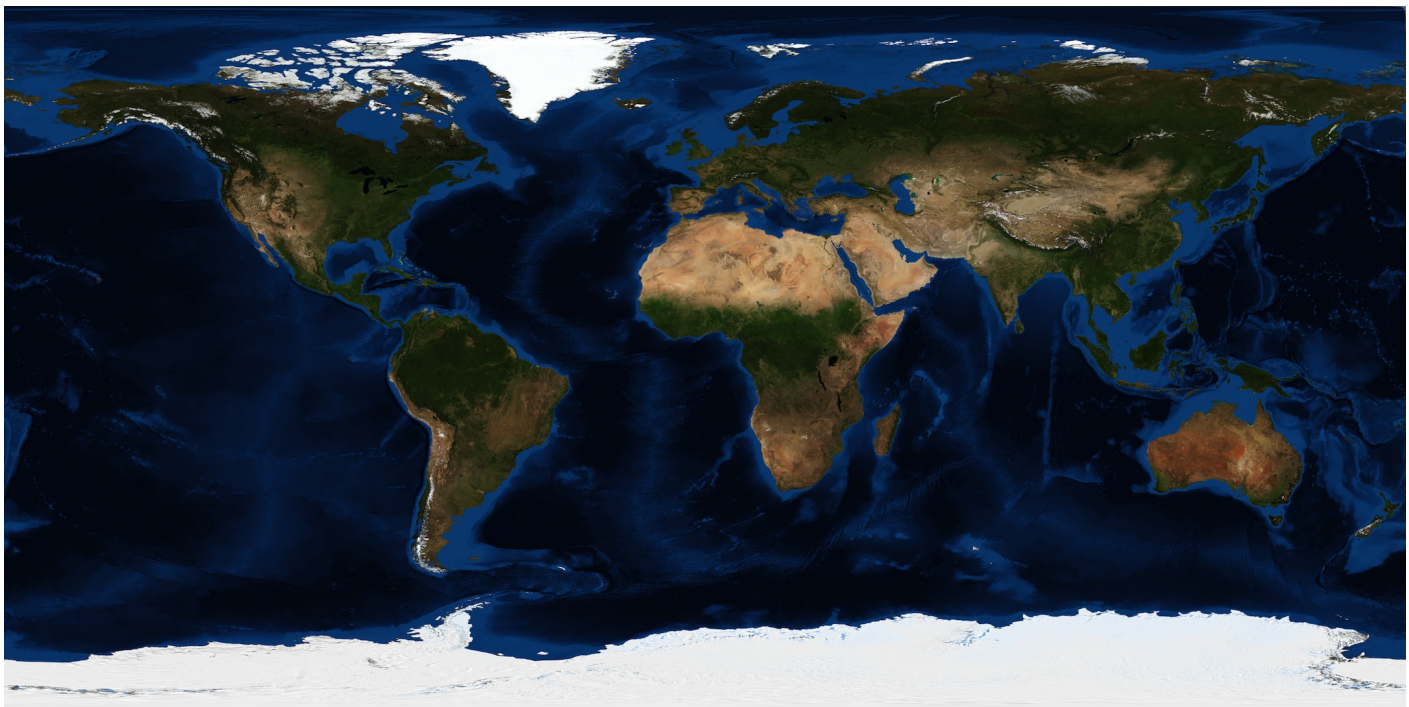
CS530 - Spring 2020
Project 1 - Geovisualization

Key Dates

Handed out: Wednesday, January 22, 2020
Due: Tuesday, February 4, 2020 before 11:59:59 pm



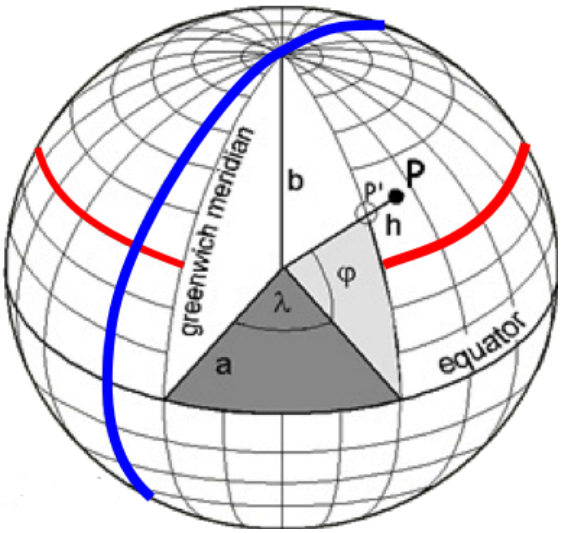
Objectives

In this first project, you will familiarize yourself with VTK while experimenting with several techniques to visualize 2D scalar data. The topic here is **geovisualization**, in other words the visualization of geographic data. Specifically, you will be visualizing a dataset that combines bathymetry/topography data with satellite imagery of the earth surface. Your visualization will include a graphical user interface that interactively controls certain aspects of the visualization.

Background

Bathymetry measures the underwater depth of the ocean floor while topography measures land elevation with respect to sea level. Together, they describe the earth surface's geometry. For simplicity, I use the term *elevation* below to jointly refer to both. NASA's Blue Marble project offers high resolution elevation datasets along with satellite imagery of the earth surface acquired at different times of the year. In this project you will learn how to use VTK to create compelling interactive visualizations of the elevation data.

Practically, the data consists of grayscale images (relief / topography) and color images (satellite views) with resolutions up to 86400 x 43200 (3.7 gigapixels). In each image, the earth surface is sampled at regular angular intervals spanning the domain $[-\pi, \pi] \times [-\pi/2, \pi/2]$ in spherical coordinates (see illustration below).



Specifically, each row of the image corresponds to a parallel (line of constant latitude, shown in red), while each column corresponds to a meridian (line of constant longitude, shown in blue). Note that the North and South poles (where parallels degenerate to points) are each represented by an entire row of uniform value.

Task 1. Height Map Visualization and Texture Mapping (30%)

Task summary

Your first task consists in visualizing the elevation dataset through an **interactive height map textured with the satellite image**. See explanations below.

Height map

A **height map** is a mapping from a gray-scale (scalar) image to a curved surface, whereby each (x, y) position associated with value f in the image is mapped to the 3D position (x, y, f). In the case of the elevation dataset, the resulting surface matches (by definition) the earth elevation profile.

Computing a height map in VTK is done with the help of the **`vtkWarpScalar`** filter, whose basic usage is demonstrated in *ImageWarp.py* (found here).

Interactivity

`vtkWarpScalar` provides a **scale factor** that allows the user to control the amount by which the image is moved up or down for a given scalar value at each point: $\{(x, y), f\}$ is in fact mapped to (x, y, Kf) by the algorithm, where K is the scale factor. This control mechanism can be tied to a **slider GUI** to permit the interactive manipulation of the scale factor and update the visualization accordingly.

Texture mapping

Height mapping as described previously produces a geometric representation of the Earth's surface. In the absence of additional visual cues, however, key aspects of the data, such as the continents' coast lines, remain ambiguous. To remedy this problem you will use the available satellite image through **texture mapping**.

In VTK, textures are stored in **`vtkTexture`** objects that can then be supplied to **`vtkActor`** to enable texture mapping during the rendering stage. Note that this mapping is only possible if the vertices of the target geometry (here, the height surface) are equipped with *texture coordinates* that specify their matching location on the texture. It is therefore important to point out that the supplied elevation datasets already contain texture coordinates among the point attributes.

Implementation

You will write for this task a program that satisfies the following requirements:

- Import the necessary files (elevation and satellite image) using the appropriate VTK readers

- Compute a height map representation of the elevation with **`vtkWarpScalar`**
- Texture map the satellite image onto the height map geometry
- Visualize the result
- Provide a slider bar GUI to interactively manipulate the scale factor of **`vtkWarpScalar`**
- Maintain consistency between visualization and GUI

API

Your program will have following API:

`> heightfield[.py] <elevation> <image>`

where **`<elevation>`** is the path of the elevation data file and **`<image>`** is the path of the satellite picture.

Report

- Include in your report images of the visualization results produced by your program
- Make sure to select camera angles that convey the presence of a 3rd dimension
- Provide several images showing the results obtained for different scale factors
- Include close-up images

Task 2. Isocontours and Color Mapping (30%)

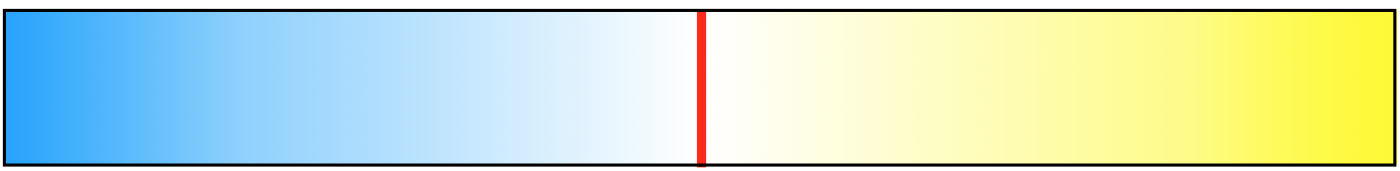
Task summary

For the second task, you will **visualize the elevation dataset using isocontours**.

Isocontours

To highlight specific values of the elevation, one needs isocontours, which are also known as *contour lines*, *isolines*, or more formally *level sets*. An isocontour corresponds to the set of points where the considered function has a particular value (the *pre-image* of that function at that value). In 2D, isocontours form closed curves. Note that we will study this topic in great detail when we consider isosurfaces in the coming weeks. By selecting a number of discrete values between the minimum and the maximum values of the data set one can get a good illustration of the value distribution across the domain. To compute isocontours in the elevation dataset you will use **`vtkContourFilter`**, the API of which is discussed here.

Specifically, you will create a visualization that shows isocontours associated with elevation values ranging from -10,000 meters to +8,000 meters in 1,000 meters increments. To improve the clarity of your visualization, you will wrap the isocontour into tubes using **`vtkTubeFilter`**. Note that the radius of your tubes must be selected carefully (not too big to avoid occlusion and not too small to make them easily distinguishable). To facilitate this selection you will create a GUI slider that lets you interactively modify this radius. In addition, you will apply a color map to the isocontours, in other words, assign to each isocontour a color that represents its value. Here, I am asking you to use the color scale shown below, where the **maximal ocean depth** will be mapped to **blue**, the **maximum mountain height** will be mapped to **yellow**, and values close to the sea level (either positive or negative) will be mapped to **desaturated colors**. The **sea level** itself (elevation 0) will be shown in **red**. Note that color maps in VTK are defined through **`vtkColorTransferFunction`** that are then supplied to the mapper via **`mapper.SetLookupTable(<color_tf>)`**, where **`<color_tf>`** is the name of your color transfer function.



To provide context to the isocontours, you will apply the same texture mapping solution as in the previous task, this time however to a flat geometry. Observe that you may need to tweak the blue and yellow ends of the color map defined above to improve the contrast of your isocontours with the underlying satellite image.

Implementation

You will write for this task a program that meets the following requirements:

- Import the elevation dataset and satellite using the appropriate VTK readers
- Define a **`vtkContourFilter`** to create a series of isocontours in 1,000 meters increments in the interval -10,000 meters to +8,000 meters.
- Display the resulting curves as tubes using **`vtkTubeFilter`**
- Color the tubes using the indicated color scale and **`vtkColorTransferFunction`**
- Provide a GUI slider bar to control the radius of the tubes
- Texture map a satellite image to the dataset
- Keep visualization and GUI selection consistent

API

The API for your program should be as follows:

```
> isocontour[.py] <elevation> <image>
```

where **`<elevation>`** is the path of the elevation data file and **`<image>`** is the path of the satellite picture.

Report

- Include in your report high quality images of the results produced by your program
- Provide images showing the results obtained for several radius selections
- Include close-up images

Task 3. Putting It All Together On A Sphere (30%)

Task summary

For this third and final task, you will **integrate height field representation, texture mapping, and isocontours and display them on a sphere** to create a compelling visualization of the elevation dataset.

Interactive scalar field visualization

The fourth task in this project consists in combining the various visualization techniques that you implemented previously in a single visualization. Specifically, you should write a program that displays the elevation dataset as a height field, applies to it the satellite imagery using texture mapping (as done in Task 1), and visualizes isocontours as tubes (cf. Task 2). As discussed previously, your visualization should include a slider bar to control the scaling factor of the height field representation. Note that you will set the radius of the isocontours to a fixed value that you found appropriate in Task 2).

Mapping to a sphere

In contrast to what you did so far, you will display height field, texture, and isocontours **on a sphere** instead of a plane. Indeed, the results achieved so far do not convey the actual shape of the various continents due to the significant distortion that is caused by the latitude / longitude parameterization of the image. To remedy this problem we need to visualize the data directly on a sphere.

To make your task easier, I am providing you with a sphere dataset that contains elevation values, normals, and texture coordinates. Note that the sphere that I am giving you is already scaled to the radius of the earth (in meters).

Implementation

You will write for this task a program that satisfies the following requirements:

- Import the elevation dataset on a sphere and satellite image using the appropriate VTK readers
- Apply **`vtkWarpScalar`** to visualize the elevation data as height field
- Provide a slider bar to control the scaling factor of **`vtkWarpScalar`**
- Texture map the satellite image onto the height field

- Apply **`vtkContourFilter`** to create a series of isocontours in 1,000 meters increments in the interval -10,000 meters to +8,000 meters
- Display the resulting curves as tubes using **`vtkTubeFilter`**
- Color the tubes using the prescribed color map and **`vtkColorTransferFunction`**
- Draw the tubes directly on the height field
- Keep visualization and GUI selection consistent

API

The API for your program should be as follows:

`> view_earth[.py] <sphere_elevation> <image>`

where **`<sphere_elevation>`** is the path of the elevation on a sphere data file and **`<image>`** is the path of the satellite image.

Report

- Include in your report high quality images of the results produced by your program
- Provide images showing the results obtained for several scaling factors
- Include images that show you results for different parts of the world and different camera perspectives
- Include close-up images

Discussion (10%)

Answer the following questions in your report. Be as specific as possible and draw from your experience in this assignment to justify your opinion.

Considering the height map technique used in Task 1:

1. What properties of the dataset were effectively visualized with this technique?
2. What are in your opinion the main limitations of this technique and how could you address them?
3. How useful did you find the slider interface in your usage of the height field representation?
4. How effective do you find this visualization technique for this dataset?

Considering the level sets considered in Task 2:

1. What specific aspects of the data were readily visible with isocontours?
2. How useful did you find the color map and why?

Considering the sphere representation in Task 3:

1. What benefits did you see to the perform the visualization on a sphere?
2. How did the resulting visualization compare to the previous ones?

Considering your findings in tasks 1, 2, and 3:

1. Did you find that the combined use of these visualization techniques in Task 3 improved upon the results of each technique applied separately? Why or why not?

Datasets

As indicated in preamble, the data used in this project is courtesy of NASA Earth Observatory and available on the web site of the Blue Marble project. Since the server can be slow at times, I am providing you with a local copy of the relevant files. Note that I converted the bathymetry / topography datasets to a single **`vtkImageData`** to simplify your programs. The dataset is available in 4 different resolutions for convenience. Similarly, the satellite image of the Earth is provided in 3 different resolutions. In both cases, use a low resolution version to test your implementation and use for your report the highest resolution that your computer can

handle. Note that some of these files are **really large** (see below) so make sure to download only what you need.

Elevation datasets:

- small version: 540 x 270, 1.6 MB
- medium version: 1080 x 540, 6.0 MB
- large version: 2160 x 1080, 24 MB
- very large version: 4320 x 2160, 89 MB

Same elevation datasets on sphere geometry:

- small version: 540 x 270, 9.2 MB
- medium version: 1080 x 540, 38 MB
- large version: 2160 x 1080, 152 MB
- very large version: 4320 x 2160, 608 MB

Satellite image:

- medium version: 2160 x 1080, 573 KB
- large version: 5400 x 2700, 2.3 MB
- (very) large version: 21600 x 10800, 36 MB

Report

Please refer to the class syllabus for late policy and formatting guidelines of project reports. Your report should also contain your answers to the questions asked in the discussion section. As a reminder, **this project must be completed individually**.

Submission

Submit your project on Blackboard before **February 4 at 11:59:59 pm**. Please observe following instructions.

- Include the 3 program files along with any other source code you may have
- Make sure that your programs have the correct API!
- Include high resolution sample images showing results for each task.
- Include a PDF report describing what you have done and answering the questions asked above. The report should include high-resolution images. Additional images can be provided as part of the project submission.
- Include all submitted files in a single tar zip'ed directory named `<login>_p1`, where `<login>` is your Purdue login.
- **Do not** include any binary file
- **Do not** include any data files
- **Do not** use hardcoded paths (relative or absolute) in your programs.

Last modified January 22, 2020