Assignment 1 - CS661
Isocontour and Volume Visualization

Due date: Feb 18, 2024, 11:59pm

Grade: 100 points (10% of the course grade)

1. 2D Isocontour Extraction: [60 Points]

In this part of the assignment, you will write a program to extract isocontour from a 2D uniform grid. The dataset is given in VTKImageData format, which you already know how to load and work with. Your program should generate the isocontour as a VTKPolyData object and write it out to disk as a VTKPolyData file (*.vtp file). This file can be easily loaded in ParaView for visualizing the extracted contour (You may have to change the color of the contour to something other than white if you have set white background in ParaView). This assignment is a simplified version of the Marching Squares algorithm that we went over in class. **You do not have to implement the traditional marching squares algorithm. If you do so and your code shows that you are handling marching square cases unnecessarily, points will be deducted.**

Notes:

- For this part of the assignment, you must write the contour extraction algorithm on your own from scratch and you are not allowed to use VTK's isocontour filter to do this. You can use VTK to read and write the data in VTK's polydata file format.
- You do not have to handle the cells that have ambiguities, and you do not have to implement the Asymptotic Decider to resolve ambiguities. Instead, please traverse the vertices of each cell in the counterclockwise order while finding the isocontour segments. You may maintain a global vtkCellArray object and insert each contour segment into it and then add it to a VTKPolyData.
- Your program should work for any isovalue provided by the user. Hence, submit a Python script (not a Jupyter Notebook), and your script should take the isovalue as an input parameter and write the extracted contour out as a *.vtp (VTKPolyData) file. Make sure your output file can be read in ParaView. The possible range of isovalues for the given data set is between (-1438, 630)
- Finally, submit a README.txt file that describes how your script should be run and how the parameter should be specified. Note that the TAs will follow instructions from your README.txt file to run your code while grading.

Dataset for this task:

The dataset that you will use in this assignment is a 2D slice taken from a 3D Scalar field of a Hurricane Simulation Data. The variable is Pressure. If you want to know more about the original data, please refer to this link: http://vis.computer.org/vis2004contest/index.html

2. VTK Volume Rendering and Transfer Function: [40 Points]

In this second half of the assignment, you will write a Python script to implement the volume rendering algorithm from the VTK library. VTK has already implemented the ray-casting algorithm that we have studied in our class. In this assignment, you will use the vtkSmartVolumeMapper() to render 3D scalar data and set a specific color and opacity transfer function. You will also use VTK's Phong Shading feature to produce advanced lighting effects to make your volume rendering more realistic. The ambient, diffuse, and specular parameters for Phong model are given below. Here are the steps that you should follow for this task:

- Load the 3D data provided with the assignment.

- Create instances of vtkColorTransferFunction and vtkPiecewiseFunction (this will work as Opacity transfer function) and set them up with the values provided below in the tables.
- Use vtkSmartVolumeMapper() class to perform the volume rendering
- Use vtkOutlineFilter to add an outline to the volume rendered data
- By default, advanced shading feature, i.e., Phong shading will be off. Create an input parameter and take input from user if the user wants to use Phong shading. If yes, then your program should turn on Phong shading while rendering.
- Create a 1000x1000 sized render window to show the rendering result.

Other relevant VTK classes you may need: vtkVolume, vtkVolumeProperty.

Color Transfer Function Specification:

Data Value	Red	Green	Blue
-4931.54	0	1	1
-2508.95	0	0	1
-1873.9	0	0	0.5
-1027.16	1	0	0
-298.031	1	0.4	0
2594.97	1	1	0

Opacity Transfer Function Specification:

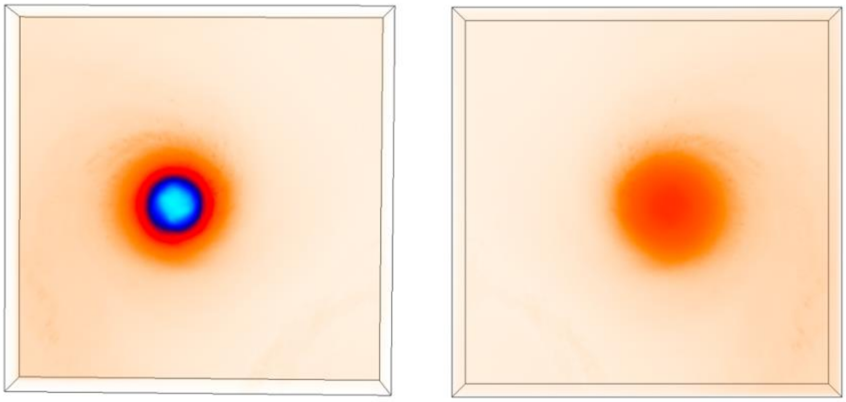
Data Value	Opacity Value
-4931.54	1.0
101.815	0.002
2594.97	0.0

Phong Shading Parameters that you should use:

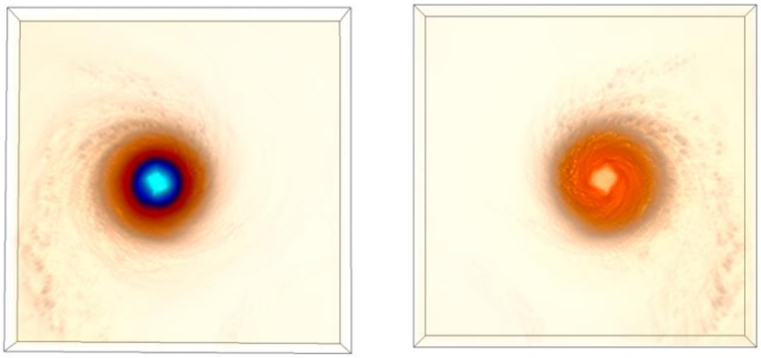
Ambient coefficient: 0.5
Diffuse coefficient: 0.5
Specular coefficient: 0.5

Again, you should write a Python script (not a Notebook), and your script will take the input parameter as to whether the user wants to use Phong shading or not. Update the above README.txt file to add instructions about how to run your volume rendering script.

If you have done everything as suggested, you should see images like the following for volume rendering:



Without Phong shading, front, and back view



With Phong Shading, front, and back view

Dataset for this task:

The dataset that you will use in this assignment is a 3D Scalar field Volume Data of a Hurricane Simulation. Loading this data in VTK will be same as loading the 2D data. The variable in the Data is Pressure. If you want to know more about the original data, please refer to this link: http://vis.computer.org/vis2004contest/index.html.

How to submit?

The HelloIITK portal will be set up for submission. There will be a time limit set. To get 100% credit, you need to submit within time limit. If you miss the deadline, you can still submit, but points will be deducted based on late submission policy. Please start early and finish it by the deadline.

Only one submission per group is expected. You should submit your codes as Python scripts with README file and any other relevant files into a single compressed (*.zip) file. Your code should be commented properly, and when printing any information as part of any assigned task, please mention the information that is being printed. Compress your solution files and name your submission as **"groupnum_rollnum1_rollnum2_Assignment1.zip".**