****

**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  1. **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** 2. 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](https://baike.baidu.com/item/%E4%BD%8D%E5%9B%BE" \t "_blank)（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。   1. 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。   1. 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。   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9 | 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，低位在前）      //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。   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | 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的倍数而添加的空字节)，以字节为单位（35-38字节）  LONG biXPelsPerMeter;//位图水平分辨率，每米像素数（39-42字节）  LONG biYPelsPerMeter;//位图垂直分辨率，每米像素数（43-46字节)  DWORD biClrUsed;//位图实际使用的颜色表中的颜色数（47-50字节）  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  2  3  4  5  6 | typedef struct tagRGBQUAD{  BYTE rgbBlue;//蓝色的亮度（值范围为0-255)  BYTE rgbGreen;//绿色的亮度（值范围为0-255)  BYTE rgbRed;//红色的亮度（值范围为0-255)  BYTE rgbReserved;//保留，必须为0  }\_\_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  2  3  4 | typedef struct tagBITMAPINFO{  BITMAPINFOHEADER bmiHeader;//位图信息头  RGBQUAD bmiColors[1];//颜色表  }\_\_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 .... .... |  1. **分析JPEG文件格式。** 2. JPEG格式简介   JPEG（Joint Photographic Experts Group）是在[国际标准化组织](https://baike.baidu.com/item/%E5%9B%BD%E9%99%85%E6%A0%87%E5%87%86%E5%8C%96%E7%BB%84%E7%BB%87" \t "_blank)(ISO)领导之下制定静态[图像压缩](https://baike.baidu.com/item/%E5%9B%BE%E5%83%8F%E5%8E%8B%E7%BC%A9" \t "_blank)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](https://baike.baidu.com/item/%E4%BA%92%E8%81%94%E7%BD%91" \t "_blank)和[数码相机](https://baike.baidu.com/item/%E6%95%B0%E7%A0%81%E7%9B%B8%E6%9C%BA)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](https://baike.baidu.com/item/%E5%AA%92%E4%BD%93" \t "_blank)上被封存起来。.jpeg/.jpg是最常用的图像[文件格式](https://baike.baidu.com/item/%E6%96%87%E4%BB%B6%E6%A0%BC%E5%BC%8F" \t "_blank)，由一个[软件](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6)开发联合会组织制定，是一种有损[压缩格式](https://baike.baidu.com/item/%E5%8E%8B%E7%BC%A9%E6%A0%BC%E5%BC%8F)，能够将[图像压缩](https://baike.baidu.com/item/%E5%9B%BE%E5%83%8F%E5%8E%8B%E7%BC%A9)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](https://baike.baidu.com/item/%E5%9B%BE%E5%83%8F%E6%95%B0%E6%8D%AE)的损伤。尤其是使用过高的压缩比例，将使最终[解压缩](https://baike.baidu.com/item/%E8%A7%A3%E5%8E%8B%E7%BC%A9)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](https://baike.baidu.com/item/%E5%8E%8B%E7%BC%A9%E6%8A%80%E6%9C%AF)十分先进，它用[有损压缩](https://baike.baidu.com/item/%E6%9C%89%E6%8D%9F%E5%8E%8B%E7%BC%A9)方式去除冗余的[图像数据](https://baike.baidu.com/item/%E5%9B%BE%E5%83%8F%E6%95%B0%E6%8D%AE)，在获得极高的[压缩率](https://baike.baidu.com/item/%E5%8E%8B%E7%BC%A9%E7%8E%87)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](https://baike.baidu.com/item/%E4%BD%8D%E5%9B%BE)[文件压缩](https://baike.baidu.com/item/%E6%96%87%E4%BB%B6%E5%8E%8B%E7%BC%A9)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](https://baike.baidu.com/item/JPEG%E6%A0%BC%E5%BC%8F" \t "_blank)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](https://baike.baidu.com/item/%E4%BA%92%E8%81%94%E7%BD%91)，可减少图像的传输时间，可以支持24bit[真彩色](https://baike.baidu.com/item/%E7%9C%9F%E5%BD%A9%E8%89%B2)，也普遍应用于需要连续色调的图像。   1. 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](http://blog.csdn.net/lpt19832003/archive/2007/07/28/Editor/FCKeditor/editor/fckeditor.html?InstanceName=ctl00_ContentPlaceHolder1_EntryEditor1_richTextEditor_richTextEditor&Toolbar=Default#SOI)、[APP0](http://blog.csdn.net/lpt19832003/archive/2007/07/28/Editor/FCKeditor/editor/fckeditor.html?InstanceName=ctl00_ContentPlaceHolder1_EntryEditor1_richTextEditor_richTextEditor&Toolbar=Default" \l "APP0" \t "_blank)、[DQT](http://blog.csdn.net/lpt19832003/archive/2007/07/28/Editor/FCKeditor/editor/fckeditor.html?InstanceName=ctl00_ContentPlaceHolder1_EntryEditor1_richTextEditor_richTextEditor&Toolbar=Default" \l "DQT" \t "_blank)、[SOF0](http://blog.csdn.net/lpt19832003/archive/2007/07/28/Editor/FCKeditor/editor/fckeditor.html?InstanceName=ctl00_ContentPlaceHolder1_EntryEditor1_richTextEditor_richTextEditor&Toolbar=Default" \l "SOF" \t "_blank)、[DHT](http://blog.csdn.net/lpt19832003/archive/2007/07/28/Editor/FCKeditor/editor/fckeditor.html?InstanceName=ctl00_ContentPlaceHolder1_EntryEditor1_richTextEditor_richTextEditor&Toolbar=Default" \l "DHT" \t "_blank)、[DRI](http://blog.csdn.net/lpt19832003/archive/2007/07/28/Editor/FCKeditor/editor/fckeditor.html?InstanceName=ctl00_ContentPlaceHolder1_EntryEditor1_richTextEditor_richTextEditor&Toolbar=Default" \l "DRI" \t "_blank)、[SOS](http://blog.csdn.net/lpt19832003/archive/2007/07/28/Editor/FCKeditor/editor/fckeditor.html?InstanceName=ctl00_ContentPlaceHolder1_EntryEditor1_richTextEditor_richTextEditor&Toolbar=Default" \l "SOS" \t "_blank)、[EOI](http://blog.csdn.net/lpt19832003/archive/2007/07/28/Editor/FCKeditor/editor/fckeditor.html?InstanceName=ctl00_ContentPlaceHolder1_EntryEditor1_richTextEditor_richTextEditor&Toolbar=Default" \l "EOI" \t "_blank)。  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） |  1. 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**](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**](https://baike.baidu.com/item/BMP/35116?fr=aladdin)

**3.** **JPEG\_百度百科**

[**https://baike.baidu.com/item/JPEG/213408?fr=aladdin**](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**](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**](https://www.cnblogs.com/llllll/category/191064.html)