CISC/CMPE 472 – Assignment 2 (25%)

Visualization and Segmentation

Due: February 17th, 2023

For this assignment you are expected to implement all functions from scratch. Each question describes the structure of the function your code should implement. You may implement as many additional helper functions as required to keep your code legible. The use of image processing libraries such as OpenCV, SimpleITK, matplotlib, etc are only permitted where explicitly stated in the question (**numpy is always permitted**). Any code from previous assignments may be updated to use image processing libraries. Any libraries that you choose to use must be easily installed into an anaconda environment using either pip install or conda install. You should include in your pdf a list of all libraries that required installation. Your code must be compatible with Python 3 (preferably a recent version). Code that does not run will not be graded. Your submission should be uploaded to OnQ as a zip file containing one python file containing your code named **CISC472_A2.py** and one PDF file containing your written responses. Also include in your zip folder any files generated in specific questions. Ensure that your name and student number are in the header of your python and pdf files.

1) **Tumor simulation** *[5 marks]*

   Using your image simulation function from assignment 1, create a 3D volume of shape (224, 224, 224) by generating a stack of slices each containing an ellipse. Your function should return a greyscale volume as a numpy array of shape (224,224,224). The width and height of the ellipse can vary randomly from slice to slice by no more than 5% in each dimension. The center point may also be moved randomly, but by no more than 5 pixels at a time. This variation will allow you to create a volume with a non-uniform shape.

   Your simulated image function should be modified such that it can accept the width and height of the ellipse, along with the center coordinates. For this question, you may assume that the size of each slice is fixed at (224,224).

   Use your smoothing filter from assignment 1 repeatedly (10-100 times) on each slice to distort the edges of the ellipses. You may update your filtering functions to use image processing libraries.

   Save a copy of your simulated volume as **simulatedTumor.npy** for use in question 2. Include this file in your submission along with your python and pdf files.

2) **Volume Reslicing** *[5 marks]*

   a) Write a function that will reslice your volume to give axial, sagittal and coronal image slices. This task involves rotating the volume axes in order to alter which 2 axes make up the primary image plane. An example is shown in Figure 1. Test your function with your simulated volume generated from question 1. Display the center slice in the volume in each plane, include these images in your PDF.

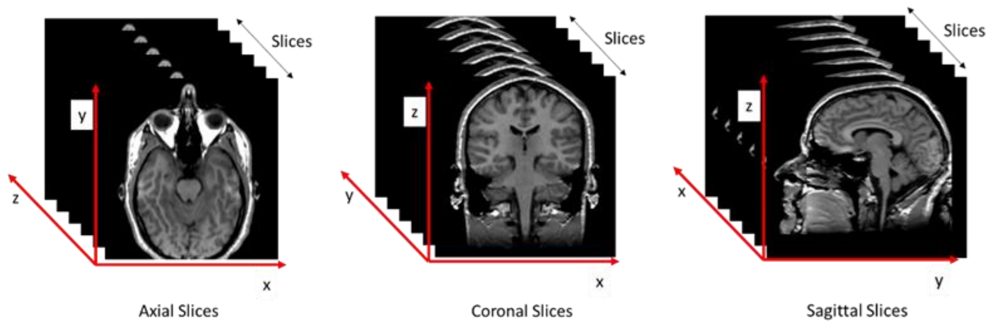

Figure 1. Volume reslicing

b) To ensure that your function is working correctly, compare the image slices your function generates to those generated by 3D Slicer as shown in Figure 2. To load your volume into slicer, open the python interactor (1) and run the following lines of code (2):

```
>>> import numpy
>>> tumourArray = numpy.load("<Replace with your file path>/simulatedTumor.npy")
>>> volume = slicer.util.addVolumeFromArray(tumourArray)
```

Your volume should appear in the Red, Green and Yellow Slice views. You may need to center the views on your volume by clicking the recenter button (3). Scroll through the slices to find approximately the center slice in each view using the scroll widget above each slice (4). To display which plane each slice viewer is currently showing, click on the push pin icon (5). Visualize each plane in the 3D viewer by clicking on the eye in the dropdown menu (6). Take a screen shot of your resulting slicer window and include it in your PDF.

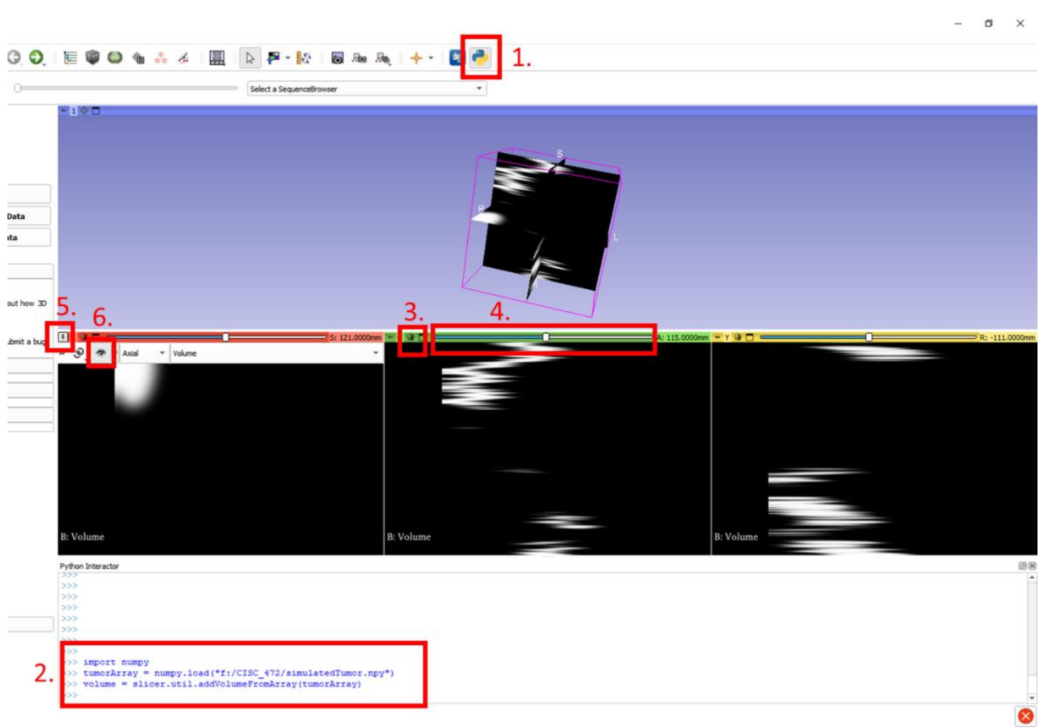

Figure 2. Sample slicer scene

c) Navigate to the Volume rendering module in 3D Slicer (1). Click the eye (2) beside the volume name on the module panel to display your volume in the 3D viewer. Under the advanced tab, modify the scalar opacity mapping to match that shown in Figure 3 (3). Include a screenshot of your whole slicer screen in your PDF.

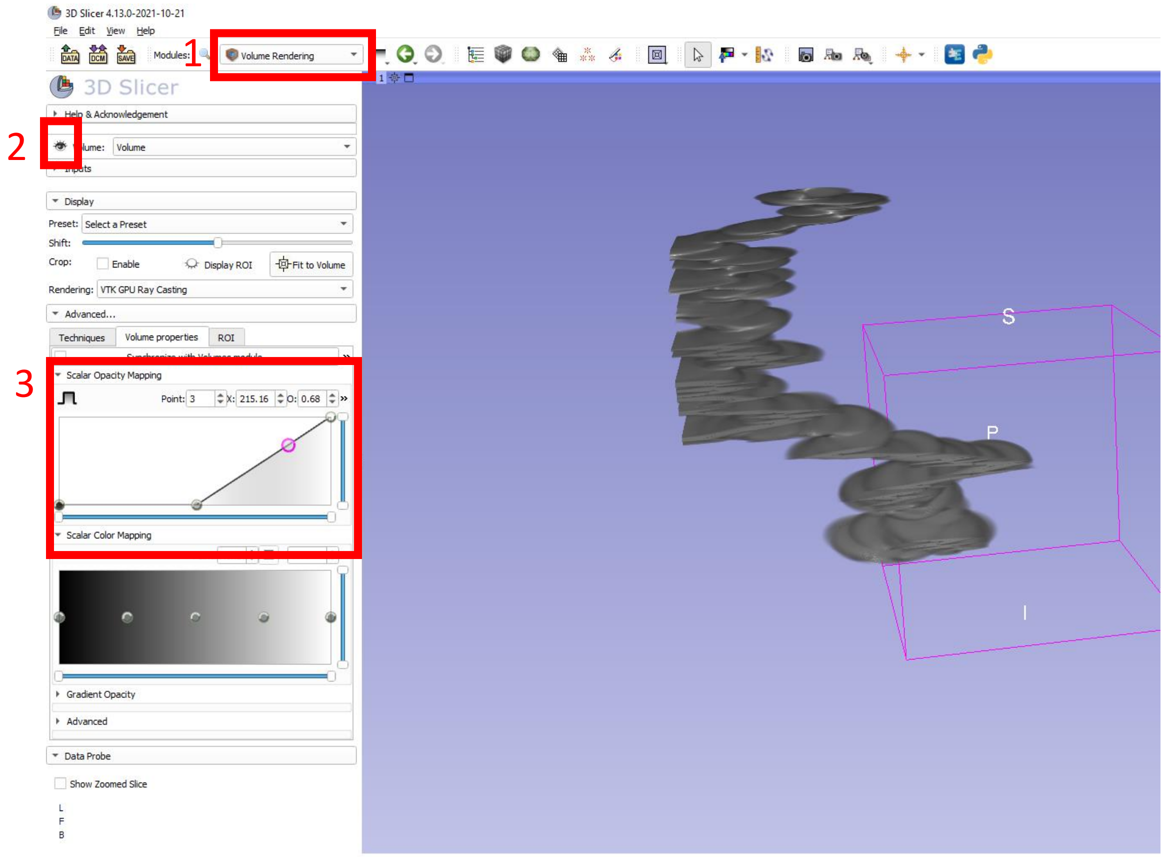

Figure 3. Volume rendering module

3) **Rendering** *[5 marks]*

a) Implement 2 functions that will perform the create a maximum intensity projection image and a digitally rendered radiograph, using the methods described in lecture. Your projections may use parallel rays, you do not need to implement perspective ray tracing (though there might be bonus marks if you do). Your function may also take the desired viewing plane as a parameter (axial, sagittal or coronal) but it is not required. You do not need to make this work for oblique planes, only the 3 standard viewing planes.

b) Test your functions using your simulated volume. Include the created projections for each plane (and each method) in your pdf.

4) **Point-based segmentation** *[5 marks]*

a) Implement a simple thresholding algorithm that will take a greyscale image/volume (may be 2D or 3D), along with an upper and lower intensity threshold and return a binary segmentation. Your segmentation must be the same size as the original image/volume. In your segmentation, pixels/voxels that fall within the 2 thresholds should be assigned the value 1, and all others outside the range should be assigned the value 0.

b) Test your function using your simulated volume (3D). Display the center slice in each plane and include these in your pdf.

5) **Region-based segmentation** *[10 marks]*

a) Implement a simplified version of the region growing algorithm that was discussed in class. Your function should take in: a greyscale image (2D only), a list of the current seeds (1 seed for object, 1 seed for background), and the maximum difference threshold. You may add additional parameters as needed, so long as they have default arguments. You may assume that your function will only be splitting the image into 2 regions, 1 region that is the object of interest, 1 region for the background. The returned segmentation should be binary and the same dimensions as the original image. Pixels within the object should be assigned the value 1 and pixels belonging to the background should be assigned the value 0.

b) Test your function using a slice of your simulated volume. Have the object seed begin at a location with the highest intensity, and the background seed at a location with the lowest intensity. Set your maximum difference threshold to 0. Include the resulting segmentation in your pdf. Experiment with varying the threshold, what do you notice about the resulting segmentation? Include any resulting images in your pdf as well.

6) **Brain tumor segmentation** *[25 marks]*

For the following question use the volume provided with the assignment called "MRBrainTumor.npy". This volume consists of an MRI scan performed on a patient with a very large brain tumor. For part c) and d) your mark is not dependent on how successful your segmentation is, but rather you will be marked on critical thinking and your application of course concepts.

Hint: to avoid distorted images when viewing different image planes, be sure to correct your volume so that it has isotropic spacing (you may use image processing libraries for this. See: https://simpleitk.readthedocs.io/en/v1.2.4/Documentation/docs/source/fundamentalConcepts.html). The original image spacing is (0.9375000000000001, 0.9375000000000001, 1.4000000000000001).

a) Using your function from question 2, reslice the given volume along each of the main imaging planes (axial, sagittal and coronal). Find a slice in each view that has a large cross section of the tumor in view. Note the location of each of these slices. Include each of the three images in your pdf along with their corresponding slice number.

b) Using your solutions from question 3, render the maximum intensity projection and digital radiograph of the volume for each imaging plane. Include the resulting images in your pdf.

Describe the differences you see between the 2 projection methods. What information does each projection method allow you to see?

c) Select one of your slices that from part a) (only 1, does not need to be all 3). Using your segmentation methods from questions 4 and 5, attempt to segment only the tumor from the image. In your pdf include all resulting images along with a description of your process (e.g. how did you decide on your threshold values, how did you select a seed location). What worked well and what didn't? What challenges did you face and how did you attempt to solve them?

d) Using your image filters from assignment 1, attempt to improve your segmentation by preprocessing your images. You may update your filtering functions to use image processing libraries. Document your process. Which filters did you use? What order did you use them in? What worked well and what didn't? Include any images that were generated in your pdf. Were you able to improve on your segmentation? Explain your reasoning for why it did or did not improve.

Mark breakdown

The mark breakdown for this assignment is shown in the table below. Please note that in addition to marks associated with each question, there are 11 marks dedicated for proper coding style. Code that is illegible or difficult to interpret or does not meet the style guidelines outlined below will be docked marks.

Question	Marks associated
1.	5
2. a)	3
b)	2
3. a)	4
b)	1
4. a)	4
b)	1
5. a)	7
b)	3
6. a)	3
b)	2
c)	10
d)	10
Coding style	10
Total	**65**

Style guidelines

Code for this assignment must be implemented in a single python file named **CISC472_A1.py.** Your code must contain an initial header that includes your name and student number. All import statements should be properly organized at the top of the file. All code must be properly structured within functions and all functions should be documented with initial comments describing the behavior of the function as well as the parameters and returns. The use of inline comments should be minimized and used only where necessary. Function and variables should have meaningful names that indicate their purpose (e.g. no single letter variable names). As mentioned earlier you must implement the functions as described, though you may use as many helper functions as you need to ensure that your code is clear and legible. For this assignment the use of image processing libraries such as OpenCV, SimpleITK, matplotlib, etc are only permitted where explicitly stated in the question (numpy is permitted for basic matrix operations). Any libraries that you use must be easily installed in an anaconda environment using a simple pip install or conda install command. Document which libraries you used in your pdf.

Reminder to make sure that your code runs using Python 3. If a function is causing errors that you are unable to fix before the deadline, please comment out that section of code. You may describe what you were trying to do in your PDF file to receive partial marks. Code that does not run will not be graded.