SUBJECT CODE: CS 6630
SUBJECT: SCIENTIFIC VISUALIZATION

LEENA KORA
EMAIL:leenak@cs.utah.edu
CADE_ACCOUNT: kora

PART1 (Height Fields)

SUB_PART1: Electric Field Potential Data

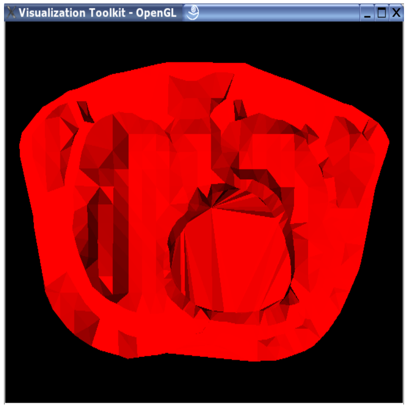
FILES USED

CODE FILE: heightfd1.tcl writeplydat.tcl
DATA FILES PROVIDED:assignment1.pts assignment1.data
POLYDATA FILE: humpoints.vtk

VTK PIPELINE: The visualization model has two pipelines.They are visualization pipeline and graphics pipeline.The role of the visualization pipeline is to transform information into graphics data. The role of the graphics pipeline is to transform graphical data into pictures.

Unconnected points -> POLYDATA READER -> DELAUNAY2D FILTER ->POLYDATA MAPPER -> ACTOR

The assignment1.pts file gives the coordinates of the data points with Z coordinate always zero. There are 618 point locations derived from an MRI scan. The assignment1.data file is the electric field potential at each point, one line per datapoint. First writeplydat.tcl creates POLYDATA-type dataset using the potential value as the Z coordinate and then writes the dataset in appropriate vtk format in file humpoints.vtk. The filter "vtkPolyData" is used to convert the dataset into polydata. Then the unconnected point set is triangulated using "vtkDelaunay2D" filter. The output of the filter is a polygonal dataset. The output of the "vtkDelaunay2D" filter is fed as input to the "vtkPolyDataMapper" filter which renders geometry using analytic primitives such as points, lines and triangular strips. Then the "vtkActor" filter is connected to the pipeline which deals with rendering attributes such as surface properties, representation and texture maps. The final image is then rendered into an interactive window. Following is the screenshot of the height field generated from the thorax data points.



SUB_PART2: Elevation Data

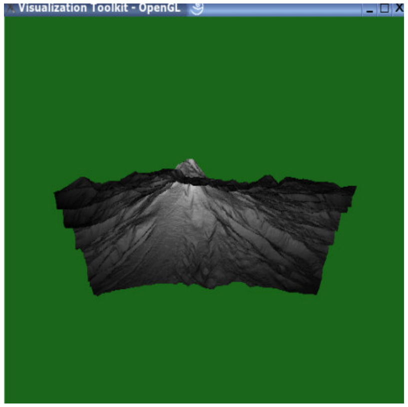
FILES USED

CODE FILE: heightfd2.tcl
DATA FILES PROVIDED: MtHood.pgm MtHood.gif

VTK PIPELINE

Gray_scale IMAGE -> PNM READER -> IMAGEDATAGEOMETRY FILTER -> WARP FILTER -> DATASET MAPPER -> ACTOR

The script heightfd2.tcl produces the height field of Mt. Hood from a .pgm file MtHood.pgm. The vtk filter "vtkPNMReader" is used to read .pgm data which creates structured point dataset. This dataset is fed as input to the "vtkImageDataGeometryFilter" which is used to extract the geometry. Then "vtkWarpScalar" filter scales the data in the z direction to produce the height field. The below screen shot is generated from the above pipeline.



PART2 (Contour Maps)

FILES USED

CODE FILE: contour.tcl
DATA FILES PROVIDED: brain.vtk body.vtk
MY DATA FILES: Leena1.jpg Leena2.jpg

The script contour.tcl generates three contour maps for three datasets brain.vtk, body.vtk and Leena1.jpg. The first two datasets are in vtk format that is they are .vtk files and the third one is in .jpg format.

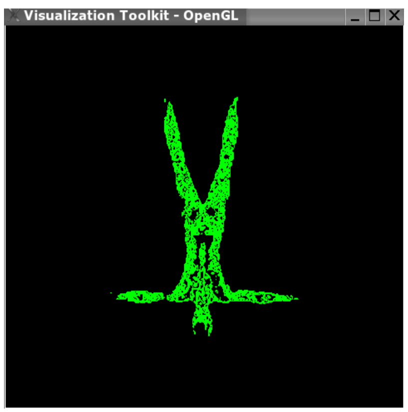
Brain and Body Data

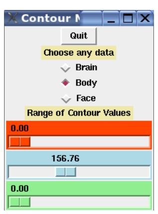
The first two datasets are read using "vtkDataSetReader" filters which can read any vtk dataset file. The output of vtkDataSetReader filter is fed as input to the vtkContourFilter which produces contour values. Then the filters "vtkDataSetMapper" are added to the pipeline to map the data and is rendered on the vtk interactive screen. The user is allowed to change the contour values through scroll bars and they can even switch between the images. The below figure shows the pipelines to process both "brain.vtk" and "body.vtk" data.

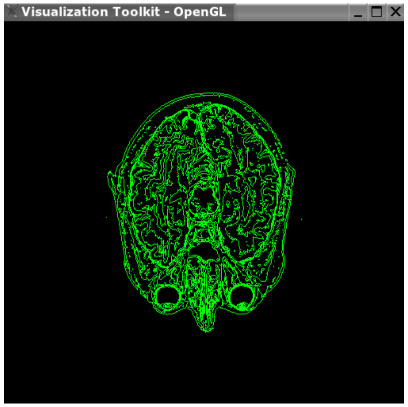
Structured points of body -> DATASET READER -> CONTOUR FILTER -> DATASET MAPPER -> ACTOR

Structured points of brain -> DATASET READER -> CONTOUR FILTER -> DATASET MAPPER -> ACTOR

The next two contour maps shows the body and brain which have 10 contour lines. The body.vtk has values ranging from 97 to 252.The body.vtk has values ranging from 0 to 236.





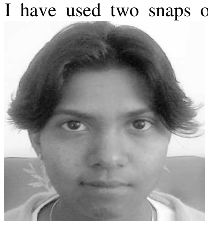


Feature Recognition through Contour Maps

FILES USED

DATA FILES USED:Leena1.jpg Leena2.jpg

I have used two snaps of my face. One with plain expression and the other with smiling gesture. Both are .jpg files. First i have converted them to gray-scale image using gimp tool. Following are two images that i have used.




The gray_scale dataset is then passed as input to the "vtkJPEGReader" filter which reads any .jpeg files. The dataset is then fed to the "vtkImageToStructuredPoints" filter which converts it into structured points. Then the processed data is passed to "vtkContourFilter" filter to generate co ntour values which are then mapped and rendered on the window using the filters "vtkDataSetMapper" and "vtkActor". The pipeline for the process is as shown below.

Gray_scale IMAGE -> JPEG READER -> IMAGETOSTRUCTUREDPOINTS FILTER -> CONTOUR FILTER -> DATASET MAPPER -> ACTOR

Questions

Q 1.What properties does a dataset need to have in order to have contour lines that make sense (and contribute to the visualization)?

Dataset should indicate continuous function in order for contour lines to make sense.They should have a range of values. The contours help us to indicate how the data is changing contributing to the visualization. The dataset should have some kind of relationship to make sense otherwise the contour map will be undistinct and does not make sense.

Q 2.Where are a few places that these properties can be found?

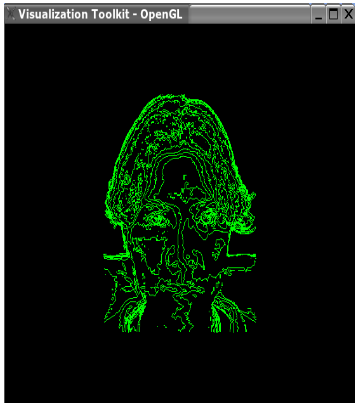
Creation of contour lines is uasually used to represent elevation and any Z-value data such as cost, mineral content or temperature. They are also used in medical images.

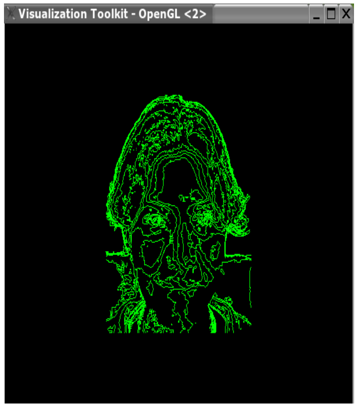
Q 3.What data are you visualizing?

I am visualizing two .jpg images of my face. One with plain expression and the other with smiling expression.

Q 4.Why do you think it is interesting?

when I visualized both the images keeping the number of contour lines as constant I got the following maps.





The second map shows more smaller circles indicating the elevation of the cheeks compared to the first map. This idea can be implemented in Artificial Intelligence. This concept can be used as feature recognition technique for Robots. Now a days scientists are trying to include human features in Robots and they are expecting human behaviour from them. The smiling face's cheeks are different from an unsmiling face and the smiling face conveys more positive human qualities. The robot with scanners in its eyes can scan the face at intervals and produce contour maps. The contour maps can then be compared to check the behaviour of the human in front of it and can behave or respond similarly.