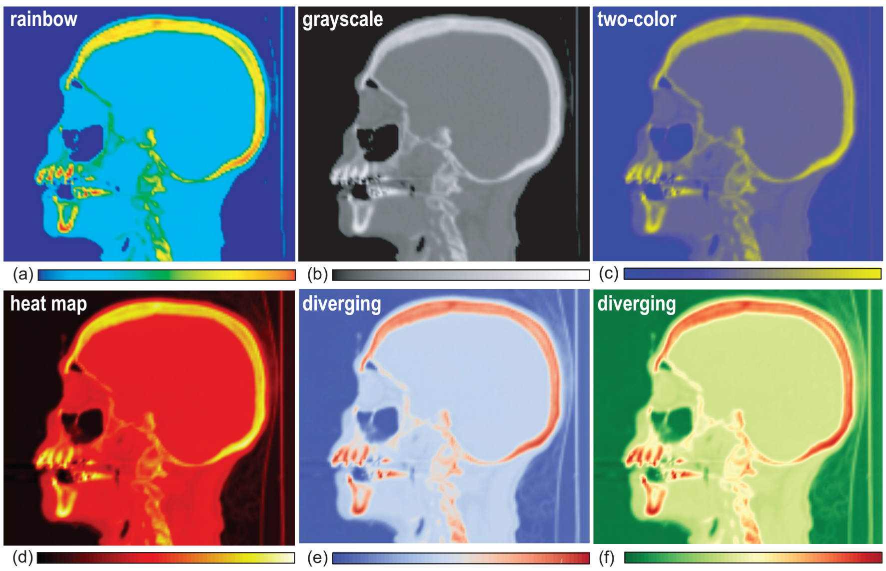
Exercises for

Chapter 5: Scalar Visualization

1 EXERCISE 1

Consider the simple scalar visualization of a 2D slice from a human head scan shown in the figure below. Here, six different colormaps are used. Explain, for each of the sub-figures, what are the advantages (if any) and disadvantages (if any) of the respective colormap.

Hints: Consider what types of structures you can easily see in one visualization as compared to another visualization.



Scalar visualization of a 2D slice dataset using six different colormaps (see Chapter 5).

2 EXERCISE 2

Consider an application where we use color-mapping to visualize a scalar field defined on a simple 2D uniform grid consisting of quad cells. The scalar field values, recorded at the cell vertices, are in the range $[s_{min}, s_{max}]$. Next to the color-mapped grid, we show, as usual, a color legend, where the two extreme colors (at the endpoints of the color legend) correspond to the values s_{min} and s_{max}, respectively. Given this set-up, can we say for sure that any color shown in the color legend will appear at at least one point (pixel) of the color-mapped grid? If so, argue why. If not, detail at least one situation when this will not occur.

3 EXERCISE 3

A simple, though not perfect, way to create 2D contours or isolines, is to use a so-caled *delta colormap*. Given a scalar contour-value (or isovalue) a, and a dataset with the scalars in the range $[m, M]$, a delta colormap maps all scalar values in the ranges $[m, a-d]$ and $[a+d, M]$ to one color, and the values in the range $[a-d, a+d]$ to another color. Here, d is typically very small as compared to the scalar range $M-m$. Consider now the application of this delta colormap to a scalar field stored on a simple uniform 2D grid having quad cells, as compared to drawing the equivalent isoline on the same grid. Detail at least two drawbacks of the delta colormap as compared to the isoline visualization. Next, explain which parameters we could fine-tune (*e.g.*, value of d, sampling rate of the grid, color-interpolation techniques used on the grid cells, or other parameters) in order to improve the quality of the delta-colormap-based visualization.

4 EXERCISE 4

Contouring can be implemented by various algorithms. In this context, the marching cubes algorithm (and its variations such as marching squares or marching tetrahedra) propose several optimizations. Compared to a 'naive' implementation of contouring, that does not use the marching technique, what are the main computational advantages of the marching technique?

5 EXERCISE 5

Consider a 2D constant scalar field, $f(x, y) = k$, for all values of (x, y). What will be the result of applying the marching squares technique on this field for an iso-value equal to k? Does the result depend on the cell type or dimension – for instance, do we obtain a different result if we used marching triangles on a 2D unstructured triangle-grid rather than marching squares on a 2D quad grid?

Hints: Consider the vertex-coding performed by the marching algorithms.

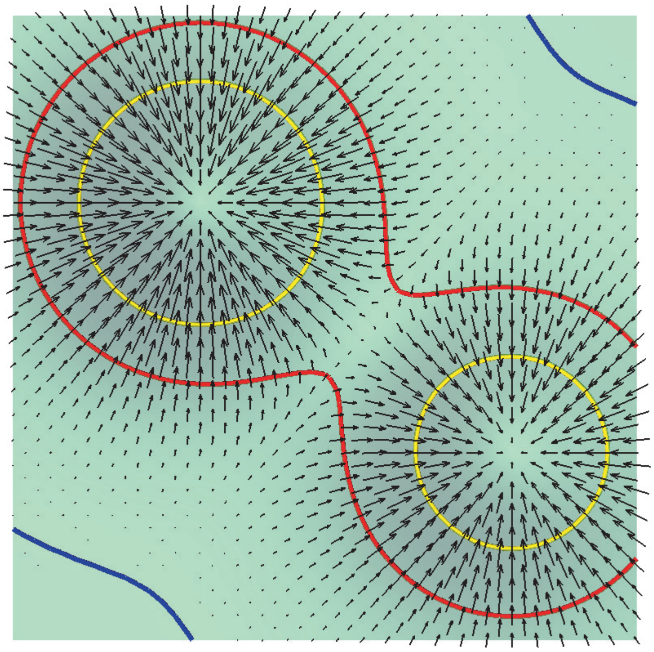
6 EXERCISE 6

An isoline, as computed by marching squares, is represented by a 2D polyline, or set of line-segments. Can two such line segments from the same isoline intersect (we do not consider segments that share endpoints to intersect). Can two such line segments from different isolines of the same scalar field intersect? If your answer is yes, sketch the situation by showing the quad grid, vertex values, isoline segments, and their intersection. If your answer is no, argue it based on the implementation details of marching squares.

7 EXERCISE 7

Consider a two-variable scalar function $z = f(x, y)$ which is differentiable everywhere. Figure 5.9 (also displayed below) shows that the gradient of the function is a vector which is locally orthogonal to the contours (or isolines) of the function. Give a mathematical proof of this.

Hints: Consider the tangent vector to the contour at a point (x, y), and express the variation (increase or decrease) of the function's value in a small neighborhood around the respective point.



Function height plot (seen from above), with three isolines (red, green, and blue), and function-gradient arrow plot (see Chapter 5).

8 EXERCISE 8

Contouring and slicing are commutative operation, in the sense that slicing a 3D isosurface with a plane yields the same result as contouring the 2D data-slice obtained by slicing the input 3D scalar volume with the same plane. Consider now the set of 2D contours obtained from a 3D scalar volume by applying the 2D marching squares algorithm on a set of closely and uniformly spaced 2D data-slices extracted from that volume. Describe (in as much detail as possible) how you would use this set of 2D contours to reconstruct the corresponding 3D isosurface.

9 EXERCISE 9

Consider an application in which the user extracts an isosurface, and would like to visualize how thick the resulting shape is, using e.g. color mapping. That is, each vertex of the isosurface's unstructured mesh should be assigned a (positive) scalar value indicating how thick the shape is at that location. For this problem:

- Propose a definition of shape thickness that would match our intuitive understanding (you can support your explanation by drawings, if needed)
- Propose an algorithm to compute the above definition, given an isosurface stored as an unstructured mesh
- Discuss the computational complexity of your proposed algorithm
- Discuss the possible challenges of your proposal (e.g., configurations or shapes for which your definition would not deliver the expected result, and/or the algorithm would not work correctly with respect to your definition).

Hints:

- Consider first the related problem of computing the thickness of a 2D contour, before moving to the 3D case.
- Consider first that the contour is closed, before treating the case when it is open because it reaches the boundary of the dataset.

10 EXERCISE 10

One of the challenges of practical use of isosurfaces in 3D is selecting an 'interesting' scalar value, for which one obtains an 'interesting' isosurface. We cannot solve this challenge in general, since different applications may have different definition of what an 'interesting' isosurface is. However, consider an application where you want to automatically select a small number of iso-values (say, 5) and display these to give a quick overview of the data changes. How would you automatically select these isovalues?

Hints: Define, first, what you consider to be the interesting structures in the data, *e.g.*, areas of rapid scalar variation, or areas where an isosurface changes topology).

End of Exercises for
Chapter 5: Scalar Visualization
