

CS 519 Scientific Visualization Machine Problem 1: Due 9/29/2017 at 11:55pm

Programming Language and Tools

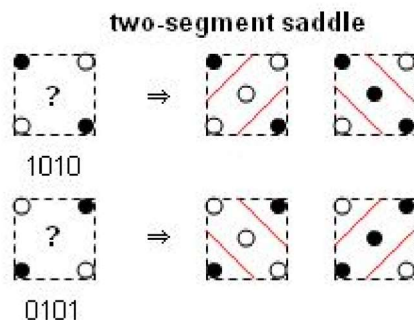
For this MP, use HTML5 and JavaScript. If you wish, you can use WebGL, although you can accomplish the MPs just using the HTML5 *canvas* element as you have done in class.

Marching Squares

You will implement marching squares to draw 2D contour lines. The algorithm is described in section 5.3.1 of *Data Visualization: Principles and Practice* by Telea. You can also refer to the Wikipedia article (http://en.wikipedia.org/wiki/Marching_squares). You need not use a table-based algorithm if you do not wish to do so.

Implementation Requirements:

1. Your user interface should support the specification of an arbitrary number of contour values. You could use a text box to allow the user to type in a list of values, for example. Similarly, your code should support the creation of multiple contour lines.
2. You should resolve the ambiguous case for marching squares by averaging the function values at the corners of the square to generate a sample in the square center as shown



3. In addition to rendering the contour lines, your code should display the function across the domain using the rainbow color map. In the interest of saving you time, you don't need to draw a legend depicting the mapping of color to values, although you would obviously do so for professional work.
4. Implement another colormap that is neither greyscale nor rainbow. Design a colormap with a clear break at the function value 0.5.

¹ "Marching-squares-isoline". Licensed under CC BY 3.0 via Wikipedia -

<http://en.wikipedia.org/wiki/File:Marching-squares-isoline.png#mediaviewer/File:Marching-squares-isoline.png>

5. Your code should consist of the following files:

ScalarVis.html

ScalarVis.js

SciVisMath.js

UGrid2D.js

You may base your code off of startup code available on GitHub at

https://github.com/shaffer1/Ullinois_SciVis/tree/master/MP1

Interpolation

You should also implement a capability to sample the functions given above in a regular grid pattern and fill in the missing values using bilinear interpolation. The values should then be rendered using the colormaps specified previously just as if they were real function values.

You should implement a control to specify a sampling frequency. This can be a text box in which you enter a single number, or some other control. For example, if you enter “4” then every 4th pixel in x and y will be sample of the function ((0,0) and (0,4) and (4,0), etc.).

You should also implement a button that will switch between interpolation and the default view of rendering the image directly from the function.

Grading

The MP is worth 10% of your total course grade. It will be graded out of 10 points as follows:

- 3 points: Correct Implementation of Marching Squares
- 2 point: Support for multiple iso-values and contour lines
- 1 point: Correct implementation of the new colormap
- 2 point: Implement function sampling
- 2 points: Correct implementation of bilinear interpolation

Submission

Submission will be through Compass. You are required to submit the following:

1. Your source code (all four files listed above).
2. An image demonstrating marching squares applied to the Gaussian function.
3. An image demonstrating your new colormap applied to the Gaussian function.
4. An image demonstrating your interpolation capability applied the Gaussian function, using every 10th pixel as a sample.