天津理工大学

计算机科学与工程学院

实验报告

2017 至 2018 学年 第 二 学期

实验一 图像文件分析

课程名称	数字图像处理				
学号	**20152180**	学生姓名	王帆	年级	**2015**
专业	计算机 科学与技术	教学班号	**2**	实验地点	主 **7-212**
实验时间	**2018** 年 **3** 月 **26** 日 第 **7** 节 至 第 **8** 节				
主讲教师	杨淑莹				

实验成绩

软件运行	效果	算法分析	流程设计	报告成绩	总成绩

实验（ 一 ）	实验名称	图像文件分析
软件环境	Windows Visual Studio 2017	
硬件环境	PC	

实验目的

1. 分析几种常用的图像文件格式。
2. 打开 BMP、JPEG 图像文件，并显示。

实验内容（应包括实验题目、实验要求、实验任务等）

一、分析几种常用的图像文件格式

1. 分析 BMP 文件格式。

要求：分析 BMP 文件。

说明：

1. BMP 格式简介
2. BMP 文件结构
3. BMP 文件块的结构

2. 分析 JPEG 文件格式。

要求：分析 JPEG 文件。

说明：

① JPEG 格式简介
② JPEG 文件结构
③ JPEG 中的关键数据块

针对某一个你感兴趣的图像处理项目，如人脸识别、身份证号码识别、汽车牌照识别等，实现以下功能.

二、打开 BMP 文件，并显示

任务：

（1）在视图中制作一个【打开位图】菜单，打开一个 BMP 位图。

（2）在视图中制作一个【显示 BMP 位图】菜单，显示一个 JPEG 位图。

三、分析 JPEG 文件，打开 JPEG 文件，并显示

任务：

（1）在视图中制作一个【打开 JPEG 位图】菜单，打开一个 JPEG 位图。

（2）在视图中制作一个【显示 JPEG 位图】菜单，显示一个 JPEG 位图。

一、分析几种常用的图像文件格式

1. 分析 BMP 文件格式

1) BMP 格式简介

BMP（全称 Bitmap）是 Windows 操作系统中的标准图像文件格式，可以分成两类：设备相关位图（DDB）和设备无关位图（DIB），使用非常广。它采用位映射存储格式，除了图像深度可选以外，不采用其他任何压缩，因此，BMP 文件所占用的空间很大。BMP 文件的图像深度可选 lbit、4bit、8bit 及 24bit。BMP 文件存储数据时，图像的扫描方式是按从左到右、从下到上的顺序。由于 BMP 文件格式是 Windows 环境中交换与图有关的数据的一种标准，因此在 Windows 环境中运行的图形图像软件都支持 BMP 图像格式。

2) BMP 文件结构

BMP 文件由文件头、位图信息头、颜色信息和图形数据四部分组成。

3) BMP 文件块的结构

BMP 文件头（14 字节）

BMP 文件头数据结构含有 BMP 文件的类型、文件大小和位图起始位置等信息。

```
typedef struct tagBITMAPFILEHEADER
{
    WORD bfType;//位图文件的类型，必须为 BM(1-2 字节）
    DWORD bfSize;//位图文件的大小，以字节为单位（3-6 字节，低位在前）
    WORD bfReserved1;//位图文件保留字，必须为 0(7-8 字节）
    WORD bfReserved2;//位图文件保留字，必须为 0(9-10 字节）
    DWORD bfOffBits;//位图数据的起始位置，以相对于位图（11-14 字节，
    //文件头的偏移量表示，以字节为单位
}__attribute__((packed)) BITMAPFILEHEADER;
```

位图信息头（40 字节）

BMP 位图信息头数据用于说明位图的尺寸等信息。

```
typedef struct tagBITMAPINFOHEADER{
DWORD biSize;//本结构所占用字节数（15-18 字节）
LONG biWidth;//位图的宽度，以像素为单位（19-22 字节）
LONG biHeight;//位图的高度，以像素为单位（23-26 字节）
WORD biPlanes;//目标设备的级别，必须为 1(27-28 字节）
WORD biBitCount;//每个像素所需的位数，必须是 1（双色），（29-30 字节
//4(16 色），8(256 色）16(高彩色)或 24（真彩色）之一
DWORD biCompression;//位图压缩类型，必须是 0（不压缩），（31-34 字节
//1(BI_RLE8 压缩类型）或 2(BI_RLE4 压缩类型）之一
DWORD biSizeImage;//位图的大小(其中包含了为了补齐行数是 4 的倍数而添
节）
LONG biXPelsPerMeter;//位图水平分辨率，每米像素数（39-42 字节）
```

```
13  LONG biYPelsPerMeter;//位图垂直分辨率，每米像素数（43-46 字节)
14  DWORD biClrUsed;//位图实际使用的颜色表中的颜色数（47-50 字节）
15  DWORD biClrImportant;//位图显示过程中重要的颜色数（51-54 字节）
    }__attribute__((packed)) BITMAPINFOHEADER;
```

颜色表

颜色表用于说明位图中的颜色，它有若干个表项，每一个表项是一个 RGBQUAD 类型的结构，定义一种颜色。RGBQUAD 结构的定义如下：

```
1  typedef struct tagRGBQUAD{
2  BYTE rgbBlue;//蓝色的亮度（值范围为 0-255)
3  BYTE rgbGreen;//绿色的亮度（值范围为 0-255)
4  BYTE rgbRed;//红色的亮度（值范围为 0-255)
5  BYTE rgbReserved;//保留，必须为 0
6  }__attribute__((packed)) RGBQUAD;
```

颜色表中 RGBQUAD 结构数据的个数有 biBitCount 来确定：

当 biBitCount=1,4,8 时，分别有 2,16,256 个表项；

当 biBitCount=24 时，没有颜色表项。

位图信息头和颜色表组成位图信息，BITMAPINFO 结构定义如下：

```
1  typedef struct tagBITMAPINFO{
2  BITMAPINFOHEADER bmiHeader;//位图信息头
3  RGBQUAD bmiColors[1];//颜色表
4  }__attribute__((packed)) BITMAPINFO;
```

位图数据

位图数据记录了位图的每一个像素值，记录顺序是在扫描行内是从左到右，扫描行之间是从下到上。位图的一个像素值所占的字节数：

当 biBitCount=1 时，8 个像素占 1 个字节；

当 biBitCount=4 时，2 个像素占 1 个字节；

当 biBitCount=8 时，1 个像素占 1 个字节；

当 biBitCount=24 时，1 个像素占 3 个字节,按顺序分别为 B,G,R；

Windows 规定一个扫描行所占的字节数必须是

4 的倍数（即以 long 为单位），不足的以 0 填充，

biSizeImage = ((((bi.biWidth * bi.biBitCount) + 31) & ~31) / 8) * bi.biHeight;

具体数据举例：

如某 BMP 文件开头：

4D42 46900000 0000 0000 4600 0000 2800 0000 8000 0000 9000 0000 0100*1000 0300 0000 0090 0000 A00F 0000 A00F0000 0000 00000000 0000*00F8 E007 1F00 0000*02F1 84F1 04F1 84F1 84F1 06F2 84F1 06F2 04F2 86F2 06F2 86F2 86F2 ....

2. 分析 JPEG 文件格式。

1) JPEG 格式简介

JPEG（Joint Photographic Experts Group）是在国际标准化组织(ISO)领导之下制定静态图像压缩标准的委员会，第一套国际静态图像压缩标准 ISO 10918-1(JPEG)就是该委员会制定的。由于 JPEG 优良的品质，使他在短短几年内获得了成功，被广泛应用于互联网和数码相机领域，网站上 80%的图像都采用了 JPEG 压缩标准。

JPEG 本身只有描述如何将一个影像转换为字节的数据串流（streaming），但并没有说明这些字节如何在任何特定的储存媒体上被封存起来。.jpeg/.jpg 是最常用的图像文件格式，由一个软件开发联合会组织制定，是一种有损压缩格式，能够将图像压缩在很小的储存空间，图像中重复或不重要的资料会被丢失，因此容易造成图像数据的损伤。尤其是使用过高的压缩比例，将使最终解压缩后恢复的图像质量明显降低，如果追求高品质图像，不宜采用过高压缩比例。但是 JPEG 压缩技术十分先进，它用有损压缩方式去除冗余的图像数据，在获得极高的压缩率的同时能展现十分丰富生动的图像，换句话说，就是可以用最少的磁盘空间得到较好的图像品质。而且 JPEG 是一种很灵活的格式，具有调节图像质量的功能，允许用不同的压缩比例对文件进行压缩，支持多种压缩级别，压缩比率通常在 10：1 到 40：1 之间，压缩比越大，品质就越低；相反地，品质就越高。比如可以把 1．37Mb 的 BMP 位图文件压缩至 20．3KB。当然也可以在图像质量和文件尺寸之间找到平衡点。JPEG 格式压缩的主要是高频信息，对色彩的信息保留较好，适合应用于互联网，可减少图像的传输时间，可以支持 24bit 真彩色，也普遍应用于需要连续色调的图像。

2) JPEG 文件结构

JPEG 文件使用的数据存储方式有多种。最常用的格式称为 JPEG 文件交换格式（JPEG File Interchange Format，JFIF）。而 JPEG 文件大体上可以分成两个部分：标记码(Tag)和压缩数据。

标记码（Tag） 2 Bytes
数据长度（大端序） 2Bytes
数据 n-2 Bytes
……
下一个数据段

标记码由两个字节构成，其前一个字节是固定值 0xFF，后一个字节则根据不同意义有不同数值。在每个标记码之前还可以添加数目不限的无意义的 0xFF 填充，也就说连续的多个 0xFF 可以被理解为一个 0xFF，并表示一个标记码的开始。而在一个完整的两字节的标记码后，就是该标记码对应的压缩数据流，记录了关于文件的诸种信息。

常用的标记有 SOI、APP0、DQT、SOF0、DHT、DRI、SOS、EOI。

注意，SOI 等都是标记的名称。在文件中，标记码是以标记代码形式出现。例如 SOI 的标记代码为 0xFFD8，即在 JPEG 文件中的如果出现数据 0xFFD8，则表示此处为一个 SOI 标记。

JPG（JFIF）一般结构
SOI （0xFFD8）
APP0（0xFFE0）
APPn（0xFFEn）
DQT（0xFFDB）
SOFx（0xFFCx）
DHT（0xFFC4）
SOS（0xFFDA）
scanData
EOI（0xFFD9）

3) JPEG 中的关键数据块

1. SOI

代表 JFIF 图像数据的开始

2 Bytes	标记码 0xFFD8

2. APP0

应用程序标记 0

2 Bytes	标记码 0xFFE0
2 Bytes	数据段长度，包含本字段，但不包括标记码
5 Bytes	固定值 0x4A46494600，字符串“JIF0”
1 Bytes	主版本号
1 Bytes	副版本号
1 Bytes	图像密度单位（0：无单位 1：点数/英寸 2：点数/厘米）
2 Bytes	X 方向像素密度
2 Bytes	Y 方向像素密度
1 Bytes	缩略图水平像素数目
1 Bytes	缩略图垂直像素数目
n Bytes	缩略图，RGB 位图数据

3. APP1

应用程序标记 1，TIFF 数据

2 Bytes	标记码 0xFFE1
2 Bytes	数据段长度，包含本字段，但不包括标记码
6 Bytes	固定值 0x457869660000，字符串“Exif”
n Bytes	标签图像文件格式数据（TIFF）

4.APPn
拓展应用程序标记 2~15， 为其他应用程序保留

2 Bytes	标记码 0xFFE1~0xFFFF
2 Bytes	数据段长度，包含本字段，但不包括标记码
n Bytes	数据内容

5.DQT
量化表，存储了对扫描数据进行量化的 8*8 矩阵。

2 Bytes	标记码 0xFFDB
2 Bytes	数据段长度，包含本字段，但不包括标记码
4 bits	精度（0：8 位 1：16 位）
4 bits	量化表 ID，一般有 2 张，最多 4 张，取值 0~3
64 Bytes	表项（当精度为 16 位时，此字段有 128Bytes）

双线内部分可以重复出现，根据量化表 ID，存储多张量化表

6.SOF0
图像帧开始

2 Bytes	标记码 0xFFC0
2 Bytes	数据段长度，包含本字段，但不包括标记码
1 Bytes	每个数据样本位数，固定值 8
2 Bytes	图像高度（像素）
2 Bytes	图像宽度（像素）
1 Bytes	颜色分量数，JFIF 中使用 YCbCr 所以为固定值 3 （1：灰度图 3：YCbCr 4：CMYK）
1 Bytes	颜色分量 ID
4 bits	颜色分量水平采样因子
4 bits	颜色分量垂直采样因子
1 Bytes	使用的量化表 ID

双线内部分将重复出现，依 ID 对颜色分量中的颜色进行描述。

7.DHT
Huffman 表，存储了对扫描数据进行压缩的 Huffman 表，共 4 张。
DC 直流 2 张，AC 交流 2 张。

2 Bytes	标记码 0xFFDB
2 Bytes	数据段长度，包含本字段，但不包括标记码

4bits	Huffman 表类型（0：DC 直流 1：AC 交流）
4bits	Huffman 表 ID，DC/AC 表分开编码
16Bytes	不同位数的码字数量
nBytes	编码内容含义

双线内部分可以重复出现，根据表 ID 及 DC/AC，存储多张 Huffman 表。

7.SOS

扫描数据开始

2Bytes	标记码 0xFFDA
2Bytes	数据段长度，包含本字段，但不包括标记码
1Bytes	颜色分量数，JFIF 中使用 YCbCr 所以为固定值 3 （1：灰度图 3：YCbCr4：CMYK）
1Bytes	颜色分量 ID
4bits	DC 直流分量使用的 Huffman 表 ID
4bits	AC 交流分量使用的 Huffman 表 ID
3Bytes	固定值 0x003F00

双线内部分可以重复出现，依 ID 对颜色分量中的颜色进行描述

8.scanData

图像的压缩数据，为了不与之前的标记码（Tag）混淆，数据中遇到 0xFF 时，需要进行判断：

1. 0xFF00：表示 0xFF 是图像数据的组成部分
2. 0xFFD0~0xFFD7：RSTn 标记，遇到标记时，对差分解码变量进行重置（归 0）
3. 0xFFD9：图像结束标记，图像压缩数据至此结束

9.EOI

代表 JFIF 图像数据的结束，即文件结尾

2 Bytes	标记码 0xFFD9

注：当图像中出现连续的 0xFF 时，当作一个 0xFF 看待。

二、打开图像文件，并显示

实现打开 BMP 与 JPEG 格式的图像并显示的功能

代码：

```
//选项：文件-打开
private void ToolStripMenuItem_openimg_Click(object sender, EventArgs e)
{
    try
    {
```

```
        //打开窗口初始化
        OpenFileDialog open = new OpenFileDialog();
        open.InitialDirectory = ".";
        open.Filter = "BMP文件(*.bmp)|*.bmp|JPG文件(*.jpg)|*.jpg|BMP文件
(*.gif)|*.gif|PNG文件(*.png)|*.png";
        open.RestoreDirectory = true;
        //如果为”打开“选定文件
        if (open.ShowDialog() == DialogResult.OK)
        {
   //读取当前文件名
   curFileName = open.FileName;

   //使用Image.FromFile创建图像对象
   try
   {
       //创建临时Bitmap对象来获取图像数据
       Bitmap img = (Bitmap)Image.FromFile(curFileName);
       //利用临时Bitmap对象构造objBitmap对象
       objBitmap = new Bitmap(img);
       //左侧窗口显示图像
       this.pictureBox_old.Image = objBitmap;
       //销毁临时Bitmap对象，解除文件占用
       img.Dispose();
       //获取图像大小
       cursize = GetPictureBoxZoomSize(pictureBox_old);
       //右侧窗口显示图像
       //pictureBox_new.Image = objBitmap;
   }
   catch (Exception ex)
   {
       MessageBox.Show(ex.Message, "错误提示", MessageBoxButtons.OK,
MessageBoxIcon.Stop);
   }
        }
        //对窗体进行重新绘制
        Invalidate();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "错误提示", MessageBoxButtons.OK,
MessageBoxIcon.Stop);
    }
```

}

示意图：

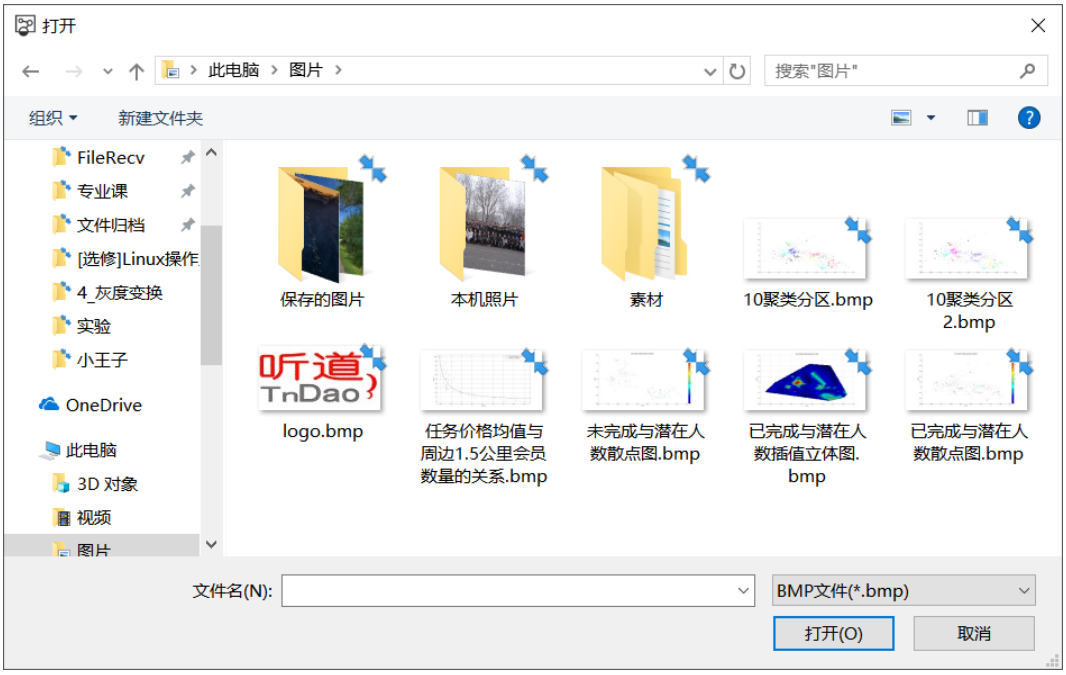

图 1：打开 BMP 文件

图 2：显示 BMP 文件

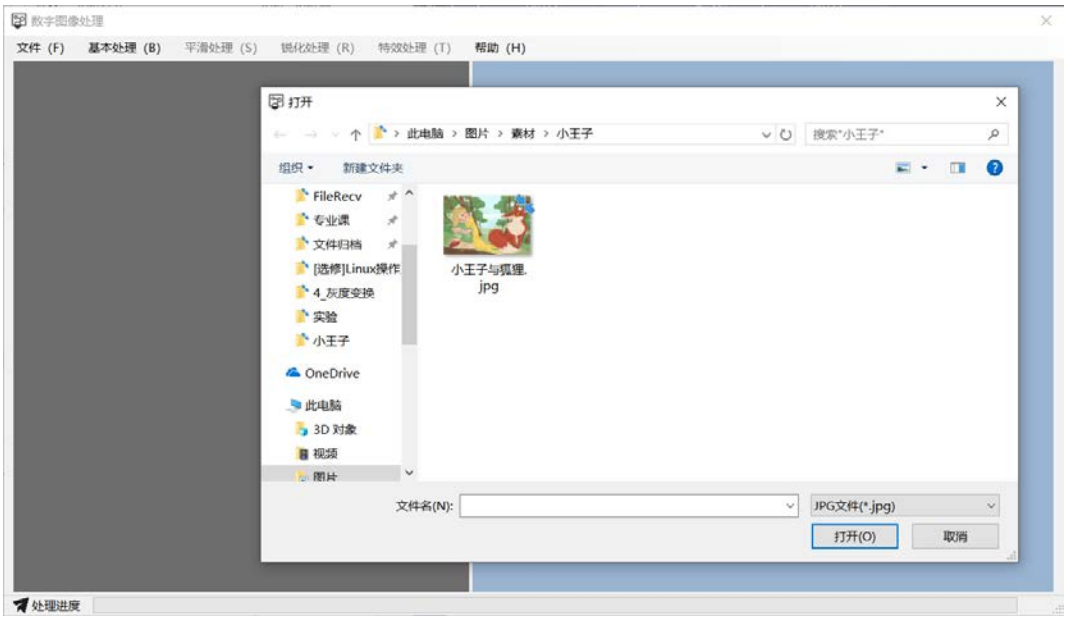

图 3：打开 JPG 文件

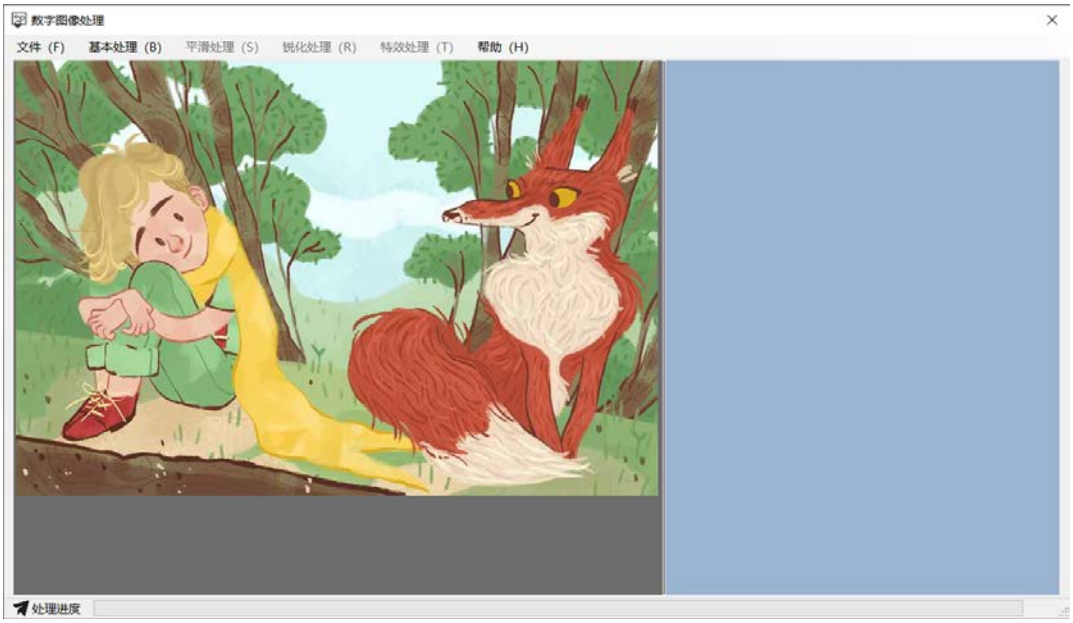

图 4：显示 JPG 文件

附录

参考文献：

1. JPG-JPEG（JFIF）文件解码—文件结构 - CSDN 博客
https://blog.csdn.net/ymlbright/article/details/44179891
2. BMP（图像文件格式(Bitmap)）_百度百科
https://baike.baidu.com/item/BMP/35116?fr=aladdin
3. JPEG_百度百科
https://baike.baidu.com/item/JPEG/213408?fr=aladdin
4. C#如何使用文件操作控件 [打开文件/保存文件]_百度经验
https://jingyan.baidu.com/article/e73e26c0c26c1a24adb6a7c1.html
5. C#图形编程 - 随笔分类 - 阿朵 - 博客园
https://www.cnblogs.com/llllll/category/191064.html