Today’s Class

- **Spatial Data Structures Motivation**
- **Spatial Data Structures**
  - Uniform Grid
  - Nested Grid
  - Octree
  - K-d tree
  - Binary Space Partition
  - Bounding Volume Hierarchy
  - Oriented Bounding Boxes
- **Readings for Today**
  - “Farthest Point Seeding for Efficient Placement of Streamlines”
  - “Image Based Flow Visualization”
- **Readings for Friday**
Motivation for Spatial Data Structures

• Closest Point
  – Collision detection
  – Surface normal estimation

• Line-Polygon Intersection
  – Ray casting
    (& recursive ray tracing)
  – Shadow calculation

• Want to do significantly better than the linear $O(n)$, $n = \# $ of objects, brute force solution!

http://i.imgur.com/Xz3Z2iL.jpg

Surface Normal Estimation

http://taylorwang.wordpress.com/
Light Rays in a Dusty Room

Ray Tracing Participating Media
Today’s Class

• Spatial Data Structures Motivation
• Spatial Data Structures
  – Uniform Grid
  – Nested Grid
  – Octree
  – K-d tree
  – Binary Space Partition
  – Bounding Volume Hierarchy
  – Oriented Bounding Boxes
• Readings for Today
  – "Farthest Point Seeding for Efficient Placement of Streamlines”
  – “Image Based Flow Visualization”
• Readings for Friday

Search Pruning via Regular Grid

• Primitives that overlap multiple cells?
• Insert into multiple cells (use pointers)
Ray-Tracing within a Uniform Grid

- Does the current cell contain an intersection?
- Yes: return closest intersection
- No: continue to march along ray

Regular Grid Discussion

- Advantages?
  - easy to construct
  - easy to traverse

- Disadvantages?
  - may be only sparsely filled
  - geometry may still be clumped
Today’s Class

• Spatial Data Structures Motivation
• Spatial Data Structures
  – Uniform Grid
  – Nested Grid
  – Octree
  – K-d tree
  – Binary Space Partition
  – Bounding Volume Hierarchy
  – Oriented Bounding Boxes
• Readings for Today
  – "Farthest Point Seeding for Efficient Placement of Streamlines”
  – “Image Based Flow Visualization”
• Readings for Friday

Adaptive Grids

• Subdivide until each cell contains no more than \( n \) elements, or maximum depth \( d \) is reached
Variations of Adaptive Grids

- **When to split?** When a cell contains “lots” of geometry, but has not yet reached the max tree depth
- **Where to split?**
  - Quadtree/Octree: split every dimension in half, always axis aligned
  - kd-tree: choose *one* dimension (often the largest dimension) and split it axis aligned (but not necessarily at the midpoint)
  - Binary Space Partition (BSP): choose a *arbitrary* cut plane
- **Which one is best?** It depends.... Often they are all equally good!

Primitives in an Adaptive Grid

- Can live at intermediate levels, or be pushed to lowest level of grid
Adaptive Grid Discussion

• Advantages?
  – grid complexity matches geometric density

• Disadvantages?
  – more expensive to traverse (binary tree, lots of pointers)

Today’s Class

• Spatial Data Structures Motivation
• Spatial Data Structures
  – Uniform Grid
  – Nested Grid
  – Octree
  – K-d tree
  – Binary Space Partition
  – Bounding Volume Hierarchy
  – Oriented Bounding Boxes

• Readings for Today
  – "Farthest Point Seeding for Efficient Placement of Streamlines”
  – “Image Based Flow Visualization”

• Readings for Friday
Bounding Volume Hierarchy
• Find bounding box of objects
• Split objects into two groups
• Recurse
Where to split objects?

- At midpoint  OR
- Sort, and put half of the objects on each side  OR
- Use modeling hierarchy

Intersection with BVH

- Check sub-volume with closer intersection first
Bounding Volume Hierarchy Discussion

• Advantages
  – easy to construct
  – easy to traverse
  – binary

• Disadvantages
  – may be difficult to choose a good split for a node
  – poor split may result in minimal spatial pruning

Oriented Bounding Box (OBB):

• generalization of the (axis-aligned) BVH

Today’s Class

- Spatial Data Structures Motivation
- Spatial Data Structures
  - Uniform Grid
  - Nested Grid
  - Octree
  - K-d tree
  - Binary Space Partition
  - Bounding Volume Hierarchy
  - Oriented Bounding Boxes
- Readings for Today
  - “Farthest Point Seeding for Efficient Placement of Streamlines”
  - “Image Based Flow Visualization”
- Readings for Friday

Delaunay Triangulation

- Contains a triangle connecting 3 points if and only if the circumcircle does not contain any other points from the set

http://www.ian-ko.com/resources/triangulated_irregular_network.htm

http://cnx.org/contents/5924d226-2458-446f-b39b-a3dd5868f36c@10/Molecular_Shapes_and_Surfaces#DelaunayTriangulation
Voronoi Diagram/Cells/Regions

- How to re-district the Netherlands into provinces so that everyone reports to the closest capital
- Cell edges are the perpendicular bisectors of nearby points
- 2D or 3D
- Supports efficient Nearest Neighbor queries

http://ccc.inaoep.mx/~rodrigo/robotica/Trigui.pdf

Delaunay / Voronoi Duality


• Start at the furthest away
• Favors long streamlines
• Retain uniformity with increasing density
• 200X Faster, but comparable quality to previous techniques
• Streamline:
  – curve that is everywhere tangent to the vector/flow field,
  – path traced by a massless particle dropped into a steady flow field
  – Stops at boundary or when it gets too close to another streamline
  – Streamlines are better if they uniform, of desired density, and longer rather than shorter
  – Termination points of streamline will be inferred as flow field singularities (source & sink), so avoid misinterpretations...
How to choose seedpoints for streamline placement?

- Uniform grid: streamlines won’t be evenly placed & undesirable patterns
- Randomly placed: Does not improve upon uniform
- Turk & Banks use “streamlets” & energy decreasing optimization to combine, delete, create, lengthen, & shorten streamlets
  - High quality but slow
- Jobard & Lefer seed new streamlines near existing streamlines
  - Faster, but can have empty spaces
- Verma et al. seed streamlines near critical points
  - Good capture of flow features, but poor density control

Figure 2: Local seeding strategy produces empty spaces due to the consecutive stopping of a series of streamlines (bottom right closeup). Some discontinuities also appear near singularities (top right closeup) and in laminar areas (bottom left closeup). Figure reproduced from [12].
• Data structures are important for streamline computation!
• Implemented in C++! Open source (CGAL)!
• Amazing results
  – fast,
  – looks pretty good, but not perfect when zoomed in, those artifacts not discussed in paper
  – Curious about adding color, texture, or animation
  – How could it extend to 3D?
• Detailed writing
  – Great Diagrams
  – Pseudocode: “confident I could implement it”
    “even a data structures student could implement it!”
• Good comparison to prior work
• Motivation could be presented more compellingly
• How is the step size chosen?

• Choose seedpoints that are furthest away from ALL existing streamlines (center of biggest void)
  – Favors long streamlines
  – Amenable to multiresolution placement (each streamline placed increases density)
• Use Delaunay Triangulation
  – Find (approximate?) biggest cavities
  – add point to triangle with largest circumcircle diameter
  – Integrate forward & backward to trace curve through flow (steady state flow only?)
  – Only re-add triangles to priority queue if their circumcircle is bigger than threshold for density & saturation.
• Ok to be somewhat sloppy/inaccurate about the Delaunay triangulation
  – when streamlines form tight curls
  – Use local maximum circumcircle
• Advanced Features
  – Variable density, Multiresolution
Today’s Class

• Spatial Data Structures Motivation
• Spatial Data Structures
  – Uniform Grid
  – Nested Grid
  – Octree
  – K-d tree
  – Binary Space Partition
  – Bounding Volume Hierarchy
  – Oriented Bounding Boxes
• Readings for Today
  – “Farthest Point Seeding for Efficient Placement of Streamlines”
  – “Image Based Flow Visualization”
• Readings for Friday

• "Image Based Flow Visualization", Jarke J. van Wij, SIGGRAPH 2002.
• Advection: “the transfer of heat or matter by the flow of a fluid, especially horizontally in the atmosphere or the sea”
• Blend warped previous frame w/ a selected background image
• Applications: Weather, climate, industrial processes, cooling, heating
• Moving particles, streamlines, moving textures, topological images
• Contributions:
  – Handles unsteady flow
    • W/ other techniques, hard to reconstruct flow
  – Efficient
    • Best performance to date (50 fps) by using graphics card features
  – Easy to implement
    • Other techniques need user to place particles (with poor placement, important features can be missed)
    • Small amount of code!

• White noise
• Pink noise: remove high frequency from white noise, both in space & time

• Pathline: position of a particle in a dynamic flow field
• Streamline: same as pathline, but for single point in time (or a constant/steady flow field)

• Particle & streamline tracking: focus on world space coordinates
• Line integral convolution: focus on screen space coordinates
• Image Based Flow Visualization: focus on images as basic primitive

• Data structures are essentially irrelevant for image based flow visualization!
• Implemented in Object Pascal (??)
• Frequency analysis (yeah “Signals & Systems”!) to understand & create background images to minimize artifacts from undersampling
• If G changes over time, then texture will “move with the flow”
  – not new random image, but spots that appear & disappear (cosine, square, exponential decay, sawtooth)
• Computation
  – Distorted mesh calculated on CPU
  – Rendering & blending on GPU
• Lots of control over results
  – Background image choice
  – Dye injection
  – Alpha values choice (decay)

• Good attempt to compare running time of different algorithms from different eras on different hardware
• Great discussion/comparison of results
• Method was unintuitive (author should take that as a compliment, right?) & really clever
  – “Work smarter not harder”
• How does resolution affect quality?
• What is coarse vs fine texture?
• Produces a visual result, but can’t be used other than to “look at” (results can’t be fed back into simulation/computation)
• Somewhat confused about “dye” and “decay of dye”
• What is the motivation for this paper?
• Use of gradient/color to help reader understand direction of flow is very important
Today’s Class

- Spatial Data Structures Motivation
- Spatial Data Structures
  - Uniform Grid
  - Nested Grid
  - Octree
  - K-d tree
  - Binary Space Partition
  - Bounding Volume Hierarchy
  - Oriented Bounding Boxes
- Readings for Today
  - “Farthest Point Seeding for Efficient Placement of Streamlines”
  - “Image Based Flow Visualization”

- Readings for Friday
Volume Rendering

• Try to display a 3D data set (the whole thing, not just a surface)
• Must determine the opacity of every pixel (voxel)
• Often called the “transfer function”
• Almost always medical images

Readings for Friday: *(pick one)*

• “A survey of algorithms for volume visualization”, T. Todd Elvins, 1992
Readings for Friday: *(pick one)*

- “Hardware-Accelerated Volume Rendering”, Pfister et al., *from the Visualization Handbook* 2004

![CT Angiography of a human brain (512^2 x 128). Transparent rendering of a non-polygonal shaded iso-surface with 2D multi-textures on an NVIDIA GeForce-4 Ti. Image courtesy of Christof Reck-Salama, University of Erlangen, Germany.](image嚓)

Readings for Friday: *(pick one)*


![Intuitive Exploration of Volumetric Data Using Dynamic Galleries](image嚓)
Readings for Friday: *(pick one)*

- “Anisotropic Ambient Volume Shading”
  Ament & Dachsbacher, IEEE Visualization 2015