**南昌航空大学**

**19学年—20学年第 2 学期 医疗软件技术基础 实验三**

专业名称： 生物医学工程 实验学时： 2

学号： 18084124 姓名： 杨振宇

实验题目：DICOM文件读取

实验环境： vscode

实验目的：

1．掌握DICOM文件的格式；

2．掌握使用C读取DICOM文件中的信息和图像。

3. 掌握OPENCV库的使用

实验内容：

（1）通过open函数读取DICOM文件；

（2）顺序读取各种Tag；

（3）通过TAG，读取相应信息；

（4）通过TAG读取图像信息，并进行调窗和显示。

实验要求：

(1) 详细报告OPENCV的配置。

(2) 程序要添加适当的注释，程序的书写要采用缩进格式。

(3) 对DICOM文件中的重要信息进行输出。

(4) 根据实验报告模板详细书写实验报告,在实验报告中给出算法的流程图。

实验程序及注释：

// 11111.cpp : �������̨Ӧ�ó������ڵ㡣

//

#include <iostream>

#include <opencv2/opencv.hpp>

#include "highgui.h"

/\*#pragma comment(lib, "opencv\_core2411.lib")

#pragma comment(lib, "opencv\_imgproc2411.lib")

#pragma comment(lib, "opencv\_highgui2411.lib")\*/

using namespace cv;

using namespace std;

struct TagValue//tag有前后两个部分

{

    unsigned short tag1;//FF FF        组号

    unsigned short tag2;//FF FF        元素号

};

enum E\_SourcePixelType         //像素的类型

{

    ESourcePixelType\_U16,       // USHORT ，为0

    ESourcePixelType\_I16,       // SHORT  ，为1

};

int main()

{

    bool isVR=true;//值类型(VR): 显式或者隐式

    bool isLitteEndian=true;

    int file\_length=0;//文件长度

    char VR[2];//长为3字节的字符数组，FF FF /0,没必要

    unsigned int pixDataLen=0;//FF FF FF FF

    unsigned int pixDataOffset=0;

    unsigned short channle=0;

    unsigned short rows=0;

    unsigned short cols=0;

    unsigned short dataLen=0;

    unsigned short validLen=0;

    E\_SourcePixelType pixelType;

    int windowsWidth=0;//窗口宽度

    int windowsCenter=0;//窗口中心？

    bool ZeroIsBlack=true;

    float RescaleSlope =0.06;

    float RescaleIntercept=0;

    FILE \*fp;//用来指向图片文件

    fp=fopen("11.dcm","rb");

    if(fp==NULL)

    {

        printf("can not open file!");

          cv::waitKey();

        return 0;

    }

    fseek(fp,0,SEEK\_END);//文件指针定位到文件末尾，偏移0个字节

    file\_length=ftell(fp);//算出长度，fp相对文件首的偏移量，单位字节

    fseek(fp,0,SEEK\_SET);//重新定位到头

    fseek(fp,128,SEEK\_SET);//跳过前128个字节的前导言

    char headchar[5];//读前缀用的

    memset(headchar,0,5);//初始化设置0

    int read\_num = fread(headchar,1,4,fp);//读到前缀，读四次

     //确认是否读到四个字节

    if(read\_num!=4)

    {

        fclose(fp);

        return 0;

    }

    //对字符串进行比较前缀是否是DICM,strcmp是不相等为1（true）

    if(strcmp(headchar,"DICM"))

    {

        fclose(fp);

        printf("File is not DICM");

        return 0;

    }

    //直至数据处理完毕

    while(ftell(fp)+6<file\_length)//为什么是六，不是8

    {

        //局部变量的定义

        //在C99标准中C同C++一样允许在for循环语句中定义变量。并且这个变量作用域被限定在for循环中,在for循环外就成为了未定义变量（C++也是）。

        TagValue tag;

        unsigned int len;//ff ff ff ff

        memset(VR,0,3);//赋0

        fread(&tag,sizeof(TagValue),1,fp);//读一个tag FFFF FFFF

        int index=ftell(fp);//得到偏移量

        if(tag.tag1==0x02)//0x02

        {

            fread(VR,1,2,fp);//读两个字节

            if(!strcmp(VR,"OB")||!strcmp(VR,"OW")||!strcmp(VR,"SQ"))

            {

                fseek(fp,2,SEEK\_CUR);//跳两字节

                fread(&len,sizeof(unsigned int),1,fp);//4个字节，值长

            }

            else

            {

                unsigned short l;

                int ss = fread(&l,sizeof(unsigned short),1,fp);

                int a=ftell(fp);//这里的a都没有定义，虽然a没有用

                len =(unsigned int)l ;

            }

        }

        else if(tag.tag1==0xfffe)

        {

            if(tag.tag2==0xe000||tag.tag2==0xe00d||tag.tag2==0xe0dd)

            {

                fread(&len,sizeof(unsigned int),1,fp);

            }

        }

        else if(isVR==true)//操作和0x02差不了太大

        {

            fread(VR,1,2,fp);

            int a= ftell(fp);//没用到

            if(!strcmp(VR,"OB")||!strcmp(VR,"OW")||!strcmp(VR,"SQ"))

            {

                fseek(fp,2,SEEK\_CUR);

                fread(&len,sizeof(unsigned int),1,fp);

            }

            else//VR为普通类型

            {

                unsigned short l;//FF FF

                l=sizeof(unsigned short);//存，长度2？？？

                fread(&l,sizeof(unsigned short),1,fp);//INT

                a= ftell(fp);

                len =(unsigned int)l ;

            }

        }

        else if(isVR==false)

        {

            fread(&len,sizeof(unsigned int),1,fp);

        }

//以上做长度读取，方便接下来跳过数据值部分，或做数值的处理，读取，各个控制变量的设置

        if(tag.tag1==0x02&&tag.tag2==0x10)//TAG为0002 0010

        {

            char msg[124];//为什么不是125

            memset(msg,0,124);

            fread(msg,1,len,fp);

            if(!strcmp(msg,"1.2.840.10008.1.2.1"))

            {

                isLitteEndian=true;

                isVR=true;

            }

            else if(!strcmp(msg,"1.2.840.10008.1.2.2"))

            {

                isLitteEndian=false;

                isVR=true;

            }

            else if(!strcmp(msg,"1.2.840.10008.1.2"))

            {

                isLitteEndian=true;

                isVR=false;

            }

        }

        else if(tag.tag1 ==0x28 && tag.tag2==0x103)//像素类型

        {

            unsigned short m;

            fread(&m,sizeof(unsigned short),1,fp);//FF FF

            if(m==0)

            {

                pixelType = ESourcePixelType\_U16;

            }

            else if(m ==1)

            {

                pixelType = ESourcePixelType\_I16;//保存m的值

            }

        }

        else if(tag.tag1==0x7fe0&&tag.tag2==0x10)//重要的图片数据位置要保存下来，包括图片的长度，起始偏移量

        {

            pixDataLen=len;

            pixDataOffset=ftell(fp);

            fseek(fp,len,SEEK\_CUR);//跳掉

        }

        //窗口配置

        else if(tag.tag1==0x28&&tag.tag2==0x10)

        {

            fread(&rows,sizeof(unsigned short),1,fp);//行

        }

        else if(tag.tag1==0x28&&tag.tag2==0x11)

        {

            fread(&cols,sizeof(unsigned short),1,fp);//列

        }

        else if(tag.tag1==0x28&&tag.tag2==0x02)

        {

            fread(&channle,sizeof(unsigned short),1,fp);

        }

        else if(tag.tag1==0x28&&tag.tag2==0x101)

        {

            fread(&validLen,sizeof(unsigned short),1,fp);

        }

        else if(tag.tag1==0x28&&tag.tag2==0x100)

        {

            fread(&dataLen,sizeof(unsigned short),1,fp);

        }

        else if(tag.tag1==0x28&&tag.tag2==0x1050)

        {

            char msg[11];

            memset(msg,0,11);

            fread(msg,1,len,fp);

            windowsCenter=atoi(msg);

        }

        else if(tag.tag1==0x28,tag.tag2==0x1051)

        {

            //fseek(fp,len,SEEK\_CUR);

            char msg[40];

            memset(msg,0,40);

            fread(msg,1,len,fp);

            windowsWidth=atoi(msg);

        }

        else if(tag.tag1==0x0028&&tag.tag2==0x0004)

        {

            char msg[40];

            memset(msg,0,40);

            fread(msg,1,len,fp);

            if(!strcmp(msg,"MONOCHROME1 "))

            {

                ZeroIsBlack=false;

            }

            else if(!strcmp(msg,"MONOCHROME2 "))

            {

                ZeroIsBlack=true;

            }

        }

        else if(tag.tag1==0x0028&&tag.tag2==0x1052)

        {

            char msg[40];

            memset(msg,0,40);

            fread(msg,1,len,fp);

            RescaleIntercept=atof(msg);

        }

        else if(tag.tag1==0x0028&&tag.tag2==0x1053)

        {

            char msg[40];

            memset(msg,0,40);

            fread(msg,1,len,fp);

            RescaleSlope =atof(msg);

        }

        else

        {

            char msg[1024];

            memset(msg,0,1024);

            fread(msg,1,len,fp);

        }

    }

    //读完了

    fseek(fp,pixDataOffset,SEEK\_SET);

    if(windowsCenter==0&&windowsWidth==0)

    {

        windowsWidth = 1 << validLen;

        windowsCenter = windowsWidth / 2;

    }

    int min\_value,max\_value;

    min\_value=windowsCenter-windowsWidth/2.0+0.5;

    max\_value=windowsCenter+windowsWidth/2.0+0.5;

    double pers = 255.0/(max\_value-min\_value);//255?

    Mat src;//

    Mat src2;//

    int nPixel= 0;

    double fCtA = 0;

    double fCtB = 0;

    fCtA = (double)256 /windowsWidth;//256???PERS

    fCtB = 128 - 256 \* (double)windowsCenter / windowsWidth;//??

    if (fCtB < 0)

    {

        fCtB = 0;

    }

    if (fCtB > 255)

    {

        fCtB = 255;

    }

    if(channle==1)

    {

        src.create((int)rows,(int)cols,CV\_8UC1);//8位

        src2.create((int)rows,(int)cols, CV\_16SC1);//16位

        for (int i = 0;i<rows;i++)

        {

            for(int j=0;j<cols;j++)

            {

                unsigned short   gray=0;

                short gray2 = 0;

                unsigned char pix[2];

                fread(pix,1,2,fp);

                if(pixelType == ESourcePixelType\_U16)

                {

                    if(validLen<=8)

                    {

                        if(isLitteEndian)

                        {

                            gray=pix[0];//BIG

                        }

                        else

                        {

                            gray=pix[1];//LITTLE

                        }

                    }

                    else

                    {

                        long temp = 0;

                        if(isLitteEndian)

                        {

                            gray=\*(unsigned short\*)pix;//FF FF

                            if(gray > 32767)

                            {

                                gray = 32767;

                            }

                            temp =gray\*RescaleSlope+RescaleIntercept;

                            temp = temp \* fCtA + fCtB;

                        }

                        else

                        {

                            gray=pix[1]+pix[0]\*256;//0为高位

                            temp = gray \* RescaleSlope + RescaleIntercept;

                            temp = temp \* fCtA + fCtB;

                        }

                        int nValue = (int)temp;

                        if(nValue>0xff)

                        {

                            nValue=0xff;

                        }

                        else if(nValue<0)

                        {

                            nValue=0;

                        }

                        nPixel = nValue;

                    }

                }

                else if(pixelType ==ESourcePixelType\_I16)

                {

                    if(validLen<=8)

                    {

                        if(isLitteEndian)

                        {

                            gray2=pix[0];

                        }

                        else

                        {

                            gray2=pix[1];

                        }

                    }

                    else

                    {

                        long temp = 0;

                        if(isLitteEndian)

                        {

                            gray2=\*(short\*)pix;

                            if(gray2 > 32767)

                            {

                                gray2 = 32767;

                            }

                            if(gray2 < -32767)

                            {

                                gray2 = -32767;

                            }

                            temp =gray2\*RescaleSlope+RescaleIntercept;

                            temp = temp \* fCtA + fCtB;

                        }

                        else

                        {

                            gray2=pix[1]+pix[0]\*256;

                            temp = gray2 \* RescaleSlope + RescaleIntercept;

                            temp = temp \* fCtA + fCtB;

                        }

                        int nValue = (int)temp;

                        if(nValue>0xff)

                        {

                            nValue=0xff;

                        }

                        else if(nValue<0)

                        {

                            nValue=0;

                        }

                        nPixel = nValue;

                    }

                }

                if(!ZeroIsBlack)

                {

                    nPixel=255-nPixel;

                }

                src.at<uchar>(i,j)=nPixel;

            }

            cout<<std::endl;

        }

    }

    else if(channle==3)

    {

        src.create((int)rows,(int)cols,CV\_8UC3);

        for (int i = 0;i<rows;i++)

        {

            for(int j=0;j<cols;j++)

            {

                unsigned char pix[3];

                fread(pix,1,3,fp);

                src.at<Vec3b>(i,j)[0]=pix[2];

                src.at<Vec3b>(i,j)[1]=pix[1];

                src.at<Vec3b>(i,j)[2]=pix[0];

            }

        }

    }

    Mat src1;

    //src1.create(rows,cols,CV\_8UC3);

    //imwrite("D:\\src.bmp", src);

    //src1=imread("d:\\s1.png");

    fclose(fp);

    cvNamedWindow("Dicomimage",0);

    cv::imshow("Dicomimage",src);

    //const CvArr\* img=(CvArr\*)&src;

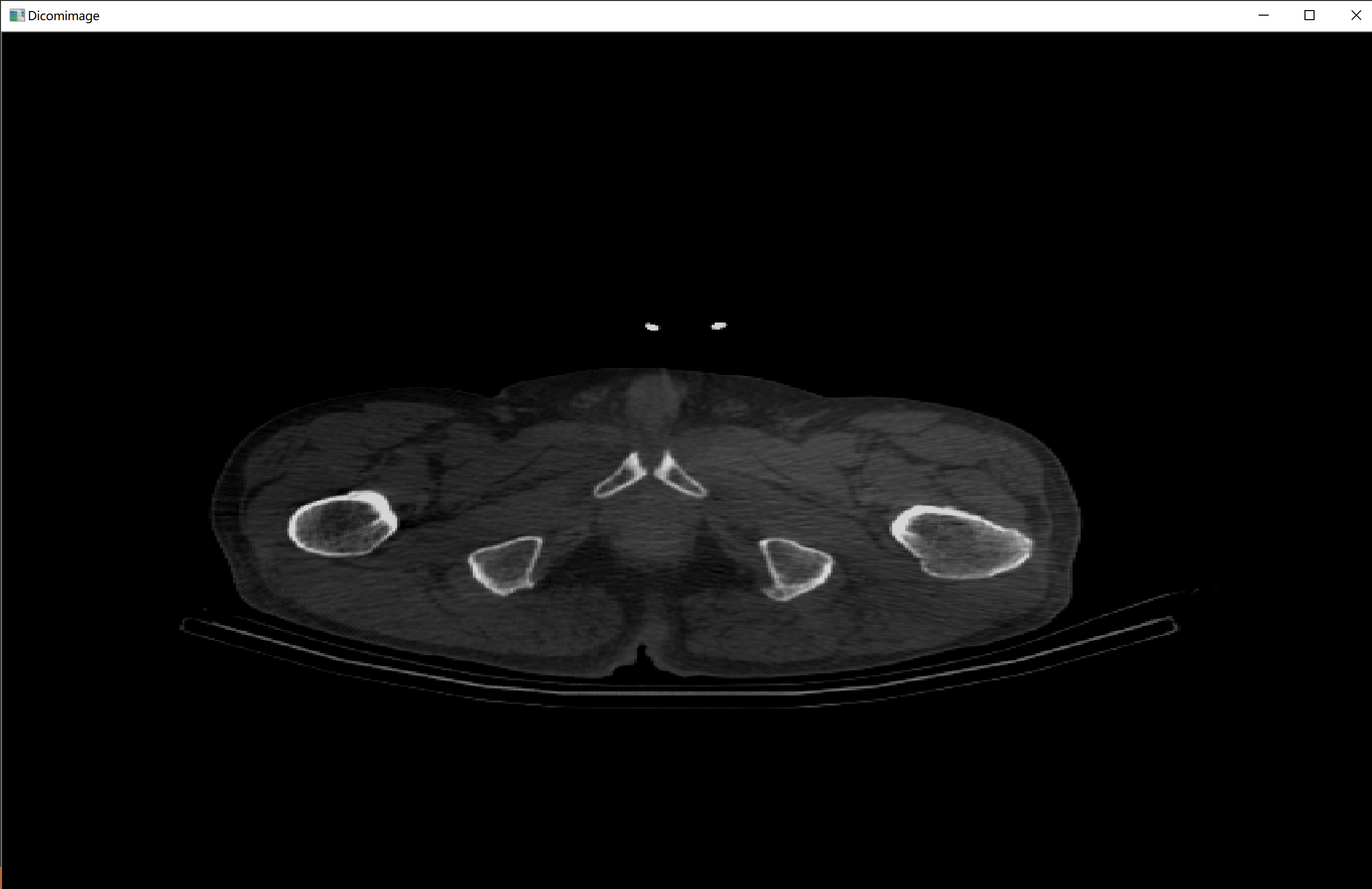
    //cvShowImage("Dicomimage",img);

    cvWaitKey(0);

    return 0;

}

实验结果：



流程图：CVV

.dcm存在

N

前缀为dicm

结束

数据未处理完

得长

读值

处理参数

定位图片数据位置

根据参数生成图片

结束

**Y**

 N

结束

 Y

 N

 Y

Vscode下的配置方法

{

    "configurations": [

        {

            "name": "win",

            "includePath": [

                "${workspaceFolder}/\*\*",

               "C:/WIN10/Program Files/opencv/build/include/" ,

               "C:/WIN10/Program Files/opencv/build/include/opencv" ,

               "C:/WIN10/Program Files/opencv/build/include/opencv2"

            ],

            "defines": [],

            "compilerPath": "C:/mingw64/mingw64/bin/gcc.exe",

            "cStandard": "c11",

            "cppStandard": "c++17",

            "intelliSenseMode": "clang-x64"

        }

    ],

    "version": 4

}

{

    // 使用 IntelliSense 了解相关属性。

    // 悬停以查看现有属性的描述。

    // 欲了解更多信息，请访问: https://go.microsoft.com/fwlink/?linkid=830387

    "version": "0.2.0",

    "configurations": [

        {

            "name": "g++.exe - 生成和调试活动文件",

            "type": "cppdbg",

            "request": "launch",

            "program": "${fileDirname}\\${fileBasenameNoExtension}.exe",

            "args": [],

            "stopAtEntry": false,

            "cwd": "${workspaceFolder}",

            "environment": [],

            "externalConsole": false,

            "MIMode": "gdb",

            "miDebuggerPath": "C:\\mingw64\\mingw64\\bin\\gdb.exe",

            "setupCommands": [

                {

                    "description": "为 gdb 启用整齐打印",

                    "text": "-enable-pretty-printing",

                    "ignoreFailures": true

                }

            ],

            "preLaunchTask": "C/C++: g++.exe 生成活动文件"

        }

    ]

}

{

    "files.associations": {

        "cstring": "cpp"

    }

}

{

    "tasks": [

        {

            "type": "cppbuild",

            "label": "C/C++: g++.exe 生成活动文件",

            "command": "C:\\mingw64\\mingw64\\bin\\g++.exe",

            "args": [

                "-g",

                "${file}",

                "-o",

                "${fileDirname}\\${fileBasenameNoExtension}.exe",

        "-I", "C:/WIN10/Program Files/opencv/build/include" ,

        "-I", "C:/WIN10/Program Files/opencv/build/include/opencv" ,

        "-I", "C:/WIN10/Program Files/opencv/build/include/opencv2" ,

        "-L", "C:/WIN10/Program Files/opencv/build/x64/mingw/lib",

        "-l", "opencv\_core349",

        "-l", "opencv\_imgproc349",

        "-l", "opencv\_imgcodecs349",

        "-l", "opencv\_video349",

        "-l", "opencv\_ml349",

        "-l", "opencv\_highgui349",

        "-l", "opencv\_objdetect349",

        "-l", "opencv\_flann349",

        "-l", "opencv\_imgcodecs349",

        "-l", "opencv\_photo349",

        "-l", "opencv\_videoio349"

            ],

            "options": {

                "cwd": "${workspaceFolder}"

            },

            "problemMatcher": [

                "$gcc"

            ],

            "group": {

                "kind": "build",

                "isDefault": true

            },

            "detail": "调试器生成的任务。"

        }

    ],

    "version": "2.0.0"

}

配置文件结束

实验小结：

本次实验我初步了解的dicom协议的使用运用opencv库显示dicom协议下的文件的读取，详细了解opencv库的配置，掌握了一些计算机库的配置方法，我学到了很多