

深圳大学实验报告

课程名称： 数字图像处理

实验项目名称： 图像显示实验

学院： 计算机与软件学院

专业： 计算机科学与技术

指导教师： 吴惠思 教授

报告人： 林浩晟 学号： 2022280310

实验时间： 2025 年 3 月 10 日星期一

实验报告提交时间： 2025 年 3 月 10 日星期一

教务部制

一、实验目的：

1. 了解 BMP 图像的基本格式
2. 掌握无格式灰度图像的显示原理和实现方法
3. 掌握 24 位 BMP 彩色图像的显示原理和实现方法

二、实验原理：

- 1、生成一个 Window 运行程序
- 2、打开并读取图像
- 3、逐个像素显示无格式 RAW 图像
- 4、主函数中，调用语句集合

三、实验用品：

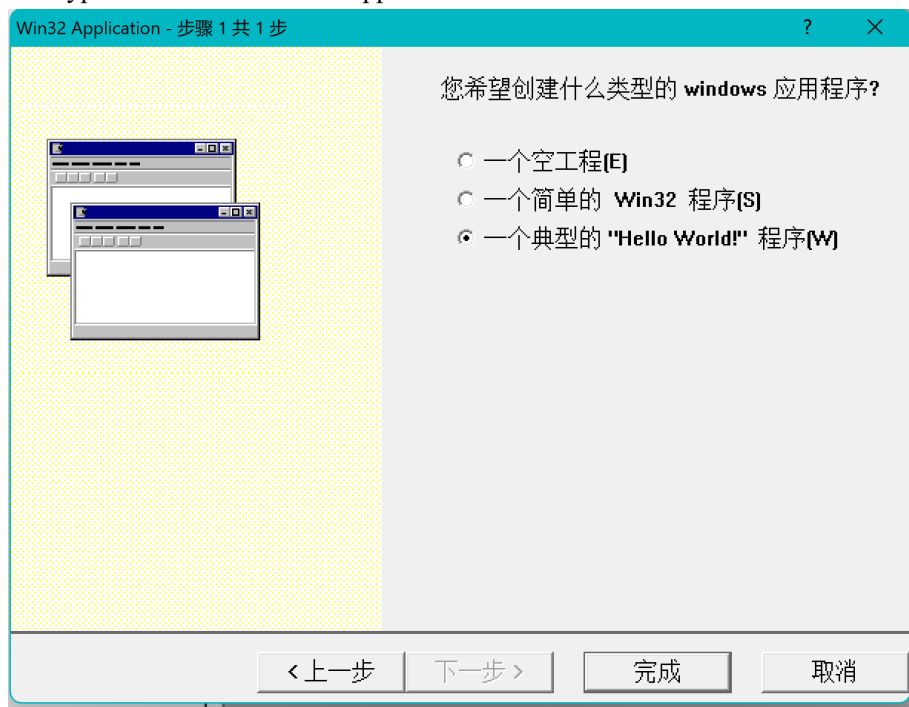
计算机、visual C++6.0

四、实验过程及内容：

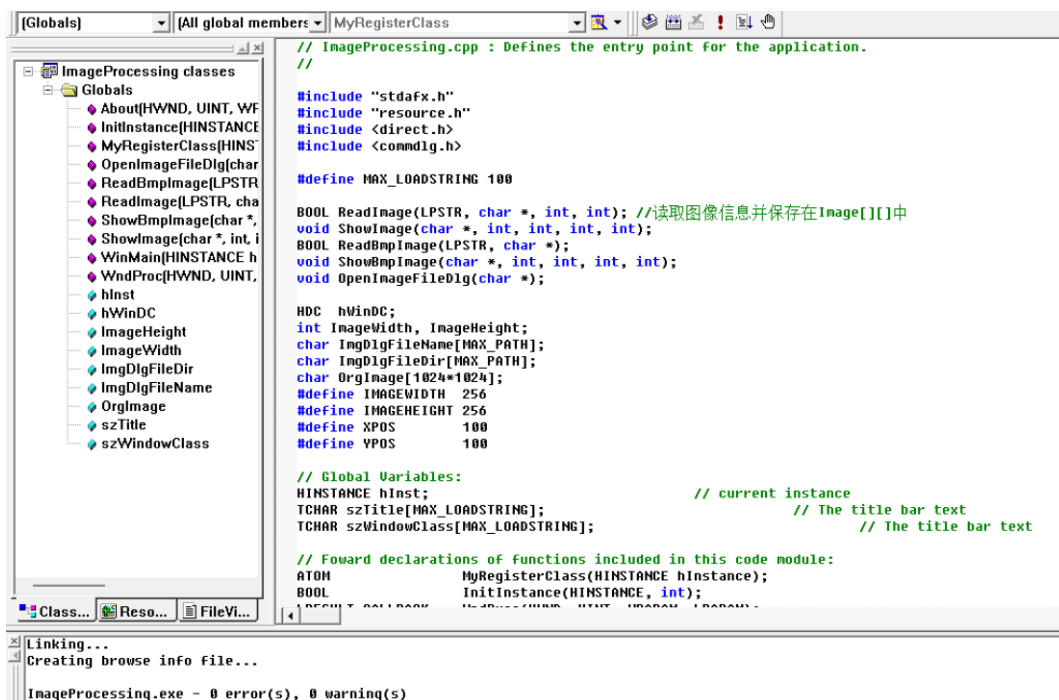
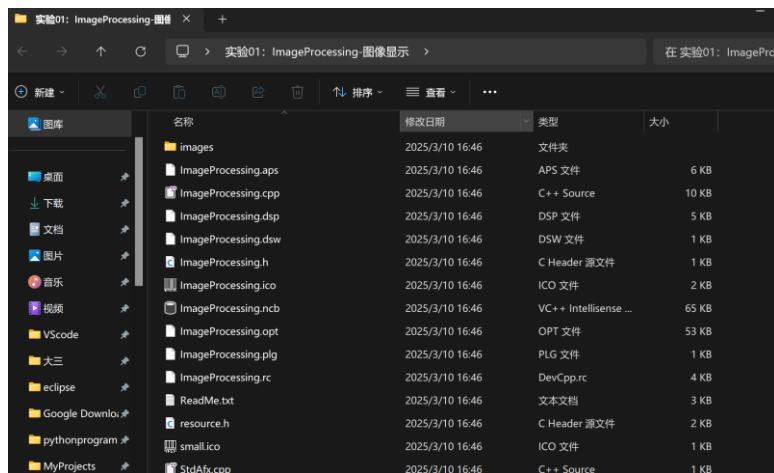
1. 首先下载好 visual6.0C++, 再根据 ppt 进行操作, 首先点击 File 菜单项, 再点击 New, 选择 Win32 Application, 输入工程名称再创建;



2. 选择 A typical "Hello World!" Application;



3. 再将实验所提供的文件解压后 copy 至工作空间中, 打开后如下图所示;



4. 再根据 PPT 中的代码提示完善显示 RAW 格式图像的代码；首先注意到各个函数参数的含义，Image: 图像数据数组，wImage: 图像的宽度（像素数），hImage: 图像的高度（像素数，xPos、yPos: 图像在窗口中的显示位置；随后检查窗口设备上下文 (hWinDC) 是否有效，如果无效，直接返回；最后用二层循环遍历像素的行列，获取像素的灰度值并且绘制出来。

```
void ShowImage(char *Image, int wImage, int hImage, int xPos, int yPos)
{
    //请在下方填入显示RAW格式图像具体代码
    if (!hWinDC) return; // 确保窗口设备上下文有效

    for (int i = 0; i < hImage; i++) {
        for (int j = 0; j < wImage; j++) {
            BYTE pixel = Image[i * wImage + j]; // 取灰度值
            SetPixel(hWinDC, j + xPos, i + yPos, RGB(pixel, pixel, pixel));
        }
    }
}
```

5. 完善显示 bmp 格式图像的代码；首先初始化 BITMAPINFO 结构，再设置位图信息

头，包括位图头的大小、图像的宽度、高度、平面数、每像素的比特数等；最后使用 SetDIBitsToDevice 函数将位图显示到窗口。

```
void ShowBmpImage(char *Image, int wImage, int hImage, int xPos, int yPos)
{
    //请在下方填入显示BMP格式图像具体代码

    BITMAPINFO bmi;
    memset(&bmi, 0, sizeof(BITMAPINFO));
    bmi.bmiHeader.biSize = sizeof(BITMAPINFOHEADER);
    bmi.bmiHeader.biWidth = wImage;
    bmi.bmiHeader.biHeight = hImage;
    bmi.bmiHeader.biPlanes = 1;
    bmi.bmiHeader.biBitCount = 24;
    bmi.bmiHeader.biCompression = BI_RGB;

    SetDIBitsToDevice(hWinDC, xPos, yPos, wImage, hImage, 0, 0, 0, hImage, Image, &bmi, DIB_RGB_COLORS);
}
```

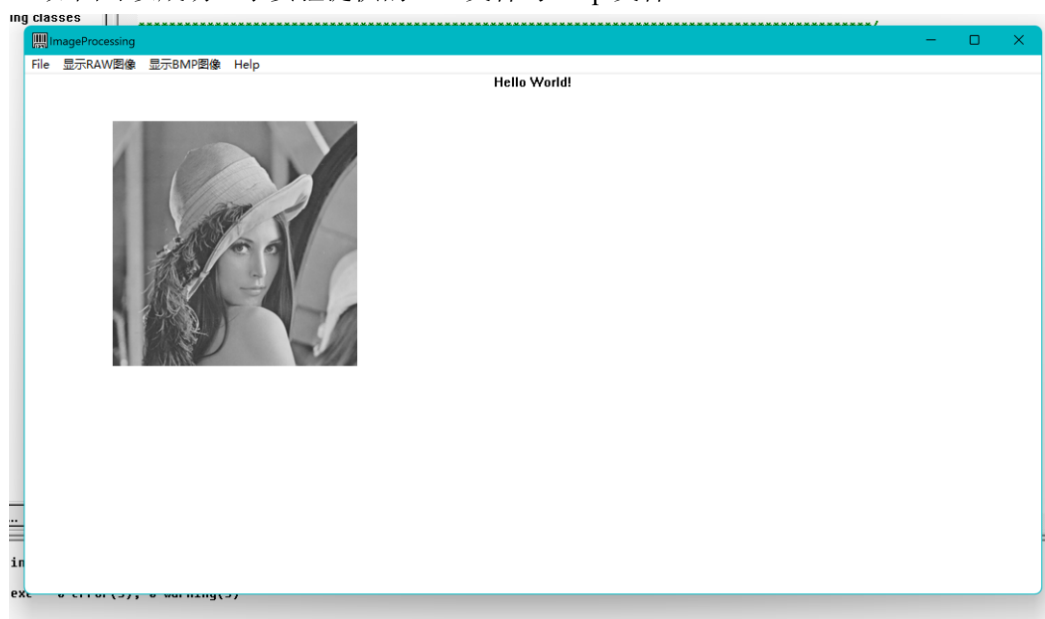
6. 打开 bmp 图像时发现打开文件对话框中不支持 BMP 文件的显示，因此对 OpenImageFileDialog 函数进行了修改，让程序可以支持打开 bmp 和 raw 两种文件。

```
void OpenImageFileDialog(char *OPDLTitle)
{
    char FileTitle[100];
    OPENFILENAME ofn;

    memset(&ofn, 0, sizeof(ofn));
    ofn.lStructSize = sizeof(OPENFILENAME);
    ofn.hwndOwner = NULL;
    ofn.nInstance = 0;
    ofn.lpstrFilter = TEXT("RAW Files (*.raw)\\0*.raw\\0BMP Files (*.bmp)\\0*.bmp\\0All Files (*.*)\\0*.*\\0"); // 支持 RAW 和 BMP
    ofn.lpstrCustomFilter = NULL;
    ofn.nMaxCustFilter = 0;
    ofn.nFilterIndex = 1;
    ofn.lpstrFile = ImgDlgFileName;
    ofn.nMaxFile = MAX_PATH;
    ofn.lpstrFileTitle = FileTitle;
    ofn.nMaxFileTitle = 99;
    ofn.lpstrInitialDir = ImgDlgFileDir;
    ofn.lpstrTitle = OPDLTitle;
    ofn.Flags = OFN_FILEMUSTEXIST | OFN_PATHMUSTEXIST;
    ofn.lpstrDefExt = "raw";

    ImgDlgFileName[0] = '\\0';
    if (GetOpenFileName(&ofn)) {
        getcwd(ImgDlgFileDir, MAX_PATH); // 获取当前目录
    }
}
```

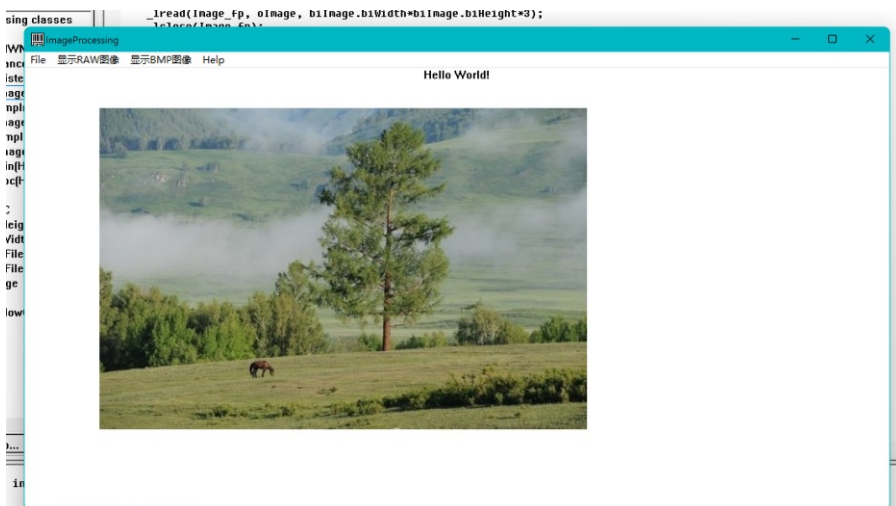
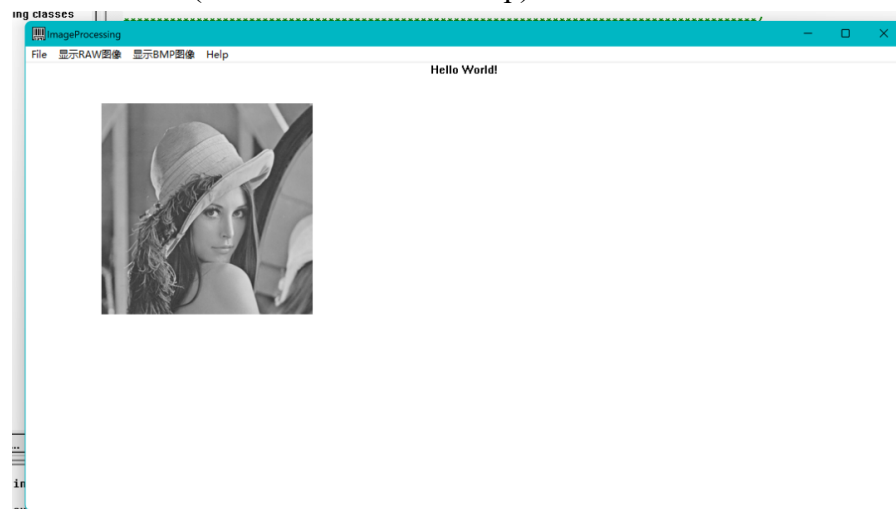
7. 如图可以成功显示实验提供的.raw 文件与.bmp 文件。



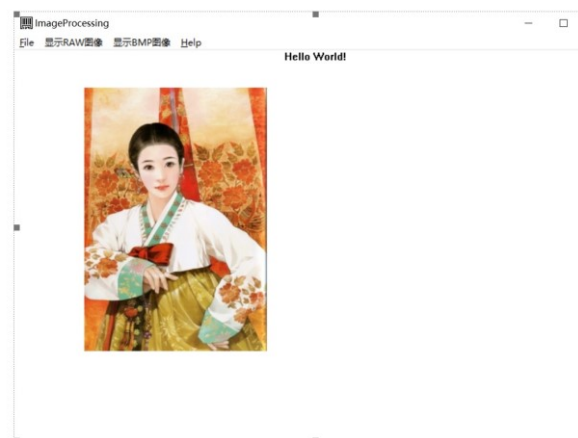


五、实验现象及数据处理：

实验中的两个图片(lena256.raw 与 tree.bmp)显示结果如下：



同时在实验的“实验程序框架运行指南”文档中的以下图片，在实验提供的图片中没有，因此不予展示。



六、实验结论：

实验成功完成！实验中提供的两张图片都通过 visual6.0C++成功显示。

实验感想：

通过本次实验，我对图像格式有了更深入的理解。在 Visual C++ 6.0 环境下，成功显示了 .raw 和 .bmp 图像。在研究 BMP 图像格式时，我了解到其文件头、信息头和像素数据的结构，这让我清晰地认识到图像存储的细节。对于无格式灰度图像的显示，我掌握了通过逐像素读取和映射灰度值来实现显示的方法，这让我对图像的底层表示有了更直观的认识。在处理 24 位 BMP 彩色图像时，我学会了如何解析 RGB 分量并正确显示彩色图像。整个实验过程不仅锻炼了我的编程能力，还让我对图像处理的基础知识有了更扎实的掌握，为后续的学习打下了坚实的基础。

<p>思考题：</p>
<p>指导教师批阅意见：</p> <p>成绩评定：</p> <p>指导教师签字：</p> <p>年 月 日</p>
<p>备注：</p>

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。