Isosurfacing (Part 3)
Announcements

• Final project
  – Come in & talk with me
  – Upcoming:
    • how to use general purpose library (later this week)
    • how to use general purpose tool (next week)
    • → you can make beautiful movies

• Quiz #2: Weds, Feb 21\textsuperscript{st}. On isolines

• Projects 6 and 7 discussed this lecture
Review
Isolines vs Isosurfaces

• Isolines:
  – Input: scalar field over 2D space
  – Output: lines

• Isosurfaces:
  – Input: scalar field over 3D space
  – Output: surface

• Commonalities:
  – Reduce topological dimension by 1
  – Produce output where scalar field is constant
Big idea #1: approximate the isolines / isosurface

- Isolines: represent them with a minimal # of segments
- Isosurface: represent them with a minimal # of triangles
Big idea #2: pre-computation of all cases

If you knew which case you had, then you would know how to proceed

- Pre-compute correct answers for all 16 cases and store them in a lookup table
- For each cell, identify which case it is in
- Then use corresponding lookup table to generate isosurface
The 16 cases
Problem case: ambiguity!!

Solution: just pick one and go with it.
Physical interpretation of ambiguity

One way connects them up, the other separates them. What’s right?
Isosurfacing

• Will follow a very similar game plan.
  – Pre-compute correct answers for all cases and store them in a lookup table
  – For each cell, identify which case it is in
  – Then use corresponding lookup table to generate isosurface
Isosurfacing

Quiz: where should the isosurface go?
Isosurfacing

Quiz: where should the isosurface go?
Isosurfacing

Quiz: where should the isosurface go?
Isosurfacing

Quiz: where should the isosurface go?
Isosurfacing

Quiz: where should the isosurface go?
Isosurfacing

Quiz: where should the isosurface go?
Quiz: where should the isosurface go?
Quiz: where should the isosurface go?
Isosurfacing

Quiz: where should the isosurface go?
Isosurfacing

We need conventions!
Isosurfacing

We need conventions!
Isosurfacing

We need conventions!
Isosurfacing

We need conventions!
Isosurfacing

static int edges[12][2] =
    { {0,1}, {1,3}, {2,3}, {0,2},
      {4,5}, {5,7}, {6,7}, {4,6},
      {0,4}, {1,5}, {2,6}, {3,7} };
Isosurfacing

Quiz: where should the isosurface go?
Isosurfacing

Quiz: where should the isosurface go?
Isosurfacing

Quiz: where should the isosurface go?
Isosurfacing

Quiz: where should the isosurface go?
Isosurfacing

Quiz: where should the isosurface go?
This is our last two cells, side by side
Uh oh ... ambiguous case is causing problem
Problem case: ambiguity!!

2D Solution: just pick one and go with it.

3D: nope, that doesn’t work here
Uh oh ... ambiguous case is causing problem
Isosurfacing

Let’s try to make something consistent
Isosurfacing

Quiz: how many triangles will we need?

Let’s try to make something consistent
Let's try to make something consistent

No! This is got us into trouble before!
Isosurfacing

Let’s try to make something consistent
Let’s try to make something consistent

Isosurfacing
Isosurfacing

Let’s try to make something consistent
From a different angle...
Summary So Far For Ambiguities

• Ambiguities cause problems:
  – If you use one interpretation for one cell, and use the other interpretation for its neighboring cell, then you get gaps
  – Always making the “intuitive” choice does not solve the problem.

• If you choose consistently, then you can avoid these problems.
How to make consistent choices

• “Asymptotic Decider”
  – Analyze scalar field and make decision

Quiz: why does this result in consistent choices?
How to make consistent choices

• Conventions!
  – E.g., always separate lowest vertex
  – This is consistent across faces
  – This is how VTK (and case_checker) works
Project 6: What should we do?

• Our choices:
  – Figure out those conventions and reproduce them perfectly
    • Correct and additional effort
  – Ignore the conventions and accept gaps
    • Incorrect, but easier

• Note: ambiguous cases don’t come up a lot in practice
Equality

• Current case assignment:
  – $F(V) < \text{isovalue}: 0$
  – $F(V) > \text{isovalue}: 1$

• What if the field value at a vertex is equal to the isovalue?

Quiz: what is the physical interpretation of having $F(v) == \text{isovalue}$?
Equality Strategy

- Case assignment (incorporating inequality):
  - $F(V) < \text{isovalue}: 0$
  - $F(V) \geq \text{isovalue}: 1$
Equality Strategy

• Case assignment (incorporating inequality):
  – $F(V) < \text{isovalue}: 0$
  – $F(V) \geq \text{isovalue}: 1$

• Quiz: calculate isolines for isovalue = 5.
Accelerating Isosurfacing

- Marching Cubes:
  - For each cell,
    - Assign case
    - Use lookup table for case to generate geometry

Quiz: what is the computational complexity of this algorithm?

Answer: $O(n_{cells})$

Quiz: could we improve the computational complexity of this algorithm?
Accelerating Isosurfacing

• Marching Cubes with Scalar Trees:
  – Preprocessing step: calculate “scalar tree”
  – For each cell that contains the isovalue,
    • Assign case
    • Use lookup table for case to generate geometry

Quiz: what is the computational complexity of this algorithm?

Answer: \( O(\text{prep time}) + O(n\text{SelectedCells} \times \text{SearchTime}) + O(n\text{SelectedCells}) \)

Quiz: when would this be superior to naïve algorithm?
Scalar Trees

• The tree consists of an array of (min, max) scalar range pairs per node in the tree.

• The (min, max) range is determined from looking at the range of the children of the tree node.

• If the node is a leaf, then the range is determined by scanning the range of its corresponding cell.
  – Optimization: one leaf corresponds to multiple cells.
Scalar Tree: Example

- Cell 0 range: 0-2
- Cell 1 range: 1-3
- Cell 2 range: 0-3
- Cell 3 range: 4-8
- Cell 4 range: 0-3
- Cell 5 range: 2-6
- Cell 6 range: 8-10
- Cell 7 range: 7-9

This is the simplest version of this data structure.
Accelerating Isosurfacing

• Marching Cubes with Scalar Trees:
  – Preprocessing step: calculate “scalar tree”
  – For each cell that contains the isovalue,
    • Assign case
    • Use lookup table for case to generate geometry

Quiz: what is the computational complexity of this algorithm?

Answer: $O(n\text{Cells} \ast \log(n\text{Cells})) + O(n\text{SelectedCells} \ast \log(n\text{SelectedCells})) + O(n\text{SelectedCells})$

Quiz: when would this be superior to naïve algorithm?
Project 6

CIS 410/510: Project #6
Due 5pm February 15th, 2018
No late work will be accepted for this assignment.
Must be submitted via SVN (emails or Canvas not accepted / no credit).

Worth 2% of your grade

1) Using Subversion, checkout the files for this project.
   a. 410 students:
      svn co svn+ssh://USERNAME@ix.cs.uoregon.edu/home/users/hank/SVN/mc410
   b. 510 students:
      svn co svn+ssh://USERNAME@ix.cs.uoregon.edu/home/users/hank/SVN/mc510

   Note: many of 410 students do not appear to have active ix accounts. I am placing
   the key file for 410 (tricase.cxx) as well as a README on the class website. You can
   use that file to see your assignments until you get ix access going. All 510 students
   have active ix accounts, and can proceed with an SVN checkout.

2) We are crowdsourcing the marching cubes tables. Your name is associated with
   ~15 cases if you are in 410 and ~35 cases if you are in 510. Find each instance and
   replace it with the correct answers.
   a. As we discussed in class, one solution to the ambiguity problem is to enforce
      conventions for how to perform a split along the diagonal for ambiguous cases.
   b. We will not pursue that solution. Instead, we will expect that people put forward
      the “simplest” solution.

3) Make sure to commit your solutions to SVN. Emails and Canvas will not be
   accepted. Please plan ahead and get an ix account.
Project 7

• Generate an isosurface
  – We will use the crowd-sourced marching cubes tables from project 6
• Posted: Monday or Tuesday
• Due: February 20