双符号码表比特数的和作为总比特数。

1. 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。

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。