

第二次试验报告

09014217 刘兴成

1、读片软件的功能设计

该读片软件具有显示、放大缩小 dw 格式图像文件的功能，同时主要还能够实现对图像的灰度调试，可调控灰度窗宽和窗位。

2、选择算法依据

所用到的算法主要是对图像的灰度值映射，算法核心思路是先将原图的 16 位数据映射到 8 位数据以供显示，然后通过外界提供的灰度窗宽和窗位进行二次映射，8 位数据映射到 8 位数据上。

3、灰度映射算法细节

首先是读取读取图像的宽和长数据，图像宽、长存储在 dw 文件的前 4 字节中，为无符号双字节整数。

```
ifstream input("D:\\\\3.dr", std::ios::binary);
unsigned short *width_=new unsigned short ;
unsigned short *height_=new unsigned short ;
input.read((char *)width_, 2);
input.read((char *)height_, 2);
width=*width_;
height=*height_;
length=width*height;
buffer=new unsigned short [length];
input.read((char *)buffer, length*2);
input.close();
```

获取图像之后，先找出原图像各个像素值中的最大值和最小值。

```
maxVal=0x0000;
minVal=0xFFFF;
for(int j=2;j<length;j++)
{
    unsigned int tmp=buffer[j];
    if(tmp>maxVal)maxVal=tmp;
    if(tmp<minVal)minVal=tmp;
}
```

然后将原图像的各个像素的值由[minVal,maxVal]线性映射到[0,255]，

```
for(int i=0;i<int(width/div);i++)
    for(int j=0;j<int(height/div);j++)
    {
        unsigned int num=buffer[int(j*div*width+i*div)];
        num=(num-minVal)*255/delt;
        img1.SetPixel(i, j, RGB(num, num, num));
    }
```

处理之后的结果 img1 作为之后所有处理的原始图像。

接下来通过一个 grayWindowImage 函数来处理所有的需要做灰度图转换的工作。该函数输入参数分别为原图像像素最小值 minVal，原图像像素最大值 maxVal，灰度窗的左边缘 left，灰度窗的右边缘 right，原始图像 src 和转换之后的图像 dst。

```

void CMFCApplication2View::grayWindowImage(int minVal, int maxVal, int left, int right, const CImage &src, CImage &dst)
{
    unsigned int width=src.GetWidth();
    unsigned int height=src.GetHeight();//获取原图像的长宽
    unsigned int delt=0;
    dst=CImage();
    dst.Create(width,height,24);//目标图像创建相同尺寸
    slided=false;
    delt=maxVal-minVal;

    unsigned int windowWidth=right-left;//获得灰度窗宽

    for(int i=0;i<width;i++)
    {
        for(int j=0;j<height;j++)
        {
            unsigned int num=GetRValue(src.GetPixel(i,j));//获得原图像像素
            unsigned int num2;
            if(num<=left)
            {
                num2=0;//在窗体左侧的全部置位0
            }
            else if(num>=right)
            {
                num2=255;//在窗体右侧的全部置位255
            }
            else
            {
                num2=(num-left)*255/windowWidth;//在窗体中的像素进行映射
            }

            dst.SetPixel(i,j,RGB(num2,num2,num2));//设置新图像的像素值
        }
    }
}

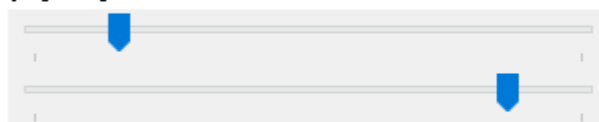
```

此函数运行的主要过程就是对原始图像每个数据进行处理,将在灰度窗之外的像素值全部置为 0 或 255, 在灰度窗之内的值进行线性映射。

软件通过多个滑动条来实现灰度窗的调节。并且可以显示当前灰度窗位置。

Min :39
Max :220
Width :181
Center:129

灰度窗宽调节



灰度窗位调节



4、程序测试结果

对三张图依次进行不同程序的灰度变化测试。

对第一张图的测试：



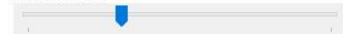


Min :30
Max :126
Width :96
Center:78

灰度窗宽调节



灰度窗位调节



change



Min :85
Max :126
Width :41
Center:105

灰度窗宽调节

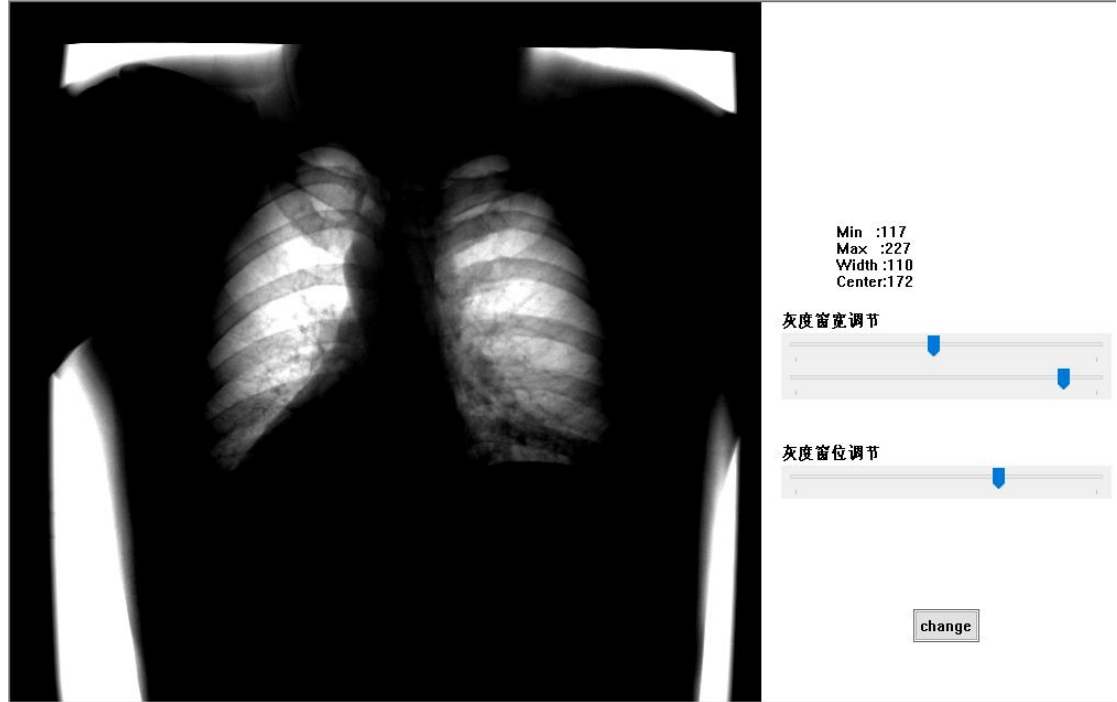
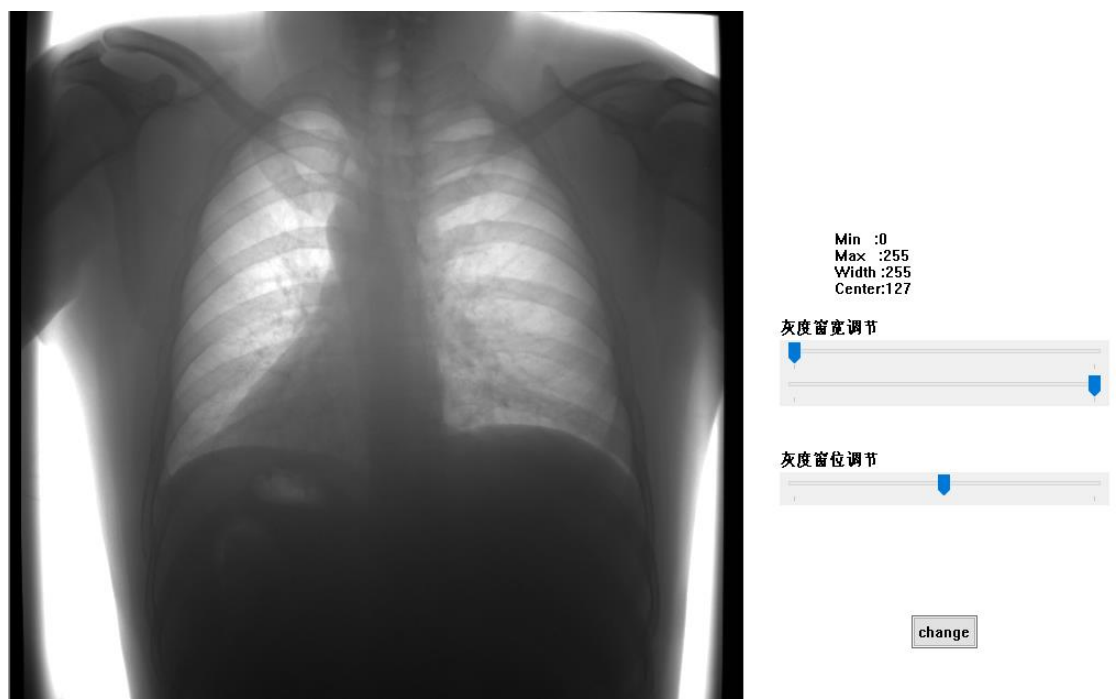


灰度窗位调节




change

对第二张图的测试：



对第三张图的测试：




Min :0
Max :255
Width :255
Center:127

灰度窗宽调节

灰度窗位调节

change



Min :150
Max :255
Width :105
Center:202

灰度窗宽调节

灰度窗位调节

change