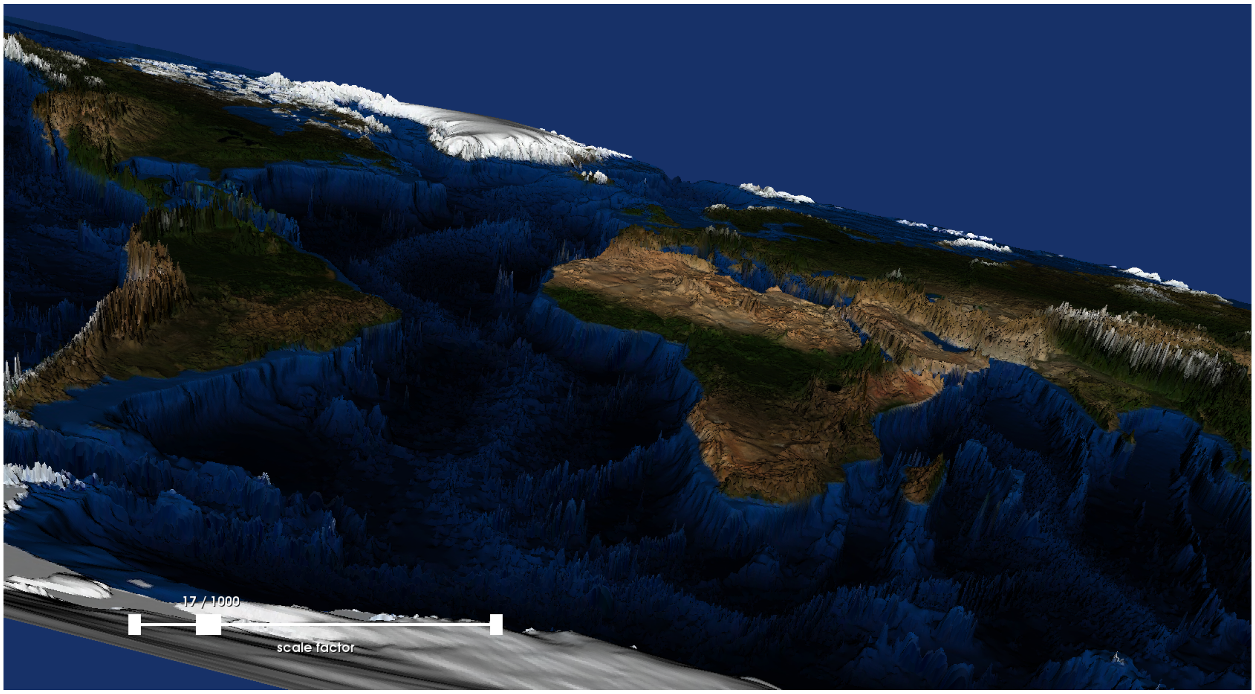
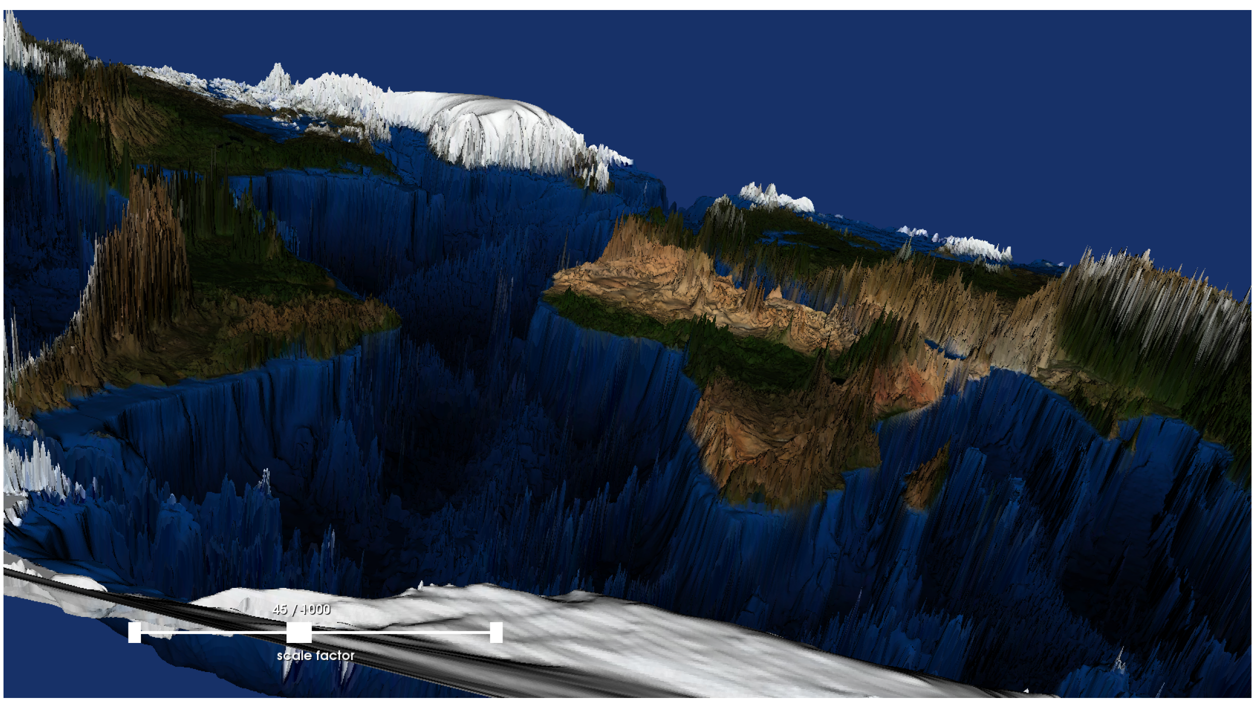
CS-530 Project 1 Report

Shih-Feng Yang

Task 1



Scale Factor: 0.017

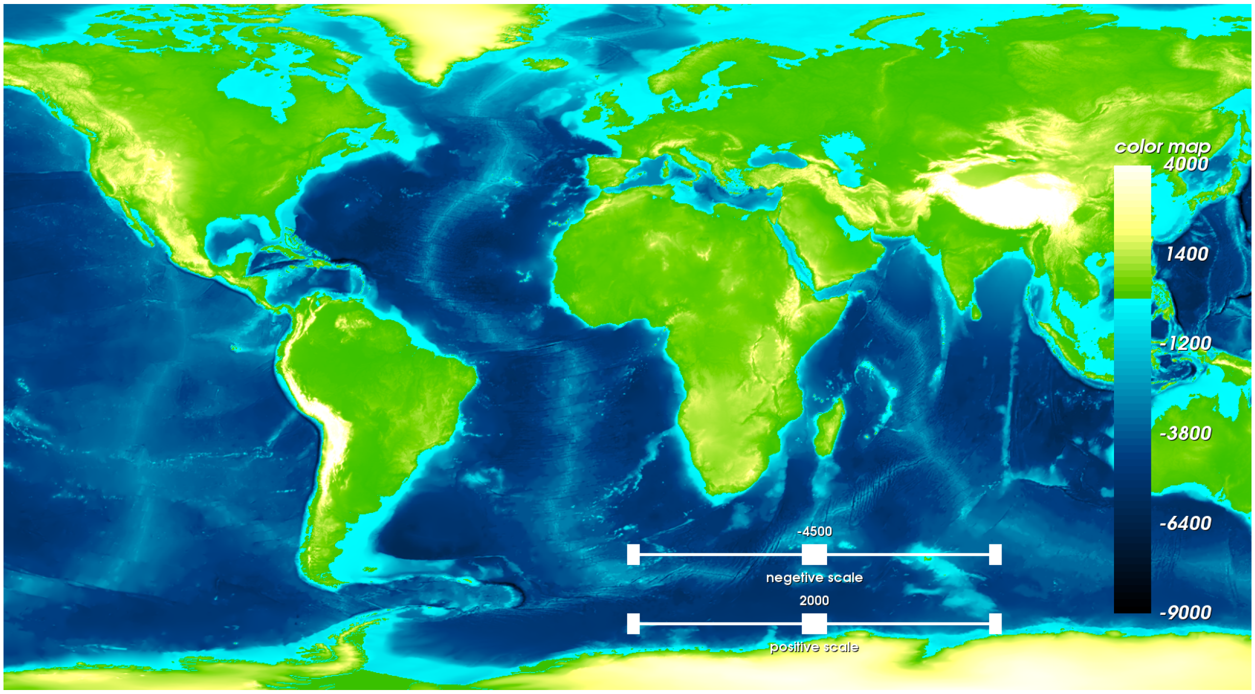


Scale Factor: 0.045

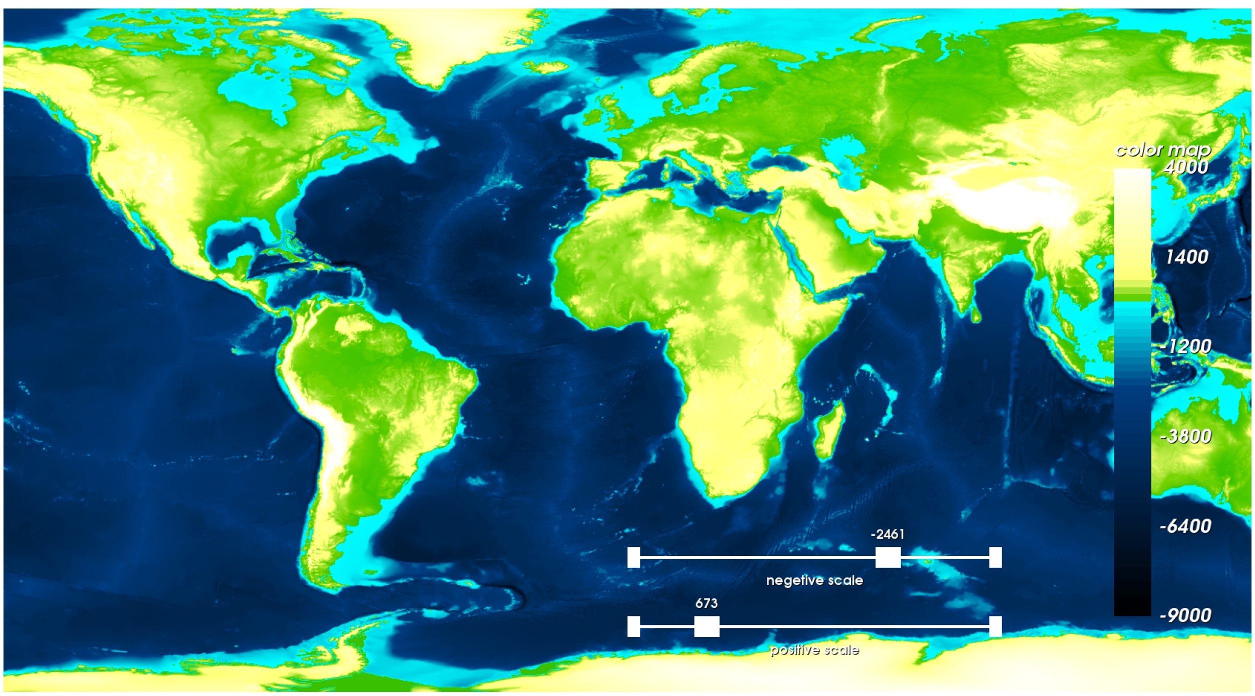
Strategy
1. Use vtkJPEGReader to load the satellite image.
2. Use vtkStructuredPointsReader to load the bathymetry texture coordinates.
3. Load bathymetry dataset by vtkImageDataGeometryFilter and then by vtkWarpScalar.
4. Load the vtkJPEGReader of the satellite image by vtkTexture.
5. Load the warp data into vtkDataSetMapper.
6. Initial the vtkActor. Load the vtkDataSetMapper by SetMapper function, and load the vtkTexture by SetTexture.
7. Create a vtkSliderRepresentation2D and setup its attributes.
8. Create a vtkSliderWidget. Load the representation as the created vtkSliderRepresentation2D, and register the callback function vtkSliderCallback for the InteractionEvent event.
9. Define the vtkSliderCallback to update the scale factor of vtkWarpScalar by the function SetScaleFactor.
10. Start the program.

Discussion
1. **What properties of the dataset were effectively visualized with this technique?**
   The exact height/depth differences on the map.
2. **What are in your opinion the main limitations of this technique and how would you address them?**
   We need to zoom in and use different angles of view to observe the map. It is hard to see the exact height/depth status by a single snapshot of the map.
3. **How useful did you find the slider interface in your usage of the height field representation?**
   By using the slider to change the scalar factor, it enables users to amplify the altitude differences between different places, such as mountains and ocean. Moreover, it enable us to observe subtle altitude differences in a single area.
4. **How effective do you find this visualization technique for this dataset?**
   For this dataset, it works good to see the subtle altitude differences on the map, including plains, ocean, mountains, plateaus, ocean trenches, etc.

Task 2



Negative: -4500, Positive: 2000



Negative: -2461, Positive: 673

Strategy

1. Use vtkStructuredPointsReader to load the bathymetry texture coordinates.
2. Create a color mapping by vtkColorTransferFunction. Use AddRGBPoint to add 6 points:
   1. Altitude -9000, RGB(0,0,0)
   2. Altitude -4500, RGB(0,0.25,0.5), the default negative altitude variable is -4500
   3. Altitude -1, RGB(0,1,1)
   4. Altitude 0, RGB(0.25,0.75,0)
   5. Altitude 2000, RGB(1,1,0.5), the default positive altitude variable is 2000
   6. Altitude 4000, RGB(1,1,1)
3. Load bathymetry dataset by vtkImageDataGeometryFilter and then by vtkDataSetMapper.
4. Create a color bar by vtkScalarBarActor, and use SetLookupTable function to load the lookup table in vtkDataSetMapper by GetLookupTable function.
5. Initial the vtkActor. Load the vtkDataSetMapper by SetMapper function.
6. Setup the vtkRenderer. Load the vtkActor and the vtkScalarBarActor.
7. Like task 1, create two vtkSliderWidget (slider1, slider2) and two vtkSliderCallback functions (callback1, callback2).
8. Setup callback1 to update the negative altitude variable and callback2 to update the positive attitude variable.
9. Start the program.

Discussion

1. **How did you select the missing bathymetric values to form the color map?**
   The middle value in the color map helps us distinguish the altitude differences because it divided all possible colors into two group. For example, if we define the middle depth value as -8000 (the depth range from -9000 ~ 0), there will be 50% colors in the color map can be used for (-9000,-8000] and 50% colors in the color map for (-8000,0).

   Thus, it would be better to set the middle value in the color map as **the median of all height/depth values**. For example, if there are 2000 points (for each point, its depth is between -9000 and 0) on the map, I would select the depth median of all points as the middle value in the color map. Let’s assume -6000 is the median. This selection can provide 50% colors for 1000 points in (-9000, -6000] and 50% colors for another 1000 points in (-6000, 0). This way can equally make every point on the map more distinguishable.

2. **What specific aspects of the geography were readily visible with the color map in your opinion?**
   The steep geography is readily visible. For example, mountains near plains, plateaus and coasts.
3. **How useful did you find the GUI in this case?**
   To verify if the missing bathymetric values can be adopted well, the sliders can help us easier to see the effectiveness after using specific values. Besides, the color map bar helps us to see the color distribution in different altitude values clearly.

4. **What would you describe as shortcomings of the color map and how would you address them?**
   When most of the altitude values are distributed in a small number of groups, the effectiveness of color map representation will be bad. For example, given 2000 points which altitude values are in (-9000, 0), if 1500 points are in (-8000, -7900) and the rest 500 points are in (-500, -450), then only 2-3 colors will be used on the map. Therefore, it will be hard to tell the differences between the points in the (-8000, -7900) group.
5. **How effective do you find this visualization technique for this dataset?**
   It works good on mountains, plateaus, and coasts. However, it doesn’t work well on ocean and plains. Comparing to task 1, we can directly see the altitude differences on a single snapshot.