

数字媒体处理技术实验报告——实验（二）

21R0361班 2021113117-王宇轩

目录

1.实验环境

2.文件列表

3.实验内容

- 3.1 对音频文件进行mp3编码和解码
- 3.2 对BMP图像进行JPEG压缩
- 3.3 对BMP图像进行JBIG压缩
- 3.4 对视频进行H.265(HEVC)压缩
- 3.5 使用FFMPEG平台对视频进行编码、转码操作

4.实验总结

1. 实验环境

- 操作系统：Windows
- 编程语言：Python、C、C++
- 所用软件：PyCharm、VS Code、cmd、FFmpeg、JBIG-Kit、HM（HEVC参考软件）
- 主要库调用：wave、PIL

2. 文件列表

```
| 实验报告.pdf
|
|—ffmpeg
|   input.mp4
|   验收视频.mp4
|
|—hevc
|   BasketballDrill.cfg
|   BasketballDrill_832x480_50_qp37.bin
|   Decode.cmd
|   Encode.cmd
|   encoder_intra_main.cfg
|   TAppDecoder.exe
|   TAppEncoder.exe
```

```
|      验收视频.mp4
|
|---jbig
|      image.bmp
|      jbig.cmd
|      pbmtojbg.exe
|      验收视频.mp4
|
|---jpeg
|      image.bmp
|      jpeg.py
|      验收视频.mp4
|
|---mp3
|      audio.wav
|      mp3.py
|      验收视频.mp4
```

3. 实验内容

3.1 对音频文件进行mp3编码和解码

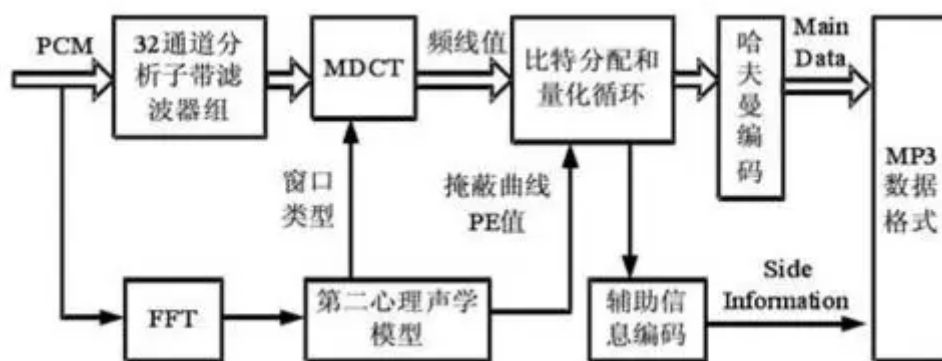
相关文件见mp3目录，如下表。

文件名	内容
audio.wav	输入音频文件
mp3.py	mp3编解码源程序
验收视频.mp4	程序运行视频，用于验收

执行mp3.py（需要环境变量调用FFmpeg），先从audio.wav获取输入音频PCM数据，写入文件**input.pcm**，随后将其编码为**output.mp3**，再次解码为**decoded.pcm**，执行过程已记录于验收视频。程序命令行输出相关数据如下：

```
压缩时间： 0.2064356803894043 秒
压缩前文件大小： 2570068 字节
压缩后文件大小： 233891 字节
压缩倍数： 9.10058%
解压缩时间： 0.06096506118774414 秒
```

Mp3处理流程图如下



基本流程如下：

1. **滤波器段** 数字音频信号以脉冲的形式送进编码器，首先会经过一滤波器段(filterbank)，由两个串联的滤波器段组成：一个多相分析滤波器段和一个MDCT(Modified Discrete Cosine Transform，经改良的离散余弦转换)，输入的音频信号在20Hz-20k Hz的频率范围内，通过第一个滤波器段后，把信号按频率分成32个小频带，称为子带(Subband)。人耳对不同频段的灵敏度是不同的，编码器可对不同的子带进行不同的量化分层。
2. **人体听觉心理学模型** 两个重要原理：最小听觉门槛判定和遮蔽效应。编码器通过分析滤波器段分析音频信号的频谱组成并应用听觉心理学模型来估计仅仅可以注意得到的噪音电平。在后面的量化和编码阶段，编码器会提高强信号附近的噪音电平，也就是减少量化位数，这样就能达到节省流量，压缩数据的目的。如果所有量化噪音都能控制在屏蔽临界曲线以下，那么处理后人耳就区别不出哪个才是原音。
3. **量化和编码** 当PCM讯号被分成好几个频段并经过一系列的处理后，最后经过MDCT，将波型转换为一连串的系数。这些系数就由Huffman编码器会选择最合适的Huffman表来做最后的压缩。

3.2 对BMP图像进行JPEG压缩

相关文件见jpeg目录，如下表。

文件名	内容
image.bmp	输入BMP图像文件
jpeg.py	JPEG压缩源程序
验收视频.mp4	程序运行视频，用于验收

运行jpeg.py文件，将image.bmp压缩生成**compressed.jpeg**，并再次解压缩生成**decompressed.bmp**。

JPEG压缩的基本流程如下：

1. **颜色空间转换** 首先将原始图像从RGB颜色空间转换为YCbCr颜色空间。Y表示亮度（灰度）分量，而Cb和Cr表示蓝色和红色的色度分量。这种转换利用了人眼对亮度更敏感、对色度变化不太敏感的特点。

- 2. **图像分块** 将图像分成8x8像素的小块。每个小块独立处理。
- 3. **离散余弦变换 (DCT)** 对每个8x8像素块进行离散余弦变换，将空域中的像素数据转换为频域中的系数。
- 4. **量化** 对DCT系数进行量化。将较高频率的系数舍入为较低的值，减少了高频细节的表示精度实现压缩。
- 5. **压缩** 对量化后的DCT系数进行熵编码压缩。
- 6. **重建** 逆量化、逆DCT变换和色度空间转换，将压缩后的数据恢复为原始的RGB图像。

3.3 对BMP图像进行JBIG压缩

相关文件见jpig目录，如下表。

文件名	内容
image.bmp	输入BMP图像文件
jbig.cmd	批处理文件，调用相关程序进行JBIG压缩
pbmtojbg.exe	用JBIGKIT编译的JBIG编码例程
验收视频.mp4	程序运行视频，用于验收

执行**jbig.cmd**，首先调用ffmpeg将输入bmp图像先转换为**image.ppm**文件，进而转换为**image.pbm**，最后调用JBIGKIT编译的JBIG编码程序**pbmtojbg.exe**，将pbm文件编码生成**image.jbg**。jbig.cmd内容如下：

```
ffmpeg -i image.bmp image.ppm
ffmpeg -i image.ppm -pix_fmt monob image.pbm
pbmtojbg image.pbm image.jbg
```

3.4 对视频进行H.265(HEVC)压缩

相关文件见hevc目录，如下表。

文件名	内容
BasketballDrill.cfg	编码配置文件
BasketballDrill_832x480_50_qp37.bin	测试序列
Decode.cmd	解码指令
Encode.cmd	编码指令
encoder_intra_main.cfg	编码配置文件
TAppDecoder.exe	解码程序

文件名	内容
TAppEncoder.exe	编码程序
验收视频.mp4	程序运行视频，用于验收

编译HEVC提供的参考软件HM，生成编解码程序，用测试序列BasketballDrill_832x480_50_qp37.bin进行编解码实验。

解码指令**Decode.cmd**如下：

```
TAppDecoder.exe -b BasketballDrill_832x480_50_qp37.bin -o BasketballDrill_832x480_50.yuv
```

解码程序**TAppDecoder.exe**输入二进制序列文件**BasketballDrill_832x480_50_qp37.bin**其大小为4.02 MB (4,224,962 字节)，解码生成**BasketballDrill_832x480_50.yuv**，大小为73.6 MB (77,275,136 字节)。

编码指令**Encode.cmd**如下：

```
TAppEncoder.exe -c encoder_intra_main.cfg -c BasketballDrill.cfg -o BasketballDrill_832x480_50.bin
```

编码程序**TAppEncoder.exe**输入视频文件**BasketballDrill_832x480_50.yuv**并将其编码为二进制序列文件**BasketballDrill_832x480_50.bin**，其中encoder_intra_main.cfg和BasketballDrill.cfg是官方提供的编码配置文件。编码后，二进制序列文件的大小为73.1 MB (76,677,120 字节)，比原视频有所降低。

3.5 使用FFMPEG平台对视频进行编码、转码操作

相关文件见ffmpeg目录，如下表。

文件名	内容
input.mp4	输入视频文件
验收视频.mp4	程序运行视频，用于验收

利用FFmpeg平台提供的命令行界面。即可快速对视频进行编码、转码操作。如下命令行指令：

```
ffmpeg -i input.mp4 -c:v libx264 -c:a aac output.mp4
```

可以将输入文件"input.mp4"转码为H.264视频编码和AAC音频编码的输出文件"output.mp4"。

4. 实验总结

熟悉了媒体的编解码流程和编码与解码的实现。