Data Visualization (CSCI 627/490)

Vector Field Visualization

Dr. David Koop
Visualizing Volume (3D) Data

- 2D visualization slice images (or multi-planar reformating MPR)
- *Indirect* 3D visualization isosurfaces (or surface-shaded display SSD)
- *Direct* 3D visualization (direct volume rendering DVR)
Generating Isolines (Isovalue = 5)
Marching Squares

2. Marching Cubes and Variants

Figure 2.10. Red, positive regions and blue, negative regions for each square configuration. The green isocontour is part of the positive region. Black vertices are positive.

Proof of Properties 1 & 2:
The Marching Squares isocontour consists of a finite set of line segments, so it is piecewise linear. These line segments intersect only at their endpoints and thus form a triangulation of the isocontour. The endpoints of these line segments lie on the grid edges, confirming Property 2.

Property 3.
The isocontour intersects every bipolar grid edge at exactly one point.

Property 4.
The isocontour does not intersect any negative or strictly positive grid edges.

Proof of Properties 3 & 4:
Each isocontour edge is contained in a grid square. Since the grid squares are convex, only isocontour edges with endpoints (vertices) on the grid edge intersect the grid edge. If the grid edge has one positive and one negative endpoint, the unique location of the isocontour vertex on the grid edge is determined by linear interpolation. Thus the isocontour intersects a bipolar grid edge at only one point.

If the grid edge is negative or strictly positive, then no isocontour vertex lies on the grid edge. Thus the isocontour does not intersect negative or strictly positive grid edges.

Within each grid square the isocontour partitions the grid square into two regions. Let the positive region for a grid square $c$ be the set of points which can be reached by a path $\zeta$ from a positive vertex. More precisely, a point $p$ is in the positive region of $c$ if there is some path $\zeta \subset c$ connecting $p$ to a positive vertex of $c$ such that the interior of $\zeta$ does not intersect the isocontour. A point $p$ is in the negative region of $c$ if there is some path $\zeta \subset c$ connecting $p$ to a negative vertex of $c$ such that $\zeta$ does not intersect the isocontour. Since any path $\zeta \subset c$ from a positive to a negative vertex must intersect the isocontour, the positive and negative regions form a partition of the square $c$. Figure 2.10 illustrates the positive and negative regions, colored red and blue, respectively, for each square configuration.

[R. Wenger, 2013]
Ambiguous Configurations

- Either works for marching squares, this isn't the case for 3D

![Ambiguous square configurations](image)

**Figure 2.12.** Ambiguous square configurations.

**Figure 2.13.** Topologically distinct isocontours created by using different isocontours for the ambiguous configuration in the central grid square.

While the choice of isocontours for the ambiguous configurations changes the isocontour topology, any of the choices will produce isocontours that are 1-manifolds and strictly separate strictly positive vertices from negative vertices. As we shall see, this is not true in three dimensions.

2.3 Marching Cubes

2.3.1 Algorithm

The three-dimensional Marching Cubes algorithm follows precisely the steps in the two-dimensional Marching Squares algorithm. Input to the Marching Cubes algorithm is a scalar grid with two topological distinct isocontours created by different resolutions of the ambiguous configurations. The first isocontour has two components while the second has one.

[R. Wenger, 2013]
3D: Marching Cubes

- Same idea, more cases [Lorensen and Cline, 1987]

![Diagram of Marching Cubes](image)

Figure 2.16. Isosurfaces for twenty-two distinct cube configurations.

[Image: Diagram showing 22 distinct cube configurations with labels for positive vertices ranging from zero to eight.]
Multiple Isosurfaces

- Topographical maps have multiple isolines to show elevation trends
- Problem in 3D? **Occlusion**
- Solution? Transparent surfaces
- Issues:
  - Think about color in order to make each surface visible
  - Compositing: how do colors "add up" with multiple surfaces
  - How to determine good isovalues?

[J. Kniss, 2002]
Volume Rendering vs. Isosurfacing

(a) Direct volume rendered
(b) Isosurface rendered

[Kindlmann, 1998]
Volume Ray Casting

Image Plane

Eye

Data Set

[Levine]
Volume Ray Casting

Image Plane

Eye

Data Set

[Levine]
Types of Compositing

- max intensity
- accumulate
- average
- first

[Levine and Weiskopf/Machiraju/Möller]
Accumulation

• If we're not just calculating a single number (max, average) or a position (first), how do we determine the accumulation?

• Assume each value has an associated color (c) and opacity (α)

• Over operator (back-to-front):
  - \( c = \alpha_f \cdot c_f + (1-\alpha_f) \cdot \alpha_b \cdot c_b \)
  - \( \alpha = \alpha_f + (1-\alpha_f) \cdot \alpha_b \)

• Order is important!

Blue Last

Blue First
Transfer Functions

• Where do the colors and opacities come from?
• Idea is that each voxel emits/absorbs light based on its scalar value
• …but users get to choose how that happens
• x-axis: color region definitions, y-axis: opacity
Assignment 5

• Map of Citi Bike trips
  - Multiple Views
  - Linked Highlighting
  - Filtering
  - Aggregation

• Due Monday, Nov. 23
Projects

- Keep working on implementation
- Be creative, don't copy
- Think about interaction
- Presentations on the last day of class (Dec. 3)
  - Plan to use Blackboard
  - Upload to Blackboard beforehand in case of technical issues
Course Evaluations
ParaView Examples
Vector Field Visualization
Examples of Vector Fields

Wind [earth.nullschool.net, 2014]
Examples of Vector Fields

Wind [earth.nullschool.net, 2014]
Examples of Vector Fields

Computational Fluid Dynamics [numerical]
Examples of Vector Fields

Earthquake Ground Surface Movement [H. Yu et. al., SC2004]
Examples of Vector Fields

Gradient Vector Fields
Examples of Vector Fields
Fields in Visualization

Scalar Fields
(Order-0 Tensor Fields)

Vector Fields
(Order-1 Tensor Fields)

Tensor Fields
(Order-2+)

Each point in space has an associated...

Scalar: $s_0$

Vector: $egin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix}$

Tensor: $egin{bmatrix} \sigma_{00} & \sigma_{01} & \sigma_{02} \\ \sigma_{10} & \sigma_{11} & \sigma_{12} \\ \sigma_{20} & \sigma_{21} & \sigma_{22} \end{bmatrix}$
Visualizing Vector Fields

- Direct: Glyphs, Render statistics as scalars
- Geometry: Streamlines and variants
- Textures: Line Integral Convolution (LIC)
- Topology: Extract relevant features and draw them
Glyphs

• Represent each vector with a symbol
• Hedgehogs are primitive glyphs (glyph is a line)
• ParaView Example
Glyphs

• Represent each vector with a symbol
• Hedgehogs are primitive glyphs (glyph is a line)
• Glyphs that show direction and/or magnitude can convey more information
• If we have a separate scalar value, how might we encode that?
• Clutter issues
Glyphs

- For vector fields, can encode direction, magnitude, scalar value
- Good:
  - Show precise local measures
  - Can encode scalar information as color
- Bad:
  - Possible sampling issues
  - Clutter (Occlusion): Can remove some points to help
  - Clutter is worse in higher dimensions
Rendering Vector Field Statistics as Scalars

- Many statistics we can compute for vector fields:
  - Magnitude
  - Vorticity
  - Curvature

- These are scalars, can color with our scalar field visualization techniques (e.g. volume rendering)
Streamlines & Variants

- Trace a line along the direction of the vectors
- Streamlines are always tangent to the vector field
- Basic Particle Tracing:
  1. Set a starting point (seed)
  2. Take a step in the direction of the vector at that point
  3. Adjust direction based on the vector where you are now
  4. Go to Step 2 and Repeat
Example

- Elliptical path
- Suppose we have the actual equation
- Given point \((x,y)\), the vector is at that point is \([v_x, v_y]\) where
  - \(v_x = -y\)
  - \(v_y = (1/2)x\)
- Want a streamline starting at \((0,-1)\)
Euler Integration – Example

2D analytic field (no need of grid and interpolation):

\[ v_x = \frac{dx}{dt} = -y \]
\[ v_y = \frac{dy}{dt} = \frac{x}{2} \]

Sample arrows:

Ground truth flows form ellipses.

\[ [x, y] \rightarrow [-y, (1/2)x], \text{ Step: 0.5} \]

Some Glyphs

[via Levine]
Streamlines (Step 1)

\[ [x, y] \rightarrow [-y, (1/2)x], \text{ Step: 0.5} \]
Euler Integration – Example

New point

\[ s_1 = s_0 + v(s_0) \cdot \Delta t = \left(\frac{1}{2} \right) T; \]

current flow vector \( v(s_1) = \left(1, \frac{1}{4}\right)^T; \)

\[ \frac{dx}{dt} = -y, \quad \frac{dy}{dt} = \frac{x}{2}. \]

Streamlines (Step 2)

\[ [x,y] \to [-y, (1/2)x], \text{ Step: 0.5} \]

Streamlines (Step 2)
Streamlines (Step 3)

\[ [x, y] \rightarrow [-y, (1/2)x], \text{ Step: 0.5} \]
Streamlines (Step 4)

\[ [x, y] \rightarrow [-y, (1/2)x], \text{ Step: 0.5} \]
Streamlines (Step 10)

\[ [x, y] \rightarrow [-y, (1/2)x], \text{ Step: 0.5} \]
Streamlines (Step 19)

\[ [x, y] \rightarrow [-y, (1/2)x], \text{ Step: 0.5} \]
Euler Method

- Seeking to approximate integration of the velocity over time
- Euler method is the starting point for approximating this
- Problems?
Euler Method

• Seeking to approximate integration of the velocity over time
• Euler method is the starting point for approximating this
• Problems?
  - Choice of step size is important
Euler Method

• Seeking to approximate integration of the velocity over time
• Euler method is the starting point for approximating this
• Problems?
  - Choice of step size is important
  - Choice of seed points are important
Euler Method

• Seeking to approximate integration of the velocity over time
• Euler method is the starting point for approximating this
• Problems?
  - Choice of step size is important
  - Choice of seed points are important
• Also remember that we have a field—we don't have measurements at every point (interpolation)
Comparison of Euler quality by step size.
Numerical Integration

• How do we generate accurate streamlines?
• Solving an ordinary differential equation

\[
\frac{dL}{dt} = v(L(t)) \quad L(0) = L_0
\]

where \(L\) is the streamline, \(v\) is the vector field, and \(t\) is “time”

• Solution:

\[
L(t + \Delta t) = L(t) + \int_{t}^{t+\Delta t} v(L(t))\,dt
\]
Higher-order methods

\[
\int_{t}^{t+\Delta t} v(L(t)) \, dt
\]

- Euler method (use single sample)

- Higher-order methods (Runge-Kutta) (use more samples)

[A. Mebarki]
Higher-Order Comparison

- Euler vs. Runge-Kutta
- RK-4: pays off only with complex flows
- Comparison

[Image via Levine]