

哈爾濱工業大學

数字媒体技术  
实验报告

题	目	编解码平台及应用
学	院	计算机科学与技术
专	业	大数据
学	号	L170300901
学	生	卢兑琬
任	课 教 师	

哈尔滨工业大学计算机科学与技术学院

2021.3

## 实验三：编解码平台及应用

### 一、实验内容或者文献情况介绍

跨媒体的情感分类

- 根据图像、视频、音频，分别提取其特征，根据特征对其表达的情感进行分类
- 是否可以在各种多媒体的特征表示之间建立转换关系，从而可以用图像特征来检索具有同样情感属性的音频和视频，或者用任意一种表达其属性。
- 情感属性可以换为：正能量性，积极性等，并进一步给出定量分数

要求

- 最后提交实验报告（包括原理介绍、论文阅读、测试数据集和测试结果等）和源代码以及可执行程序
- 具体参见模板。

- 对真彩色 BMP 图像进行颜色量化，将颜色数目减少为 256 种颜色
  - GIF 图像都是通过颜色量化生成,然后比较 256 种颜色图像与原始图像之间的 PSNR 值
  - 对二值图像进行 JBIG 压缩(专门针对二值图像)
  - 尝试运行 JBIG 代码进行编解码:  
<https://www.cl.cam.ac.uk/~mgk25/jbigkit/>
  - 将可执行程序的操作说明写清楚!
  - 使用 FFMPEG 平台对视频进行编码、转码操作
  - 访问 ffmpeg 官网，学会使用 ffmpeg 进行视频操作

对真彩色 BMP 图像进行颜色量化，将颜色数目减少为 256 种颜色

BMP (\*BMP, \*.RLE):

是指在 DOS 和视窗兼容电脑上使用的“标准视窗 Bitmap”图像。该 BMP 文件格式用于视窗的基础画面,支持 RGB 模式和 Indexed 模式。初期 BMP 文件没有压缩,但自 Windows 3.0 之后,可以使用 RLE 压缩方式压缩文件。BMP 文件有黑白和 4, 8, 24 比特格式,分别表现 2 色, 16 色, 256 色和特鲁色。乐视和阿尔法频道不支持。

Compuserve GIF:

它是 GraphicsInterchange Forbat 的缩写,其设计是为了尽量减少文件传输时间。

以 Betwap 方式的压缩形式,是专门支援 8Bet256 颜色的 Indexed 模式的文件格式,在网页制作中经常使用。

GIF 文件有两种类型, 根据编码方法分为 87A 和 89A。 89A 是最新版本, 将多张图片合二为一, 可以产生像动一样的效果, 多用于在网络及多媒体环境下实现简单的视频。(potshop 支持 GIF 格式, 但不支持视频 GIF 文件。)

GIF 根据显示图像的方式分为 Non-Interlaced 和 Interlaced 方式, 共有 4 种方式可以储存。

## Graphics Interchange Format

GIF 也是为加快传输速度而压缩存储的方式之一, 它的特点在于在習上多用, 可以在图像文件内将多张图像放在一个文件中。

与 JPG 文件相比, 压缩率稍差, 而传输速度快, 图像损伤小。遗憾的是 GIF 最多只能表现 256 种颜色。



### <根据压缩率 GIF 文件的画质变化>

GIF 文件容量控制色数和 dither 值。

dither 价格是柔和表现颜色和颜色界线的的数据。

随着颜色数量的减少, dither 价格下降, 画质会变粗糙, 但容量会减少。

保存 GIF 时, 选项上可选择 256 / 128 / 64 / 32 / 16 / 8 / 4 / 2

像单纯的插图一样,如果规定颜色数量,根据颜色数量指定的话,容量就会大幅减少。



很好奇一下子把照片图片缩小到极端颜色的数量会怎么样,所以试了一次。dither 价格保持不变,只减少了颜色数,结果就出来了。

PSNR

PSNR 计算如下:

$$\begin{aligned}
 PSNR &= 10 \log_{10} \left( \frac{MAX_I^2}{MSE} \right) \\
 &= 20 \log_{10} \left( \frac{MAX_I}{\sqrt{MSE}} \right) \\
 &= 20 \log_{10}(MAX_I) - 10 \log_{10}(MSE)
 \end{aligned}$$

这里 MAXI 太阳的后裔是该影像的最低价,8 位数的灰色影像是 255。另外,MSE 的平均误差可以计算如下。

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

在这里, I 是 m x n 尺寸的灰石形象, K 是指 I 中包含杂音的形象,即歪曲的形象。由于官方 1 的分母中有 MSE,所以 MSE(平均平方差)越小,就拥有越大的

PSNR。因此,品质好的形象会拥有较大的 PSNR,品质不好的形象会拥有较小的 PSNR。

PSNR 与图像质量之间的关系

让我们来寻找一下在同一原图中被不同程度 JPEG 压缩损失的图像的 PSNR。



从 JPEG 压缩损失图片的 PSNR 可以看出,压缩率越大,PSNR 也越小。

34.8227dB → 30.9394dB → 25.8699dB

我们可以亲眼看到,越是往右,歪曲现象就越严重。

34.8227dB (质量不错) → 30.9394dB (质量差一点) → 25.8699dB (质量差一点)

## 二、 算法简介及其实现细节

对二值图像进行 JBIG 压缩

JBIG 是为了压缩李真的形象,特别是 Facsimily 的形象而开发的。但是它也用于其他用途。比组合 4(packs)标准提高了 20%甚至 50%。在特殊情况下,压缩率提升了 30 倍。1993 年规格化。

JBIG 采用 IBM 专利的一种算术编码方式。这一方式被命名为“Q-coder”。这一方式就是以前面的 bit 或画的前排 bit 计算概率。根据扫描顺序,需要发生压缩或压缩动作,因此后面的螺栓不能参照压缩(reference)。增加若干超大头(5%),可支持有序传输功能。

JBIG 采用的好处如下:

对 Bitonal 及 Grayscale 形象卓越的无损失压缩功能

越小的文件越快速的亿岁及 Retrieve 功能

对格雷斯凯尔形象的更有效的压缩功能

具有在更高的分辨率上存储压缩图像的能力,故 OCR 认识具有更好的准确度

Progressive Capability -

逐渐将代码图像进行装饰时,原图像的阻隔上图版本将首先打开,然后加倍分辨率进行装饰。

Progressive Encoding -

在拥有一个数据库的应用程序中,可有效构建具有不同分辨率的 Serve 输出功能。另外,还可以开发低速和高速通讯链接的图像浏览器应用程序。即,低分辨率的局域网可以高速传输或显示,相对高分辨率的局域网可以低速链接。

JBIG 的主要用途是替代 CCITT Group 3/4 中使用的低效压缩算法。对所打印文本或扫描 Line Art 图片,JBIG 的压缩率比 Group4 的压缩率高出 10 至数十个百分点。

电脑生成的图像打印文本比 Group4 更出色 500%。

使用 Grayscale 对图片进行明暗处理或去污处理,图像质量提高 2~3 倍。

JBIG 对 NOIZ 非常敏感,在阁下的形象内 NOIZ 增加时可减少压缩率。

JBIG 可代替无损失 JPEG 使用。JBIG 在每像素 2~5Bit 人的形象中,创造出比无损失 JPEG 更高的品质,对拥有 6~8 像素数据的形象产生相同结果。

更小 20~60%的文件可以节省费用及快速传送文件。

JBIG 的使用领域:

Document Imaging:基于纸张的电子扫描,存储和管理  
存取账户

网络应用或所有网络应用

A0 大小的工学纪录片储存。由于可使用多重激光,使用者可提前查看障碍商图,在浏览器上显示自己想要的分辨率。

JPEG 压缩的基本原理

- 将视觉上重要的部分和不太重要的部分分离(DCT 变换)
- 把重要部分活下来,把不重要的部分损失掉

减少数据量(量化)

- 通过颜色型号的变换,分为重要部分和不重要部分

李(RGB->YIQ)





327KB (335,684byte)



44.9KB (46,024byte)



视频放大缩小字体功能



放大缩小字体功能

导出函数

DLL 库导出的函数有三个（除下列方法之外，还有原代码提供的 jbig2\_encode\_generic 方法：

```
// -----
uint8_t *
jbig2_encode_generic(struct Pix *const bw, const bool full_headers,
                    const int xres, const int yres,
                    const bool duplicate_line_removal,
                    int *const length);

uint8_t *
jbig2_encode (int width, int height, int stride, bool zeroIsWhite, uint8_t * const bw, int *const length);
void release (uint8_t * const memblock);
#endif // JBIG2ENC_JBIG2_H__
```

jbig2\_encode 方法用于编码传入的字节数组。其参数有六个：

1. Width: 图像宽度
2. Height: 图像高度
3. Stride: 一行像素所占的双字(DWORD)数
4. ZeroIsWhite: 将 BW 的 0 视为白色
5. BW: 图像的字节数组 (Byte[]), 此字节数组是原始黑白图像的二进制数据, 默认情况下, 二进制位 1 表示白色, 0 表示黑色
6. Length: 字节数组长度
7. 传出参数: 编码后的 JBIG2 字节数组

### 三、 实验设置及结果分析（包括实验数据集）

使用 FFMPEG 平台对视频进行编码、转码操作

1. Demuxing  
:从文件(DCF)中提取录像, 音频数据流
2. decoding  
:使用特定视频光盘获取原始视频数据
3. Encoding  
:从原视频数据转换成特定视频光盘
2. 墨星(Muxing)  
:将转换的数据(录像)装入文件(DCF)

基本 FFmpeg 使用方法(命令语)

ffmpeg 《输入选项》, 《输入文件》, 《打印选项》, 《打印文件》

输入选项

- 指定输入文件:-i input.ts
- 强制指定形式:-f mkv

ex) ffmpeg -i input.ts -c:v libx264 output.mkv



软件编码

`-c:v libx264 (H.264)`

`-c:v libx265 (H.265, HEVC)`

参考。利用 FFmpeg 剪视频

ex) `ffmpeg -i input.avi -ss 600 -t 120 -vcodec copy -acodec copy clip.avi`

`-ss 10 分钟(600 秒)`开始部分文件

`-t 2 分钟(120 秒)`左右选择部分

将 10 分钟的 avi 文件分成一半的时候

`ffmpeg -i input.avi -vcodec copy -acodec copy -t 300 before.avi`

`-t 300(秒)`指 5 分钟之前没有指定起点, 所以从开始到 5 分钟为止。

`ffmpeg -i input.avi -vcodec copy -acodec copy -ss 300 after.avi`

`-ss 300(秒)`5 分钟开始的意思, 因为结尾部分没有指定, 所以从 5 分钟开始一直到最后。

参考。利用 FFmpeg 的视频合成

把合页视频的名单制作成文本文件。

```
$ cat > list.txt
```

```
file a.mkv
```

```
file b.mkv
```

```
$ ffmpeg -f concat -i list.txt -c copy output.mkv
```

```
ffmpeg -i <source>.mp4 -c:a copy -c:v copy <target>.mkv
```

`-i` 是为指定输入文件而设定的, `-c` 是设定 codeo, 但设定 video 时是 v, 设定 audio 时是 a。一般会指定 codex 的名字, 会使用原样的 codex, 所以交给 copy 就可以了。如果在目的文件名称上使用 mkv, 可以自动转换成 mkv 形式。