KTH Stockholm
CSC :: HPCViz
Visualization, Spring 2016, DD2257
Tino Weinkauf

Homework assignment No. 03

Due April 29, 2016

Task 3.1: Marching Squares Implementation — 2+9+4 P

You are given different two-dimensional scalar fields sampled on a regular grid. Table 1 gives an overview of the data sets that come with this task. We may load other data sets during the interview. We may use any isovalue during the interview.

(a) Load a given data set from disk and visualize the mesh of the regular grid using line primitives. Respect the bounding box size and the relative cell size of the data set for the visualization.

(b) Implement the *Marching Squares* algorithm for the extraction of isocontours of the scalar field. Visualize both the grid mesh and the isocontours for a given isovalue c using line primitives. The isovalue c is a parameter that can be changed by the user. To solve ambiguities, implement the *Midpoint Decider* strategy. A good test data set is `SimpleGrid.am`.

(c) Implement the *Asymptotic Decider* strategy to solve ambiguities. Compare the results with the Midpoint Decider strategy. Find examples (data set and isovalue) for which the visualizations differ significantly.

Task 3.2: Isocontour Visualization of Complex Data Set — 5 P

You are given a larger two-dimensional scalar field `IsabelTemparature.am`. It represents the temperature over North America during the Isabel hurricane in 2010. See Figure 1.

Use your implementation of the *Marching Squares* algorithm to visualize n different equidistant isocontours of the data set between the minimal and maximal temperature in that data set. The integer n is a parameter that can be changed by the user. *Do not visualize the grid mesh in this task.*

(Bonus Points: +2) Assign a different color to each isocontour in a perceptually appropriate manner.

Technical details for both tasks:

Create a new 2D experiment. Allow the user to define a file name using a string property `ADD_STRING_PROP()` and isovalues / isovalue numbers using float / integer properties `ADD_FLOAT32_PROP()` / `ADD_INT32_PROP()`. Different deciders can be selected using, e.g., a boolean property that is added by `ADD_BOOLEAN_PROP()`.

You can load the `.am` regular grid data sets using the `load(file_name)` method of the `ScalarField2` class defined in the header file `math/Field2.hpp`. The data dimensions and bounding box extents can be obtained by the `dims()`, `boundMin()`, and `boundMax()` methods, respectively. The position and data value of a node (i, j) can be obtained by the methods `nodePosition(i,j)` and `nodeScalar(i,j)`. The scalar field can be sampled inside of the bounding box using `sampleScalar(xx)` and `sampleScalar(x,y)`, where `xx` has type `StaticVector<float32,2>` and `x` and `y` are of scalar `float32` type.

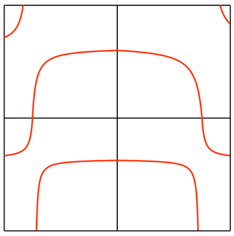
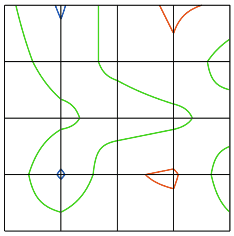
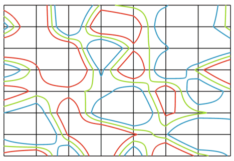
Data Set	Isovalues	Visualization
`SmallGrid.am`	0	
`SimpleGrid.am`	1.5, 5, 8.5	
`WideGrid.am`	0.4, 0.5, 0.6	

Table 1: Scalar fields for Task 1. The structure of the uniform grid is shown together with different isocontours.

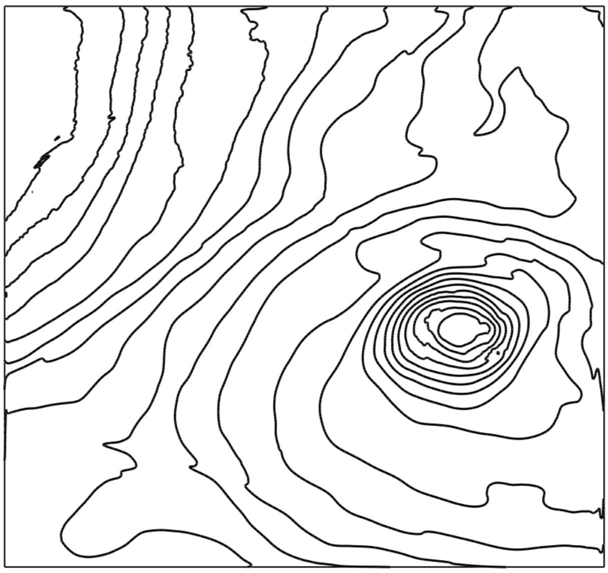


Figure 1: Isocontour visualization of the temperature over North America during the Isabel hurricane in 2010.