

暨南大学本科实验报告专用纸

课程名称 数字图像处理 成绩评定

实验项目名称 图像的频域滤波 指导教师 刘晓翔

实验项目编号 05 实验项目类型 综合型 实验地点 三楼机房

学生姓名 赵俊文 学号 2022104002

学院 智能科学与工程学院 系 专业 人工智能

实验时间 2024 年 12 月 9 日 上 午 ~ 12 月 23 日 上 午

（一） 实验目的

①熟悉图像频域滤波中高通和低通滤波器；②实现理想、巴特沃斯高/低通滤波器。

（二） 实验内容和要求

应利用 Visual C++6.0 软件开发工具编写程序，实现理想、巴特沃斯高/低通滤波器，程序执行结果正确。

（三） 主要仪器设备

仪器：计算机

实验环境： Windows XP + Visual C++6.0

（四） 实验步骤（附代码）与调试

1.巴特沃斯高/低通滤波器（ $D>0$ 表示低通滤波， $D<0$ 表示高通滤波）

a.在 bmp 文件中添加函数，在工具栏中添加对应按钮，函数代码如下

//巴特沃斯

```
void FFT_Filter(int D){  
    int w = lpDIB_FT->bmiHeader.biWidth;  
    int h = lpDIB_FT->bmiHeader.biHeight;  
    int LineBytes = (w * lpDIB_FT->bmiHeader.biBitCount + 31)/32 * 4;  
    BYTE* lpBits =  
(BYTE*)&lpDIB_FT->bmiColors[lpDIB_FT->bmiHeader.biClrUsed];
```

```

complex<double>* Backup_FD = new complex<double>[w * h];
int i,j;
for(i = 0;i < w * h; i++){
    Backup_FD[i] = gFD[i];
}

double dis;
for(i = 0;i < h; i++){
    for(j = 0;j < w;j++){
        dis = sqrt((i - h / 2) * (i - h / 2) + (j - w / 2) * (j - w
/ 2));
        if(D > 0)
            gFD[i * w + j] *= 1 / (1 + pow(dis / D,4));

        else
            gFD[i * w + j] *= 1 / (1 + pow(-D/dis ,4));

    }
}

double temp;
BYTE* pixel;
for(i = 0;i < h; i++){
    for(j = 0;j < w;j++){
        pixel = lpBits + LineBytes * (h - 1 - i) + j;
        temp = sqrt(gFD[j * h + i].real() * gFD[j * h + i].real() +
            gFD[j * h + i].imag() * gFD[j * h + i].imag()) * 2000;

        if (temp > 255)
            temp = 255;
        *pixel = (BYTE) (temp);

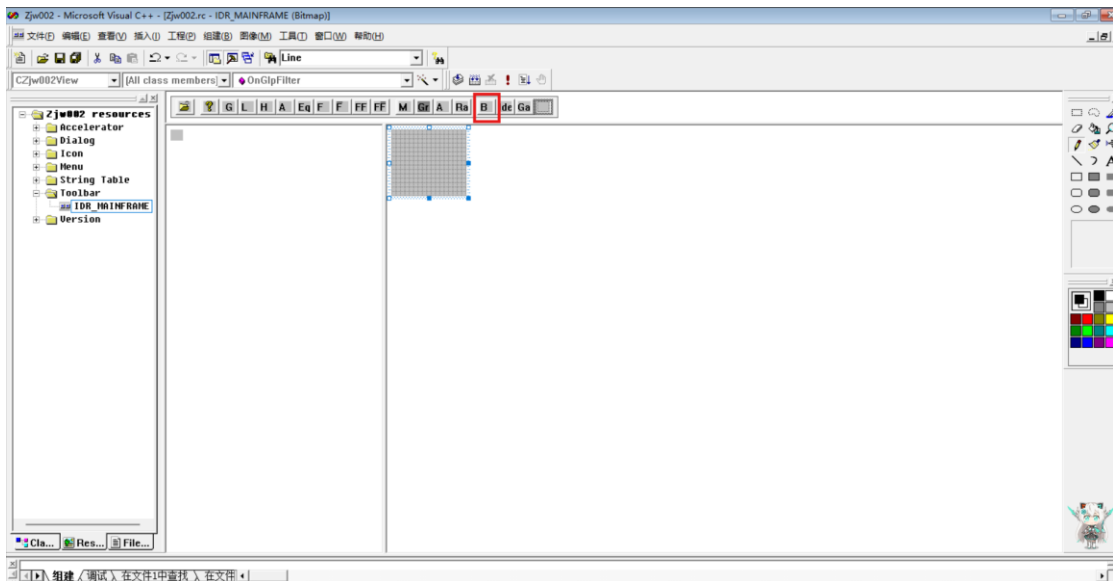
    }
}

IFFourier();

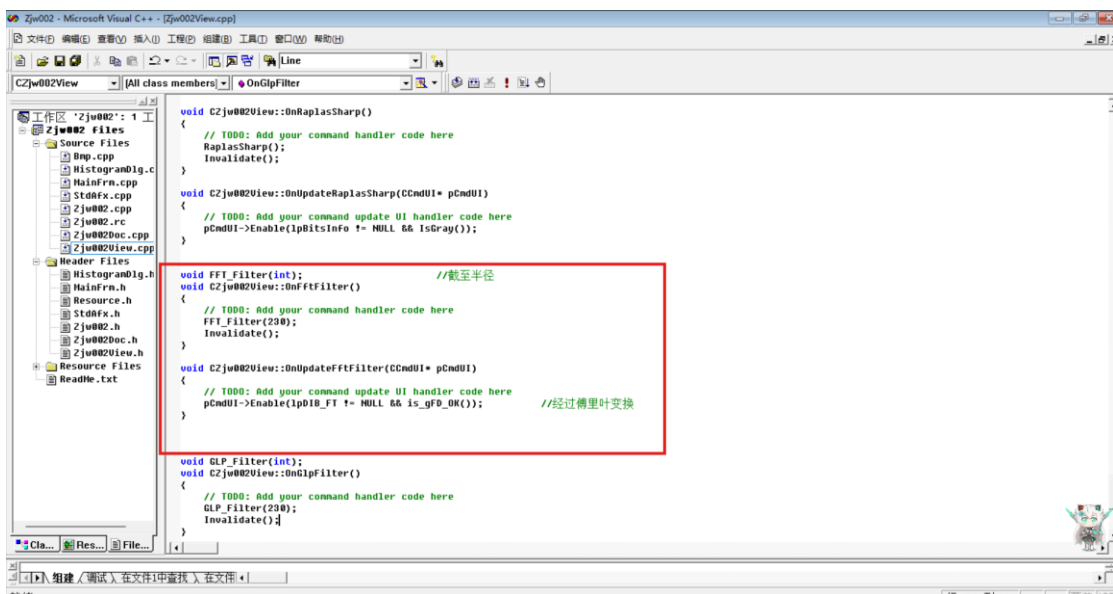
delete gFD;
gFD = Backup_FD;

}

```



b.在视图类中添加按钮的响应函数，并在函数中调用上述巴特沃斯函数



2. 高斯高/低通滤波器（ $D > 0$ 表示低通滤波， $D < 0$ 表示高通滤波）

a. 在 bmp 文件中添加函数，在工具栏中添加对应按钮，函数代码如下

```
void GLP_Filter(int D){
    int w = lpDIB_FT->bmiHeader.biWidth;
    int h = lpDIB_FT->bmiHeader.biHeight;
    int LineBytes = (w * lpDIB_FT->bmiHeader.biBitCount + 31)/32 * 4;
    BYTE* lpBits =
    (BYTE*)&lpDIB_FT->bmiColors[lpDIB_FT->bmiHeader.biClrUsed];

    complex<double>* Backup_FD = new complex<double>[w * h];
```

```

int i,j;
for(i = 0;i < w * h; i++){
    Backup_FD[i] = gFD[i];
}

double dis;
for(i = 0;i < h; i++){
    for(j = 0;j < w;j++){
        dis = sqrt((i - h / 2) * (i - h / 2) + (j - w / 2) * (j - w
/ 2));
        if(D > 0) {
            gFD[i * w + j] *= exp(-1 * pow(dis, 2) / (2 * pow(D, 2)));
        }
        else {
            gFD[i * w + j] *= 1 - exp(-1 * pow(dis, 2) / (2 * pow(D,
2)));
        }
    }
}

double temp;
BYTE* pixel;
for(i = 0;i < h; i++){
    for(j = 0;j < w;j++){
        pixel = lpBits + LineBytes * (h - 1 - i) + j;
        temp = sqrt(gFD[j * h + i].real() * gFD[j * h + i].real() +
            gFD[j * h + i].imag() * gFD[j * h + i].imag()) * 2000;

        if (temp > 255)
            temp = 255;
        *pixel = (BYTE) (temp);

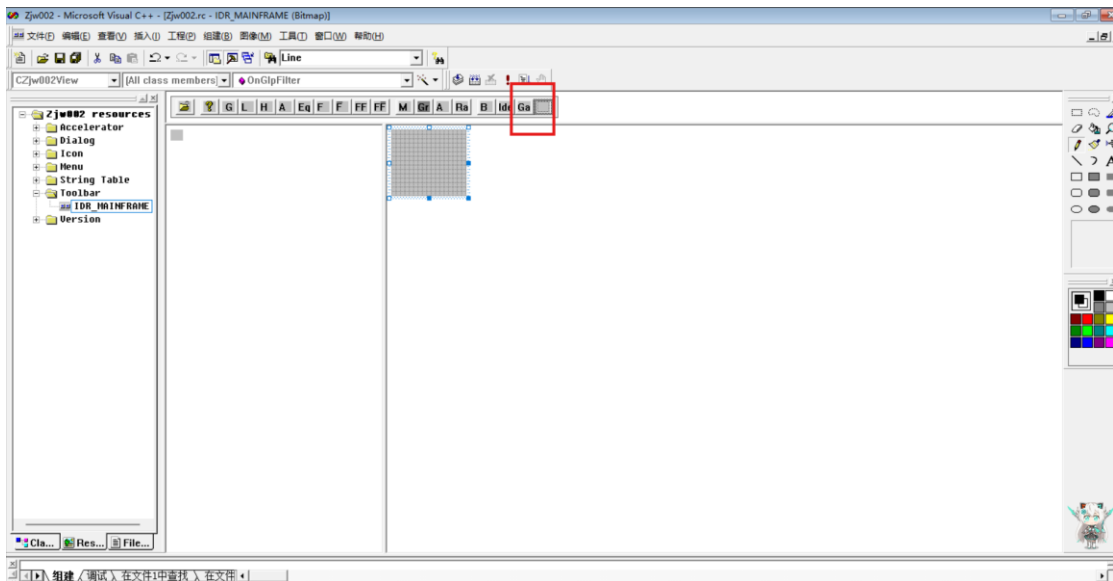
    }
}

IFFourier();

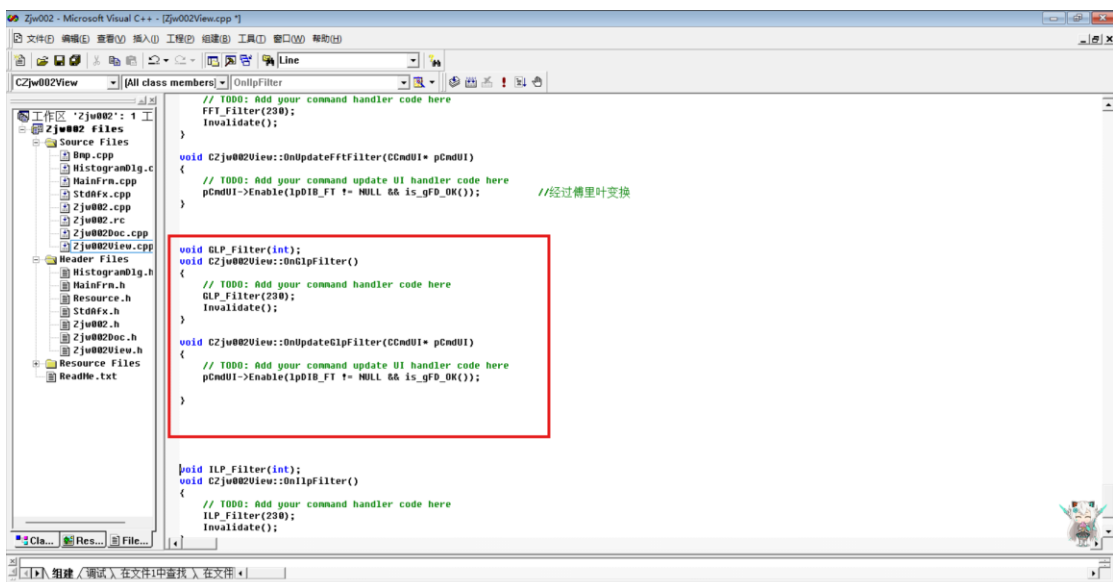
delete gFD;
gFD = Backup_FD;

}

```



b.在视图类中添加按钮的响应函数，并在函数中调用上述函数



3.理想高/低通滤波器（ $D > 0$ 表示低通滤波， $D < 0$ 表示高通滤波）

a.在 bmp 文件中添加函数，在工具栏中添加对应按钮，函数代码如下

```
void ILP_Filter(int D){
    int w = lpDIB_FT->bmiHeader.biWidth;
    int h = lpDIB_FT->bmiHeader.biHeight;
    int LineBytes = (w * lpDIB_FT->bmiHeader.biBitCount + 31)/32 * 4;
    BYTE* lpBits =
    (BYTE*)&lpDIB_FT->bmiColors[lpDIB_FT->bmiHeader.biClrUsed];

    complex<double>* Backup_FD = new complex<double>[w * h];
```

```

int i,j;
for(i = 0;i < w * h; i++){
    Backup_FD[i] = gFD[i];
}

double dis;
for(i = 0;i < h; i++){
    for(j = 0;j < w;j++){
        dis = sqrt((i - h / 2) * (i - h / 2) + (j - w / 2) * (j - w
/ 2));

        // 保留低频
        if (D > 0) {
            if(dis <= D)
                gFD[i * w + j] *= 1;

            else
                gFD[i * w + j] = 0;
        }
        // 保留高频
        else {
            if(dis <= -1 * D)
                gFD[i * w + j] = 0;

            else
                gFD[i * w + j] *= 1;
        }
    }
}

double temp;
BYTE* pixel;
for(i = 0;i < h; i++){
    for(j = 0;j < w;j++){
        pixel = lpBits + LineBytes * (h - 1 - i) + j;
        temp = sqrt(gFD[j * h + i].real() * gFD[j * h + i].real() +
                    gFD[j * h + i].imag() * gFD[j * h + i].imag()) * 2000;

        if (temp > 255)
            temp = 255;
        *pixel = (BYTE) (temp);
    }
}

```

```

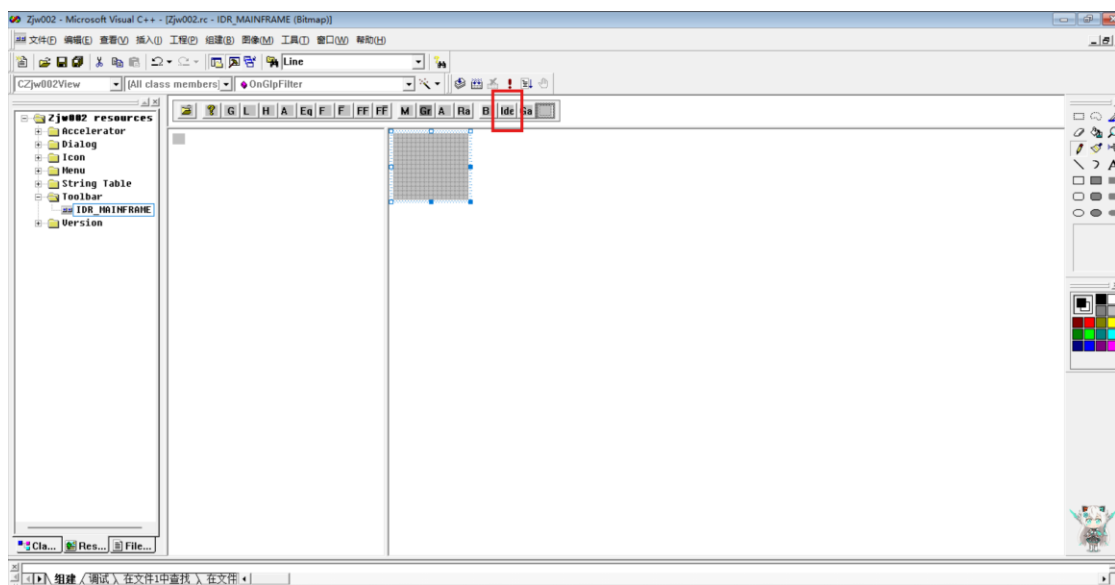
}

IFFourier();

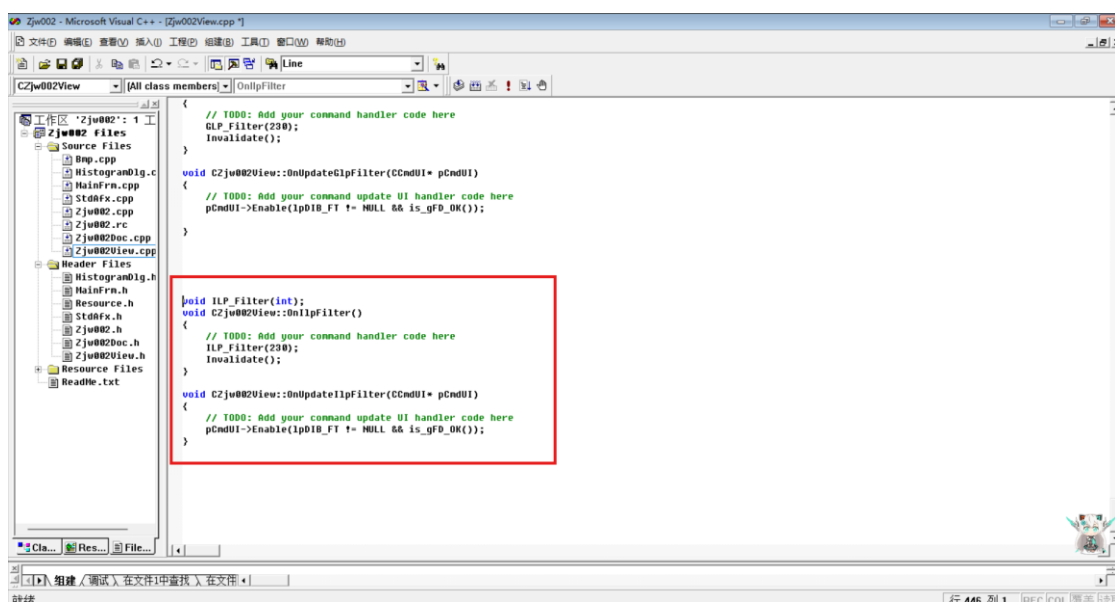
delete gFD;
gFD = Backup_FD;

}

```



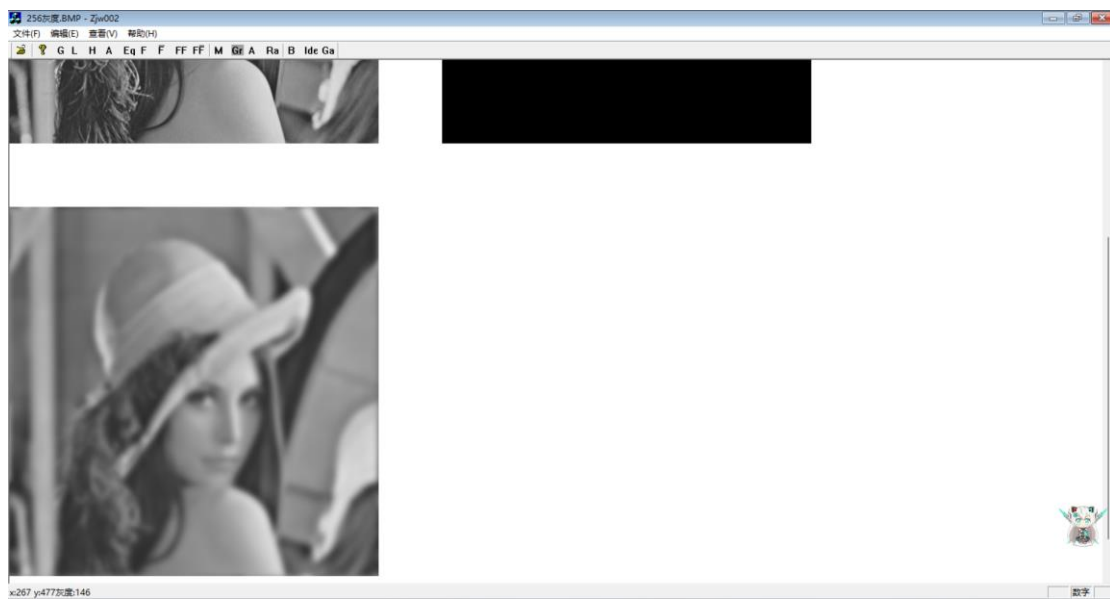
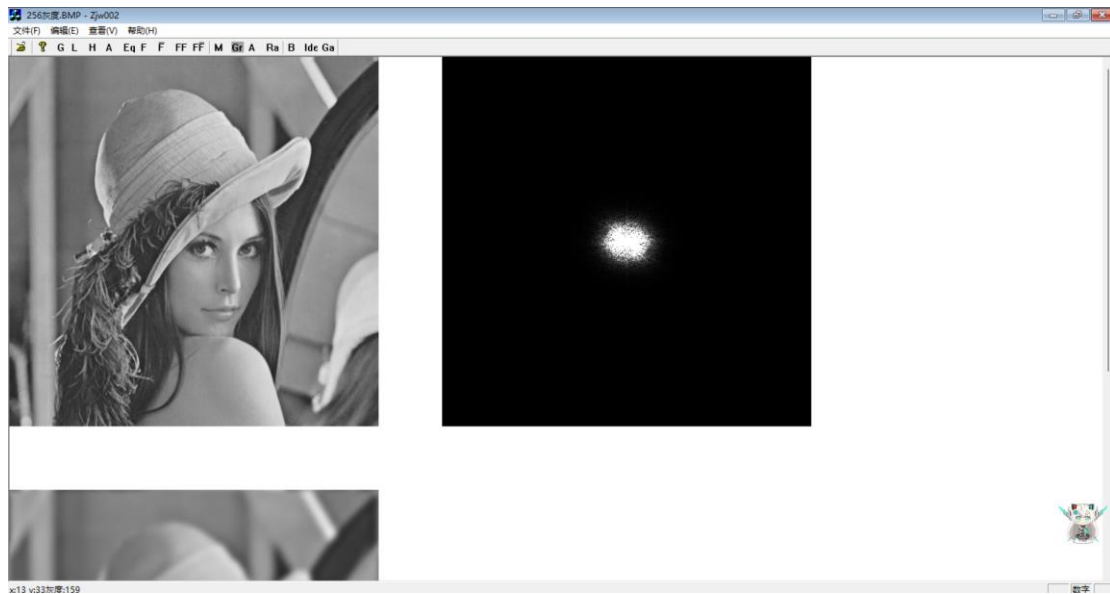
b.在视图类中添加按钮的响应函数，并在函数中调用上述函数



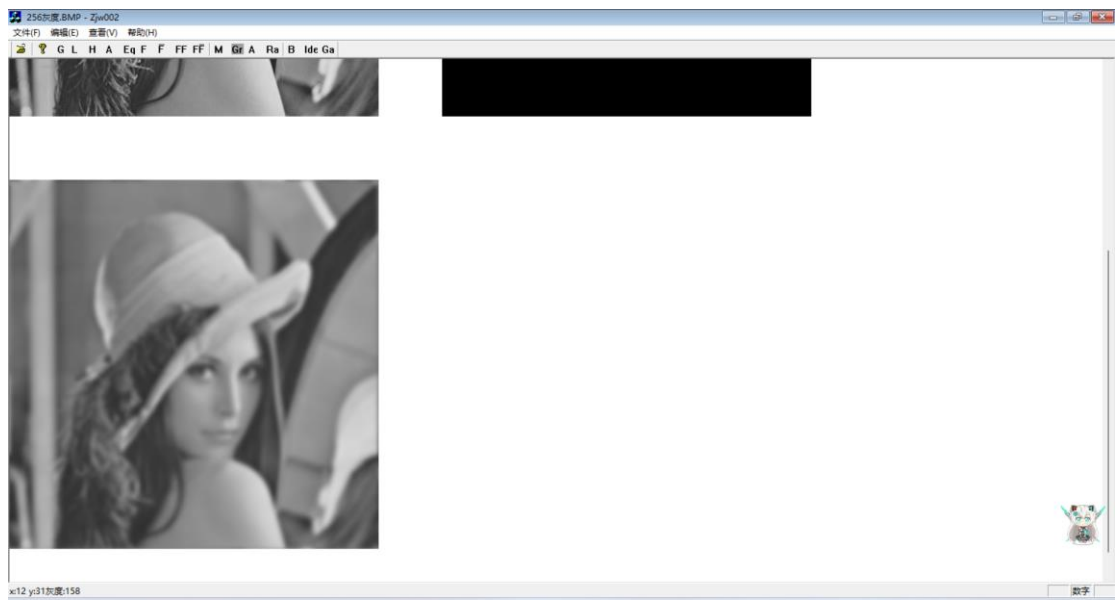
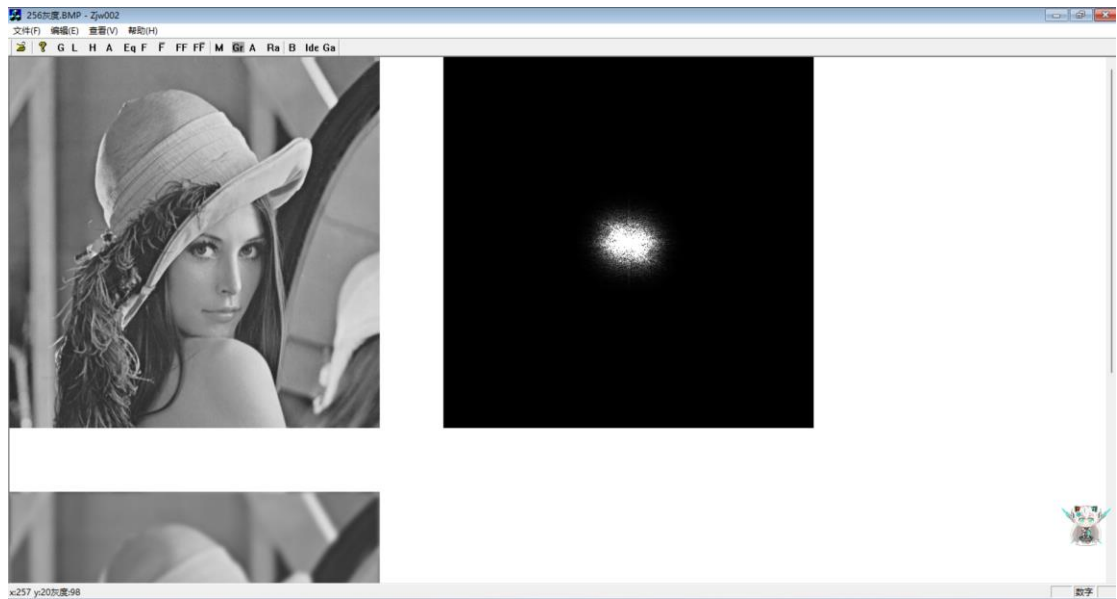
(五) 实验结果与分析

1. 低通滤波器的比较(D=25)

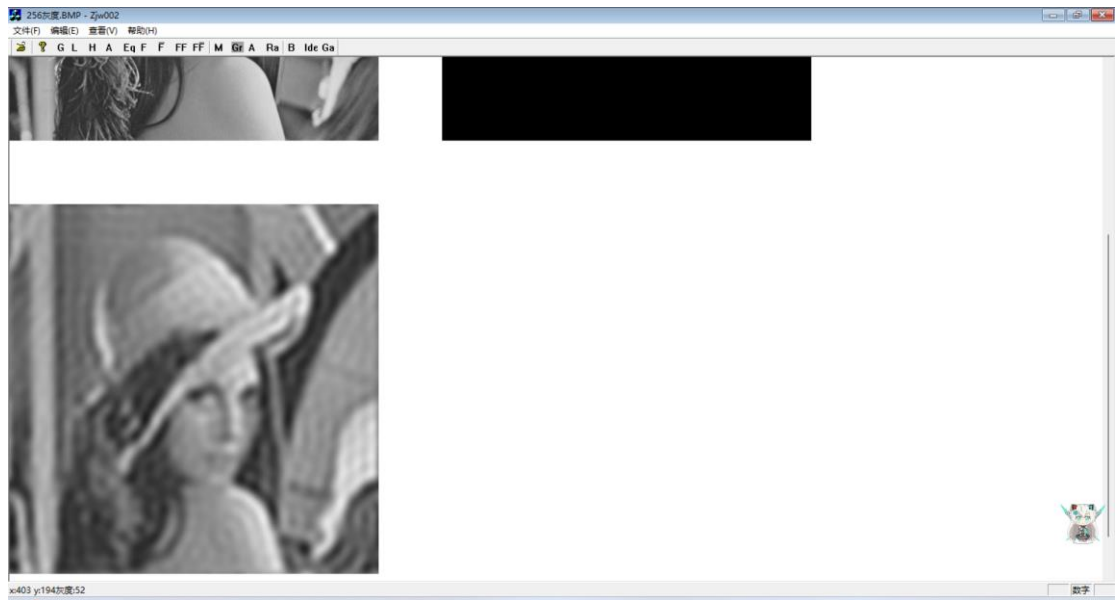
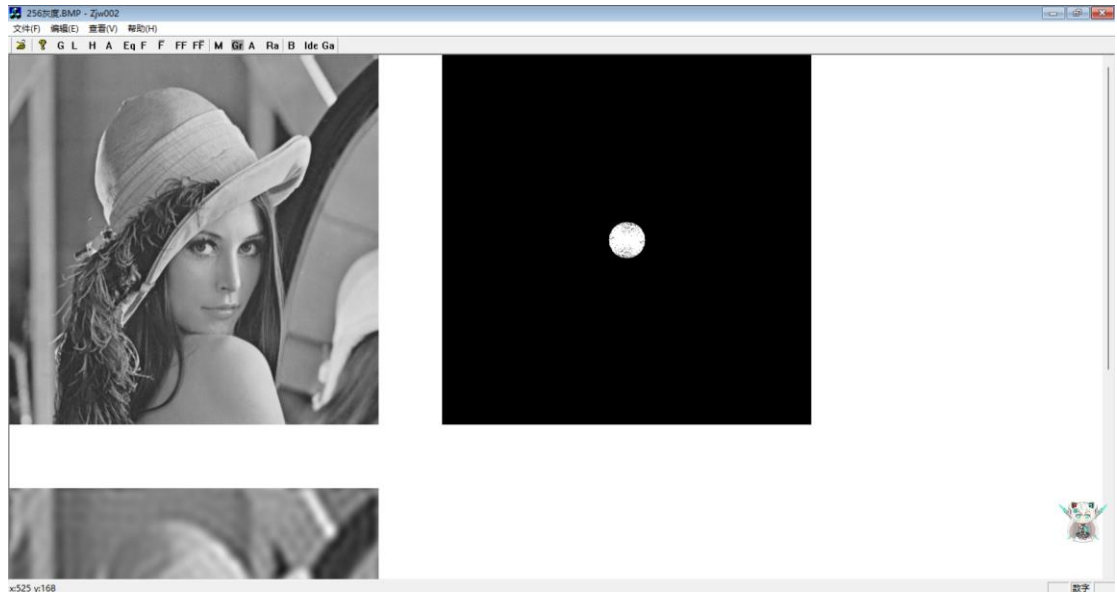
a. 巴特沃斯低通滤波器



b. 高斯低通滤波器

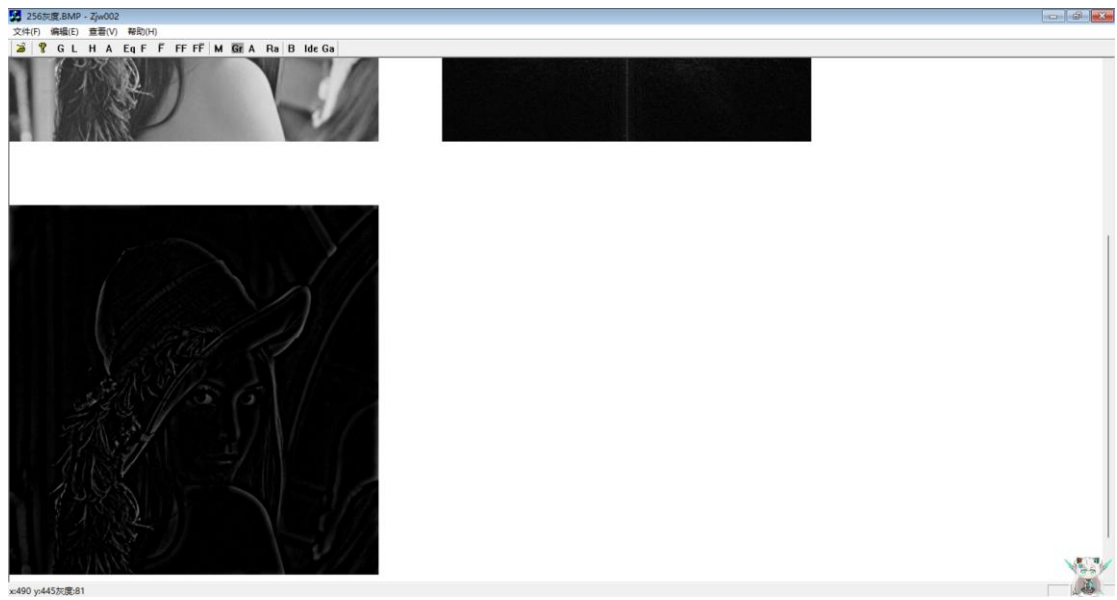
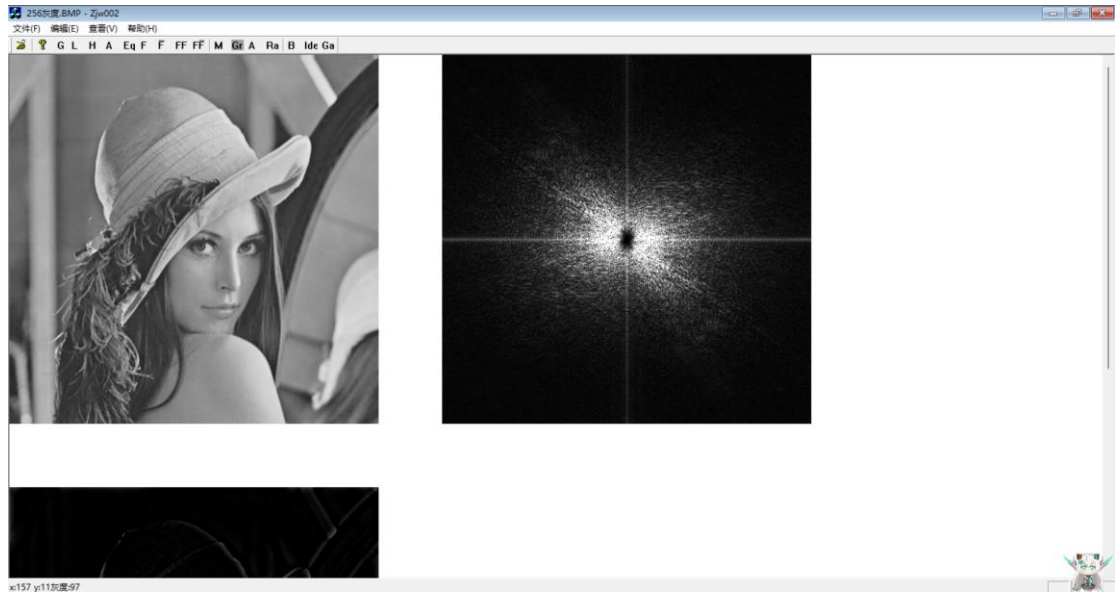


c. 理想低通滤波器

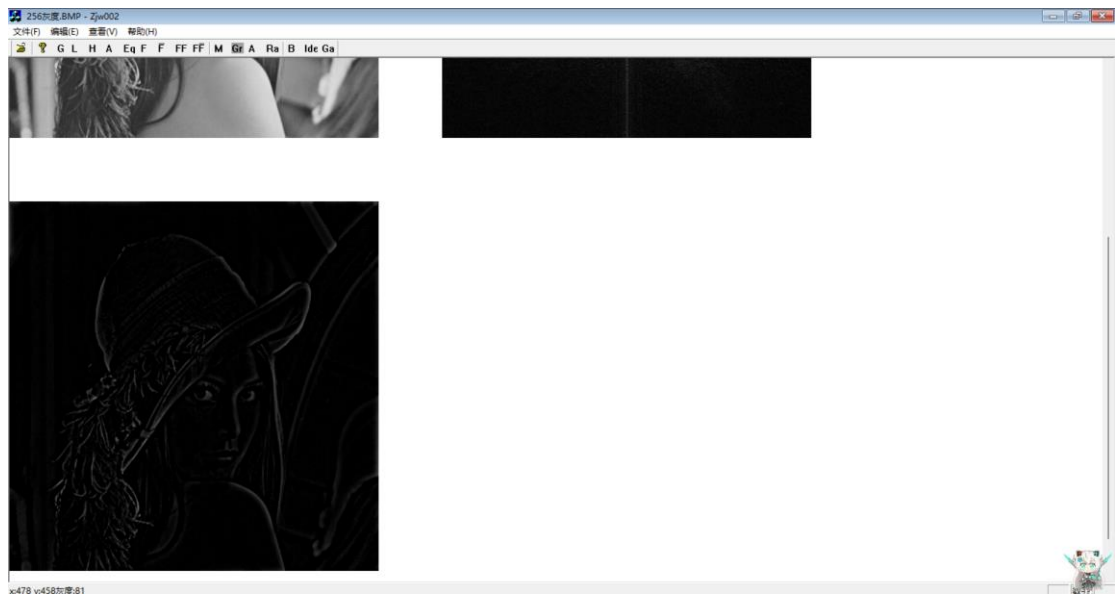
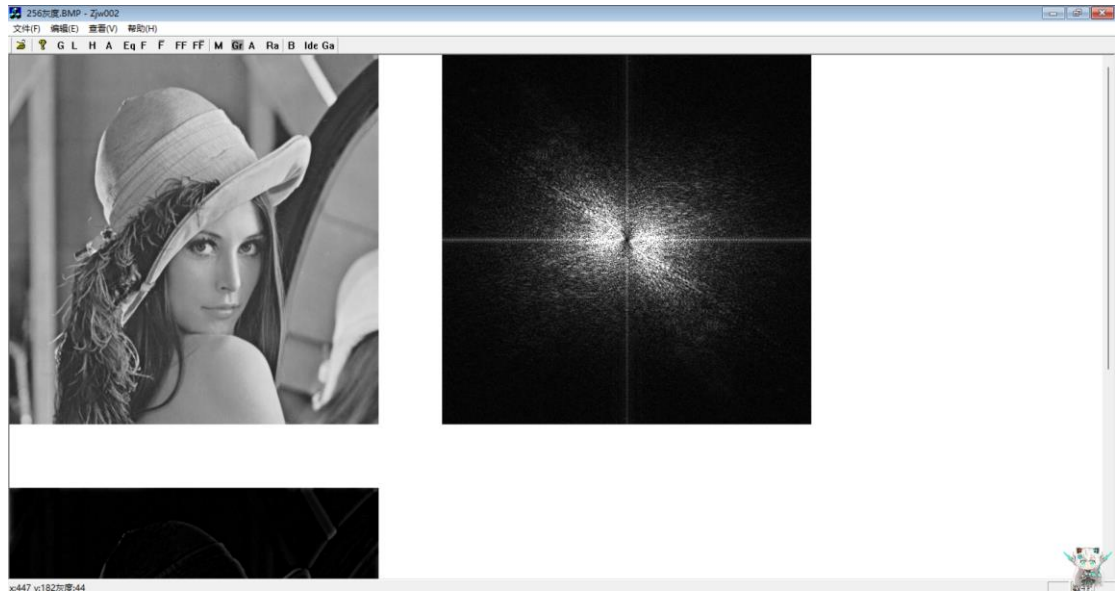


2. 高通滤波器的比较(D=-20)

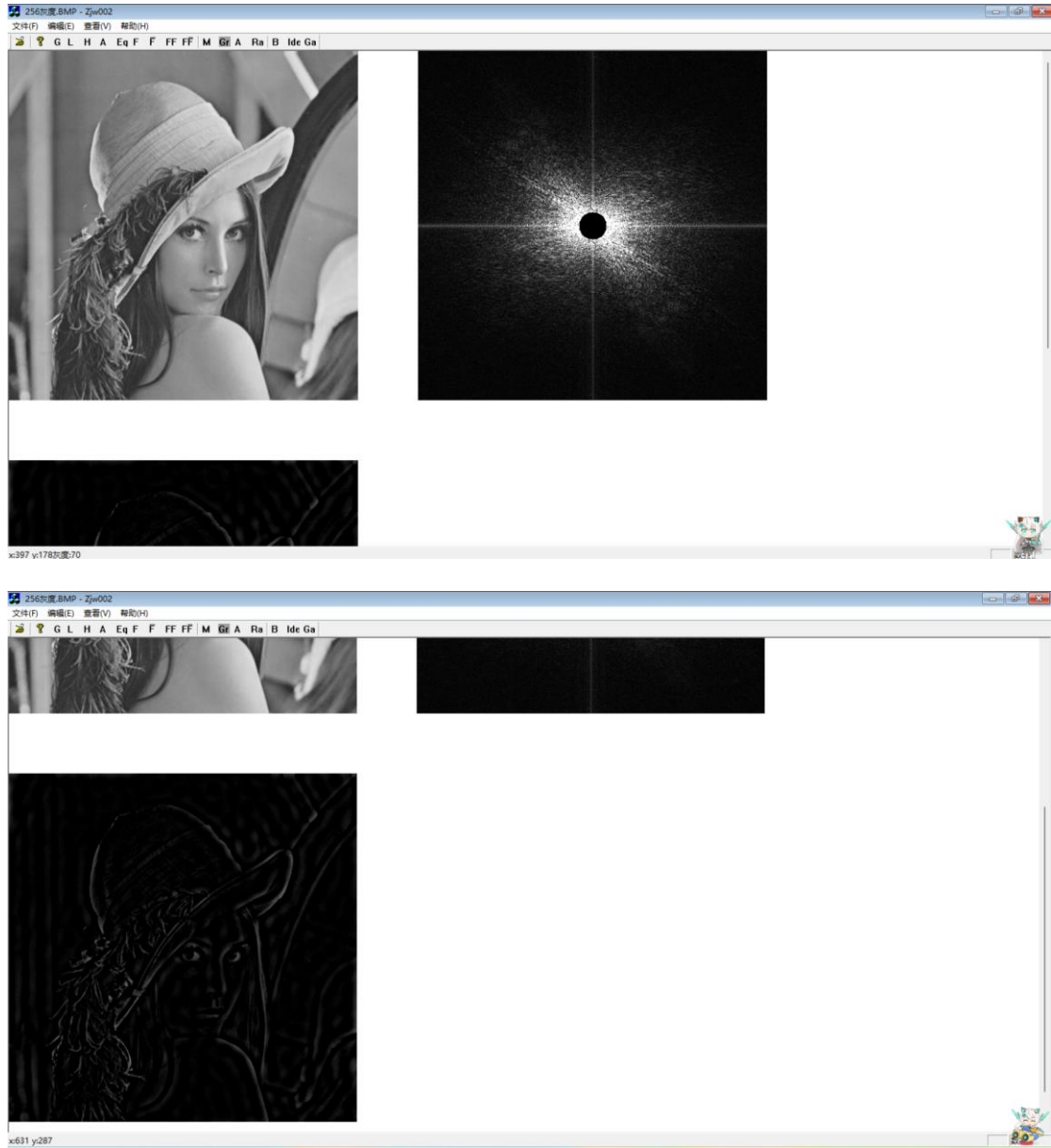
a. 巴特沃斯高通滤波器



b. 高斯高通滤波器



c. 理想高通滤波器



4.实验分析

通过本次实验，我掌握了图像的频域滤波，用代码实现了理想、巴特沃斯和高斯高/低通滤波器，并用其对实际的图像进行相应处理。同时掌握了对不同滤波器的物理意义作用的理解，了解到了各种滤波器对图像的滤波功能的差异，认识到了它们在实际中的作用。