DataVis Assignment 5

Quick Facts

- 2 tasks: Volume Rendering in ParaView, C++ app with VTK
- You should solve the tasks in groups of three
- Fill out **all** of the following pages
- **Upload all submission files to OPAL!**

Submission files for OPAL:

☐ Filled out submission form as PDF

☐ 2 screenshots of direct volume visualization for 5.1.1 as JPG, PNG or PDF

☐ 1 screenshot of isosurface visualization for 5.1.2 as JPG, PNG or PDF

☐ Cpp file for 5.2

☐ scan or photo of page 4 (task 5.3)

Team Members

Team Member 1: Last Name	Team Member 1: First Name	Team Member 1: Mat. No.

Team Member 2: Last Name	Team Member 2: First Name	Team Member 2: Mat. No.

Team Member 3: Last Name	Team Member 3: First Name	Team Member 3: Mat. No.

Introduction

Please copy the source files into the existing folder structure so that “assignment5” and all its subfolders are located next to “assignment4”. You should re-use the folders “data” and “vtk” from the previous assignment. We provide a new data set called “headsq-half.vti”, a CT image of a person’s head.

Important note: Please upload **only** the source code files. No build files, no libs, no additional files. Please do not use any system-specific routines. Everything should compile under Visual Studio out of the box.

Task 5.1: Volume Visualization with Paraview

Description: Get in touch with ray-casting and isosurfaces.

Import the given dataset `/data/headsq-half.vti` and change the representation from *outline* to *volume*. You can use the *smart* volume rendering mode. Make use of the color transfer function editor to get a feeling for this kind of visualization and try different color and opacity settings. Then, find transfer settings that emphasize distinct parts of the CT data (while keeping the surrounding tissue as contextual information).

5.1.1 Find 2 different transfer function settings that show such interesting parts and make a screenshot for each (including the chosen transfer function). Shortly describe your intended visualization for each image.

5.1.2 What is the density value for bones? Use the contour filter and try to find a good value in order to produce a clean mesh. Make a screenshot.

Task 5.2: Scientific Visualization with C++ and VTK

Description: Assignment5.cpp has lots of useful hints in the code on how to use the classes and how to put everything together. Furthermore there are some helper methods implemented in `vtkhelper` that you can use.

Now the task is to combine two different approaches to volume rendering,

a) direct volume visualization and

b) contour rendering

into one single output image.

Furthermore provide a slider in order to change the iso level interactively. For that you will need your own callback as a derived class of `vtkCommand` which is already given in the code.

Steps:

- Given the data reader use a `vtkSmartVolumeMapper` to let render the volume data directly
    - Turn on GPU-based rendering for high performance
    - set blending mode to **composite**

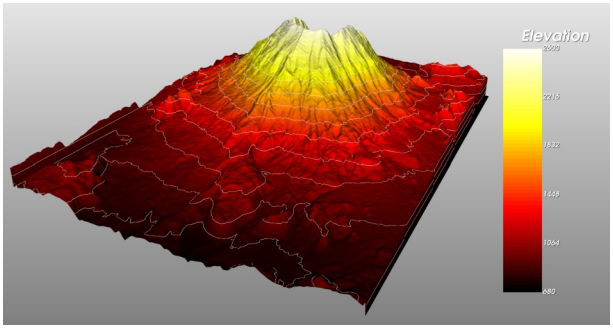
    - assign a meaningful opacity transfer function via `vtPiecewiseFunction` and add at least 2 density-opacity pairs
    - assign a meaningful color transfer function via `vtkColorTransferFunction` and add at least 3 density-color pairs
    - instead of `vtkActor (cp. Assignment4)` use `vtkVolume`
    - utilize `vtkVolumeProperty` for even more properties and apply it to the volume via `SetProperty`
    - create a renderer and a render window, test your code with `doRenderingAndInteraction(window)` method from `vtkhelper.h`
    - if everything went fine you can delete the line `doRenderingAndInteraction(window);`

- Put marching cubes on top
    - use `vtkMarchingCubes` filter
    - set number of contours to one and use a meaningful value (e.g. from task 5.1.2)
    - create mapper (`vtkDataSetMapper`)
    - create actor and set mapper
    - add actor to renderer
    - test your intermediate result with `doRenderingAndInteraction(window)` like you did before

- create an interactive iso value slider
    - create a slider 2D representation, add a title and set min/max values
    - create a corresponding slider widget and interactor
    - put everything together in the window
    - use the given callback class to make the slider observable, understand the callback code and the observer pattern
    - start your combined model with `doRenderingAndInteraction(interactor)` from `vtkhelper.h`

- for better performance test your app in release mode with debugging off (standard hot key is Strg+F5 in Visual Studio)
- upload the cpp only!

- additional hints:
    - Use comments and document your code!
    - Code quality matters!
    - It's okay to build upon existing examples, but: name your sources!



Task 5.3: Compute Marching Squares by hand

Goal: Given a cell in a uniform grid with cell size *h*, see the following figure, your task is to calculate and draw the corresponding lines for the iso-value $v = -1$.

- Mark the locations where the isolines should intersect the edges. Remember that the values $v_{i,j}$ are interpolated linearly. Draw as exact as possible.

The next step is to connect the intersection points on the edges to create the isoline segments for this cell. You will discover an ambiguity that should be resolved explicitly by calculating the asymptotes.

- Write down the general equation for the isoline and solve it for x and y respectively. You will find a singularity in each formula at the x/y-position exactly where the asymptote that is orthogonal to the x/y-axis intersects the cell. Draw the asymptotes into the given picture.

Resolve the ambiguity with the help of the asymptotic decider you know from the lecture.

- Finally draw the isoline segments for $v = -1$