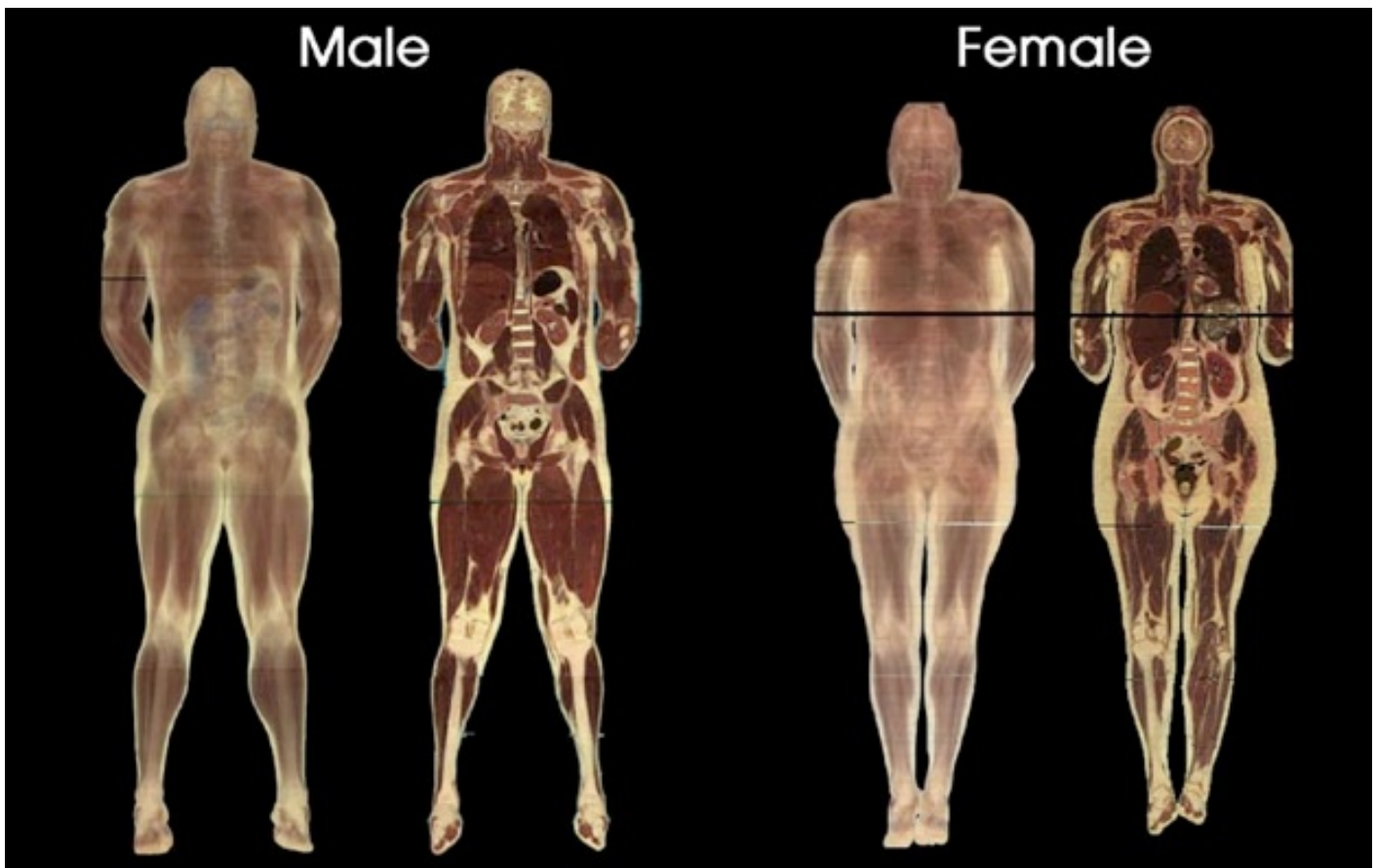


CS530 - Spring 2020

Programming Assignment 2 - Isosurfaces



Key Dates

Handed out: February 5, 2020

Due: February 18, 2020 before 11:59 pm

Objective

The second assignment focuses on the visualization of three-dimensional scalar fields using **isosurfaces**. Specifically, we will consider a medical imaging dataset corresponding to CT scans of a female subject. Your objective will be to combine isosurfaces, transparency, clipping planes and color mapping to reveal in your visualization important anatomical structures present in the data.

Background

The data corresponds to [CT scans](#) of a human female head and feet, courtesy of the [Visible Human Project](#). For each dataset I am providing you with both high and low resolution versions. The small resolution was created by halving the size of the original data along each direction (it is therefore 8 *times* smaller). For each resolution, you will find both a CT volume and an additional dataset corresponding to the magnitude of the gradient of the CT image. As a reminder, the gradient of a scalar field is the vector-valued derivative of the scalar field. Here however, we are only interested in the magnitude of the derivative. In each case, the CT data is of type **unsigned short**, yielding values between 0 and 65535, while the gradient magnitude data is of type **float**. Note that the small resolution is mainly provided as a convenience for the testing phase of your implementation and I expect to see high quality results at full resolution in your reports. The interesting features in this dataset

correspond to skin, bones, and muscles. Note that other soft tissues (e.g., fat or brain) are too difficult to extract.

Overview

Your experimentation with the head dataset should allow you to identify values that correspond to the skin, skull, and muscles. In the foot dataset, you will use isosurfaces to reveal the bone structure as well the skin and surrounding muscles. Note that for each dataset, the CT and gradient magnitude volumes will offer you complementary means to identify interesting features. To permit the selection of good isovalues, you will create a GUI that will let you select individual values associated with the various structures you are looking for. A slider bar will allow you to interactively browse the value range in search for the proper isovalue, while two additional slider bars will allow you to select a range of gradient magnitudes to act as a filter on the constructed isosurface. To mitigate the occlusion caused by the nested structure of the surfaces, you will be using transparency and clipping planes. The latter will allow you to cut away the occluding part of the surfaces to reveal internal structures. Finally you will use color mapping to convey quantitative properties of the selected isosurfaces.

Task 1 - Interactive Isosurfacing

The first task in this project consists in visualizing a 3D scalar dataset using isosurfacing while supporting the interactive modification of the corresponding isovalue. This initial step will allow you to identify a set of isovalues that correspond to the major anatomical structures present in each dataset. Specifically, skin, bones, and muscles.

Implementation

Write a program that uses [vtkContourFilter](#) to extract isosurfaces and ties it to a slider bar GUI to interactively manipulate the isovalue. To address occlusion issues, your program will offer interactive control over three clipping planes implemented as a [vtkClipPolyData](#) filter used with an implicit function called [vtkPlanes](#), which allows you to clip away portions of the volume to reveal internal structures. You will also add a color bar to your render window to show the meaning of the colors used in your visualization. This is done with the help of [vtkScalarBarActor](#). An example showing how to use this class is available [here](#). Note that this example uses a data file contained in the VTKData distribution. If you do not have it, it can be found on [github](#). Specifically, your program must satisfy following requirements.

- Take in input the name of the scalar volume to be visualized from the command line
- Perform an isosurface visualization of the dataset
- Provide a slider bar to control the isovalue to be used within a range of interesting values
- Ensure consistency between slider bar selection and isosurface
- Provide three additional slider bars GUI to control the position of three clipping planes moving along the X, Y, and Z coordinate axis, respectively
- Show the color scale through a color bar
- Optionally import the initial isovalue and initial X, Y, and Z clipping plane positions from the command line

API

Your program will have following API:

```
> python isosurface.py <data> [--val <value>] [--clip <X> <Y> <Z>]
```

where <data> is the 3D scalar dataset to visualize, <value> is the optional initial isovalue to be used by the program, and <X> <Y> <Z> are the optional initial positions of the 3 clipping planes.

Report

Include in your report high-quality images showing each of the major anatomical structures that you were able to extract with isosurfaces. Each image should include a color bar. Indicate the isovalues that you identified for each structure and the visual settings associated with each visualization. Use clipping planes at your discretion

to facilitate the visual inspection of your results. In addition, include answers to following questions in your report.

1. Which isosurfaces look the most interesting? Justify your answer.
2. How did you select the position of clipping planes?

Task 2 - Value vs. Gradient Magnitude

Now that you have identified interesting isosurfaces corresponding to major anatomical structures in each dataset, your second task is to color map the value of the gradient magnitude on those isosurfaces. In other words, you will visualize the values that the gradient magnitude takes at the set of positions defined by each isosurface. This visualization should give you additional insight into the properties of each isosurface.

Implementation

Write a program that takes both scalar volume and gradient magnitude in input as well as a set of pre-selected isovalues in [Task 1](#) and perform the color coding of the gradient magnitude on the isosurfaces. More precisely, you will use [vtkProbeFilter](#) jointly on the geometry of the isosurfaces and on the gradient magnitude volume to associate each vertex of the isosurfaces with the corresponding gradient magnitude value. The color mapping itself will be controlled by a color map supplied by the user. Specifically, your program must meet following requirements:

- Take in input the name of the scalar volume and the corresponding gradient magnitude volume from the command line
- Take in input the name of a file containing a set of isovalues **(1)** to be used for isosurface extraction
- Perform the resampling of the gradient magnitude on these isosurfaces and apply color mapping to visualize the resulting values
- Optionally take in input a color map to be used for the color mapping of gradient magnitude on the isosurfaces, otherwise resort to a default color map
- Provide the same three sliders GUI **(2)** used in [Task 1](#) to control the position of 3 clipping planes to be used in the visualization
- Include a color bar to document the selected color map

Notes

(1) The isovalues to be used by the program must be included in a file (see API below).

(2) The GUI only controls the clipping planes. There is no isovalue slider for this task.

API

Your program will have following API:

```
> python isogm.py <data> <gradmag> <isoval> [--cmap <colors>] [--clip <X> <Y> <Z>]
```

where **<data>** is the 3D scalar dataset, **<gradmag>** is the corresponding gradient magnitude, **<isoval>** is the name of a file containing the isovalues to be used, **<colors>** is the optional name of a file containing a color map definition with following syntax:

- Lines preceded by the character '#' are comments and should be ignored
- Each non-comment line must contain a scalar value followed by the R, G, and B values of the associated color.

Finally, **<X> <Y> <Z>** are the optional initial positions of the 3 clipping planes.

Report

Include in your report images of the color mapped isosurfaces computed by your program. Indicate clearly for each surface the corresponding isovalue. Each image should include a color bar. In addition, answer following questions:

1. What differences can you identify between the various isosurfaces in terms of their associated gradient magnitude distribution?
2. How do you interpret these results? Justify your answer.
3. What does that tell you about the value of the resulting visualization? Explain.

Task 3 - Two-dimensional Transfer Function

Combining the insight you gained from [Task 1](#) and [Task 2](#) you will now use the gradient magnitude data to filter out unwanted portions of your isosurfaces. Specifically, [Task 2](#) has shown you what values of the gradient magnitude coincide with certain portions of your isosurface. You can therefore use this information to discard from an isosurface uninteresting portions by specifying a gradient magnitude interval outside which isosurface points should be removed. This kind of downstream filtering can be achieved by using [vtkClipPolyData](#) again but this time you will be applying it directly to the gradient magnitude values attached to the vertices of the isosurface. In fact, you will need to use two such filters for a given $[grad_{min}, grad_{max}]$ gradient magnitude interval, one to discard all the values less than $grad_{min}$ and one to discard all the values larger than $grad_{max}$ among the values that passed the first filtering stage.

Implementation

You will write a program similar to the one you wrote for [Task 2](#) except that this time, the gradient magnitude information will not be directly displayed on the isosurface after resampling but instead used to determine which points should be removed from the isosurface in the visualization pipeline. Your program should allow the user to interactively modify the $[grad_{min}, grad_{max}]$ selection interval to facilitate the identification of an optimal range. Specifically, your program should meet following requirements:

- Take in input the name of both scalar volume and gradient magnitude datasets to be processed from the command line
- Optionally take in input the initial isovalue to consider
- Provide a GUI with two slider bars to allow the user to control the range (min et max) of gradient magnitude values to select
- Provide an additional slider bar to control the value of the selected isovalue
- Provide the same GUI as in [Task 1](#) to control the position of 3 clipping planes to be used in the visualization
- Perform a resampling of the gradient magnitude on the isosurface
- Filter the isosurface using two consecutive [vtkClipPolyData](#) consistent with the gradient magnitude range currently selected
- Visualize the resulting filtered isosurface

Note that in contrast to [Task 2](#), you will only be considering one isosurface at a time here. This is meant to facilitate the adjustment of the gradient magnitude range on a per surface basis.

API

Your program will have following API:

```
> python iso2dtf.py <data> <gradientmag> [--val <val>] [--clip <X> <Y> <Z>]
```

where **<data>** is the 3D scalar dataset and **<gradientmag>** is the corresponding gradient magnitude to visualize, and **<X>** **<Y>** **<Z>** are the optional initial positions of the 3 clipping planes.

Report

Include in your report high quality images for each dataset, corresponding to each filtered isosurface, while precisely indicating the matching isovalue. Indicate for each visualization which parameters / visual settings were used to create the image. In addition, provide an answer to following questions, with images to support your answers as necessary.

1. To what extent did the gradient magnitude filtering help in refining the isosurface selection? Be specific.
2. Which isosurfaces benefited the most from this filtering? Why?

Task 4 - Complete visualization

Using the isovalues that you identified in [Task 1](#) and the gradient magnitude filtering ranges that you discovered in [Task 3](#) to precisely characterize the various anatomical structures present in the data, you will now create a visualization where all the filtered isosurfaces are shown simultaneously using color and transparency.

Implementation

For this final implementation task, you will write a program that incorporates the various features of the programs written so far and uses transparency in addition to clipping planes to allow for all relevant isosurfaces to be shown simultaneously without excessive occlusion. Remember to use [Depth Peeling](#) to achieve correct transparency results! Specifically, your program must satisfy following requirements.

- Take in input from the command line the name of both scalar volume and gradient magnitude datasets to be processed
- Take in input the name of a file that specifies, for each isosurface, the corresponding isovalue, the associated gradient magnitude range, and the associated color and transparency.
- Perform a resampling of the gradient magnitude on each isosurface
- Filter each isosurface using two consecutive *vtkClipPolyData* consistent with the gradient magnitude range selected for that isosurface
- Visualize the resulting filtered isosurfaces with correct colors and transparency
- Provide the same GUI as in [Task 1](#) to control the position of 3 clipping planes to be used in the visualization

API

Your program will have following API:

```
> python isocomplete.py <data> <gradientmag> <params> [--clip <X> <Y> <Z>]
```

where **<data>** is the 3D scalar dataset, **<gradientmag>** is the corresponding gradient magnitude to visualize, **<params>** is the name of a file containing all the information necessary to select, filter, and visualize the isosurfaces. The format of this file will be as follows.

- Lines starting with '#' will be considered comments and ignored.
- Each following non-comment line that is not a comment will indicate an isovalue, a gradient magnitude range (min then max value), a RGB color and an opacity. It will have the form "**<isovalue> <grad_min> <grad_max> <R> <G> <alpha>**" where "**<grad_min> <grad_max>**" indicates the gradient magnitude filtering range for the isosurface, **<R> <G> **" specify the color associated with **<isovalue>**, and **<alpha>** is the associated opacity.

Report

Include in your report, high quality images showing the results produced by your method for your particular selection of parameters for each dataset.

- Comment on your selection of the transparency for each isosurface.
- How does transparency benefit your visualization? Explain.

Summary Analysis

Comment on the effectiveness of isosurfaces for these medical imaging datasets in your report. Isosurfaces in general are widely used in medical applications.

1. What explanation can you propose for this success?

2. Comment on the quality of the images you were able to obtain in each case.
3. Discuss any shortcomings of the isosurfacing technique you may have come across in this project.
4. Comment on the role and meaning of gradient magnitude to filter isosurfaces.
5. Comment on the benefits and limitations of transparency and clipping planes to enhance the visualization.

Datasets

The datasets are available online. All datasets are of type `vtkStructuredPoints`, which is itself a specialized type of `vtkImageData`.

Feet: CT ([small](#), [large](#)), gradient magnitude ([small](#), [large](#))

Head: CT ([small](#), [large](#)), gradient magnitude ([small](#), [large](#))

Submission

Submit your solution for this project on Blackboard before **February 18, 2020, 11:59 pm**. Refer to the instructions below.

- Include all program files
- Make sure that your programs have the correct API!
- For [Task 4](#), include the necessary parameter file for each dataset named `head_params.txt` and `feet_params.txt`.
- Include **high resolution sample images** showing results for each task.
- Include a **html or PDF report** summarizing what you have done and answering all the questions asked. The report should include high-resolution images.
- Include all submitted files in a single directory named `<myLogin>_p2`, where `<myLogin>` is your Purdue login.
- **Do not** include binary file
- **Do not** include data files
- **Do not** use absolute paths in your code

Last modified February 5, 2020