CS5630/6630

Isosurfacing - Speed

Slides from Professor Hansen
Rendering

View point

Render $\iff$
Rendering &
Isosurface Extraction

Construct ⇔ Render ⇔

View point

Isovalue
Isosurface Extraction

• Complexity

n: number of cells
k: number of cells an isosurface intersect

n = 100
k = 16

n = 100
k = 70
Concepts

• **Active cell (k)**
  - a cell intersecting with isosurface

• **Inactive cell**
  - a cell not intersecting with isosurface
Octrees: Basic Idea

• **Construct an octree**
  - Every internal node stores (min, max) of intensity value of the subtree
  - Every external node is a cell

• **When traverse octree, compare isovalue $T$ with min and max of each internal node**
  - If $T$ is within interval (min, max), search the children
  - Otherwise, skip searching the children

• **Avoid traversals of inactive cells (which don’t intersect with the isosurface)**

• **Useful for interactive visualization (dynamic isovalues)**
Octrees

- One way to handle the big data problem is to use hierarchical data structures (hierarchical volumes)

8x8x8  →  4x4x4  →  2x2x2  →  ...

Less amount of data are required
Octrees

- A hierarchical volume can be constructed using the octree

```
Entire volume

Half in each dim

...```
Octrees

- An octree can be full or non-full
  - Full octree – each parent node has exact 8 children
  - Condition: all the dimensions are the same power of 2
  - A full octree example: $16 \times 16 \times 16$
    - Level 1: (8) $8 \times 8 \times 8$; level 2: (8) $4 \times 4 \times 4$
      - Level 3: (8) $2 \times 2 \times 2$; level 4: (8) $1 \times 1 \times 1$
Octrees (5)

• In general, there are two ways to construct an octree for a volume
  – Top-down:
    1 → 8 → 8x8 → 8x8x8 → 8x8x8x4 → ...
    → n (n = total number of data)
  – Bottom-up:
    n → n / 8 → n / 8x8 → n/8x8x8 → n/8x8x8x4 → ...
    → 1

• Which one has a better node/data ratio?
Other Methods

**MinMax Octree**
*Wilhelms and Van Gelder 90/92*

**Search Complexity:**
$O(k \log(n/k) + k)$
*Livnat, Shen and Johnson 96*
Extrema Graph: Concepts

• **Extrema points**: the grid points whose values are higher (or lower) than all its adjacent grid points
  - Maximum points
  - Minimum points
  - Corners of the entire dataset

• **Note**: saddle points are not extrema points

• **Graph**: A set of points (vertices) and edges (arcs) that connect the points

• **Connected graph**: all vertices are connected through the edges
Extrema Graph: Basic Idea

• Given a dataset, construct a connected graph containing all extrema points (the graph is called extrema graph)

• Traverse graph edges to find active cells

• Start with an active cell
  - Extract isosurface of this cell
  - Then look at its neighbors
  - Continuous perform the above operation
  - This approach (isosurface propagation) uses the spatial coherence of cells
Other Methods

Extrema Graphs
Itoh and Koyamada 94

Volume Thinning
Itoh, Yamaguchi and Kotamada 96

Search Complexity
- Avg $O(n \exp(2/3))$
- Worst case $O(n)$
Livnat, Shen and Johnson 96
The Span Space
The Span Space

Livnat, Shen, Johnson 96

- **Given:**
  - Data cells in 8D
- **Past (active list):**
  - Intervals in a 1D Value space
- **New:**
  - Points in the 2D Span Space
- **Benefit:**
  Points do not exhibit any spatial relationships
The Span Space

- **Search**
  - Find all the points
    - minimum < isovalue
    - isovalue < maximum
  - Semi-infinite area
    - Quadrant

![Diagram showing the span space with points and isovales](image_url)
The Span Space

- **Search for rectangles using Kd-tree**
  - $O(n \log(n))$ to build
  - Search Complexity
    - $O(\sqrt{n+k})$
- **Recursively divide each axis along median**
The Span Space

- **NOISE**: $O(\sqrt{n+k})$
  
  Livnat, Shen, Johnson 96

- **Optimal**: $O(\log(n)+k)$
  
  Cignoni et al. 96
  Better search algorithm
The Span Space

- **NOISE**: $O(\sqrt{n+k})$
  
  Livnat, Shen, Johnson 96

- **Optimal**: $O(\log(n)+k)$
  
  Cignoni et al. 96

- **ISSUE** (parallel)
  
  - $O(k)$, $O(\log(n)+k)$, $O(\text{?})$
  
  Shen, Hansen, Livnat 97
Beyond the Optimal Barrier?

- $O(n) \Rightarrow O(k \log(n)) \Rightarrow O(\sqrt{n} + k) \Rightarrow O(\log(n)+k)$

- Can we do better than $O(\log(n)+k)$?

- Better then $k$??
Beyond the Optimal Barrier?

• Examine all of the cells.  
  Marching Cubes.

• Examine only cells containing the isosurface.  
  NOISE / optimal.

• Examine only cells containing the isosurface.  
  \textit{AND} are visible.  
  View dependent.
A View-dependent Approach

- Reduce the amount of data

\[ O(k) \rightarrow \text{Search} \rightarrow \text{construct} \rightarrow \text{Render} \rightarrow O(k) \]

View point

Isovalue
A View-dependent Approach

- Reduce the amount of data
  - Reduce during the search...

Search → construct → Render

View point

O(V(k))  O(V(k))

Isovalue


