

24780 Engineering Computation: Problem Set 9

(*) In the following instructions (and in all course materials), substitute your Andrew ID wherever you see *yourAndrewId*.

You need to create a ZIP file (which may appear as a compressed folder in Windows) and submit the ZIP file via the 24-780 Canvas. The filename of the ZIP file must be:

`PS09-YourAndrewID.zip`

For example, if your Andrew account is hummingbird@andrew.cmu.edu, the filename must be:

`PS09-hummingbird.zip`

Failure to comply with this naming rule will result in an automatic 5% deduction from this assignment's credit. If we cannot identify the submitter of the file, an additional 5% credit will be lost. If we are ultimately unable to connect you with the submitted ZIP file, you will receive 0 points for this assignment. Therefore, ensure strict adherence to this naming rule before submitting a file.

The ZIP file must be submitted to the 24-780 Canvas. If you find a mistake in a previous submission, you can re-submit the ZIP file with no penalty as long as it's before the submission deadline.

Your Zip file should contain:

- `ps9.cpp`
- `noerror.png` or `noerror.jpg`. A screenshot from the compiler server showing there is no error either in `.png` or `.jpg` format.

Do not include project files and intermediate files generated by the compiler. But, do not worry about some files or directories that are automatically added by the archiver (`__MACOSX__` file for example).

Notice: The grade will be assigned to the final submission only. In the case of multiple file submissions, earlier versions will be discarded. Therefore, when resubmitting a ZIP file, it **MUST** include all the required files. Also, if your final version is submitted after the submission deadline, the late-submission policy will be applied, regardless of how early your earlier version was submitted.

Ensure that your program can be compiled without errors on one of the compiler servers. Do not wait until the last minute, as the compiler servers may become very busy just minutes before the submission deadline!

Submission Due: Please refer to Canvas.

START EARLY!

Unless you are a good programmer, there is no way to finish the assignment overnight.

Binarization [ps9.cpp] (50 pts)

(*) We ask you to include a screenshot from the compiler server showing no error in the Zip file you submit.

Download ps9.png from Canvas, which is one slice of CT scan of a human skull.

Write a program that reads ps9.png, which is stored in the program's resource directory and render it. Set up your project so that ps9.png is copied to the resource directory, and use `FsChangeTo-ProgramDir()` before loading ps9.png.

In the main function, first load ps9.png, and then open a window that is the same size png image. When the user presses the space key, binarize the image, or make it black and white.

To do so, you need to visit each pixel, and if one of the components of the pixel is less than 220, make all components zero, otherwise make all components 255.

Fig. 1 shows the input image, and if you do it correctly, your graphics window should look like the image shown in Fig. 2.

Contour Extraction [ps9.cpp] (50 pts)

We often need to extract contour from a binarized image for many applications. Add a functionality to your program so that the contour is extracted when the user presses the ENTER key after the image is binarized.

You do not have to consider the situation if the user pressed the ENTER key before pressing the SPACE key, as long as your program does not crash.

To extract contour, you can visit each pixel, and change the white pixel to black if all the neighbors are white.

There are many ways to do it. Probably the easiest way to do it is (1) first visit all (x,y) coordinates and check if the pixel is white and neighboring at least one black pixel in the Red component, and if it is not clear Green component of the pixel, and (2) visit each pixel and copy Green component of the pixel to Red and Blue components.

To make it easy, you can assume no contour is on the edge of the image (you won't have to consider pixels on the edge of the image).

If you do it correctly, you will see an image like Fig. ??.



Fig. 1: Input

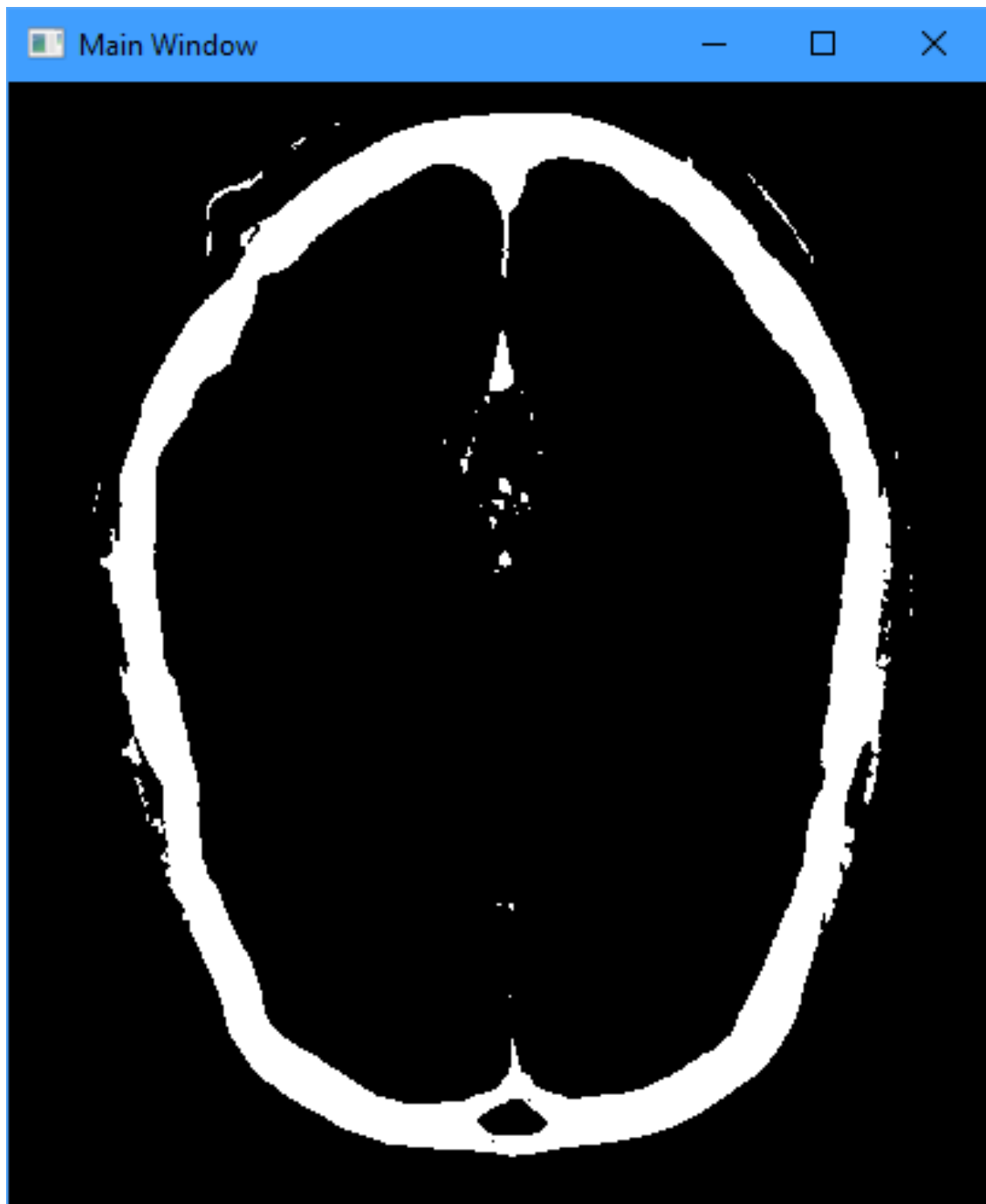


Fig. 2: Binarized image

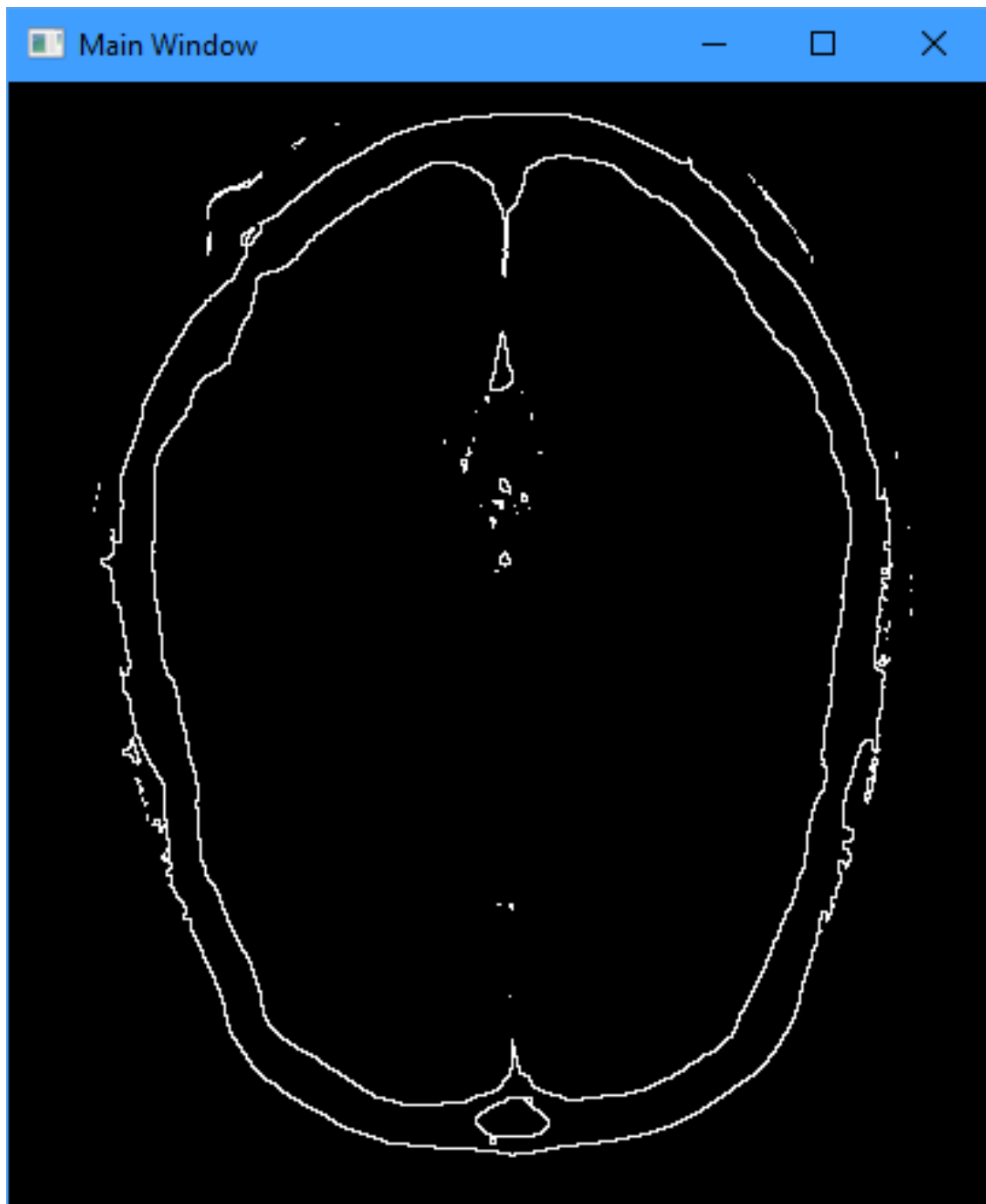


Fig. 3: Extracted contour

Test Your Program with One of the Compiler Servers

Test your program with one of the following compiler servers:

```
http://freefood1.lan.local.cmu.edu  
http://freefood2.lan.local.cmu.edu  
http://freefood3.lan.local.cmu.edu  
http://freefood4.lan.local.cmu.edu
```

You need to make sure you are not getting any errors (red lines) from the compiler server.

It is a good practice to remove warnings as well. However, we will not take points off for warnings as long as your program satisfies requirements of the assignment.

You can only access these servers from CMU network. If you need to access from your home, use CMU VPN. Please visit the CMU computing services web site how to install the VPN.