Visualization Tools for Adaptive Mesh Refinement Data

Gunther H. Weber[1], Vincent E. Beckner[1], Hank Childs[2], Terry J. Ligocki[1], Mark C. Miller[2], Brian Van Straalen[1] and E. Wes Bethel[1]

[1]Lawrence Berkeley National Laboratory
[2]Lawrence Livermore National Laboratory

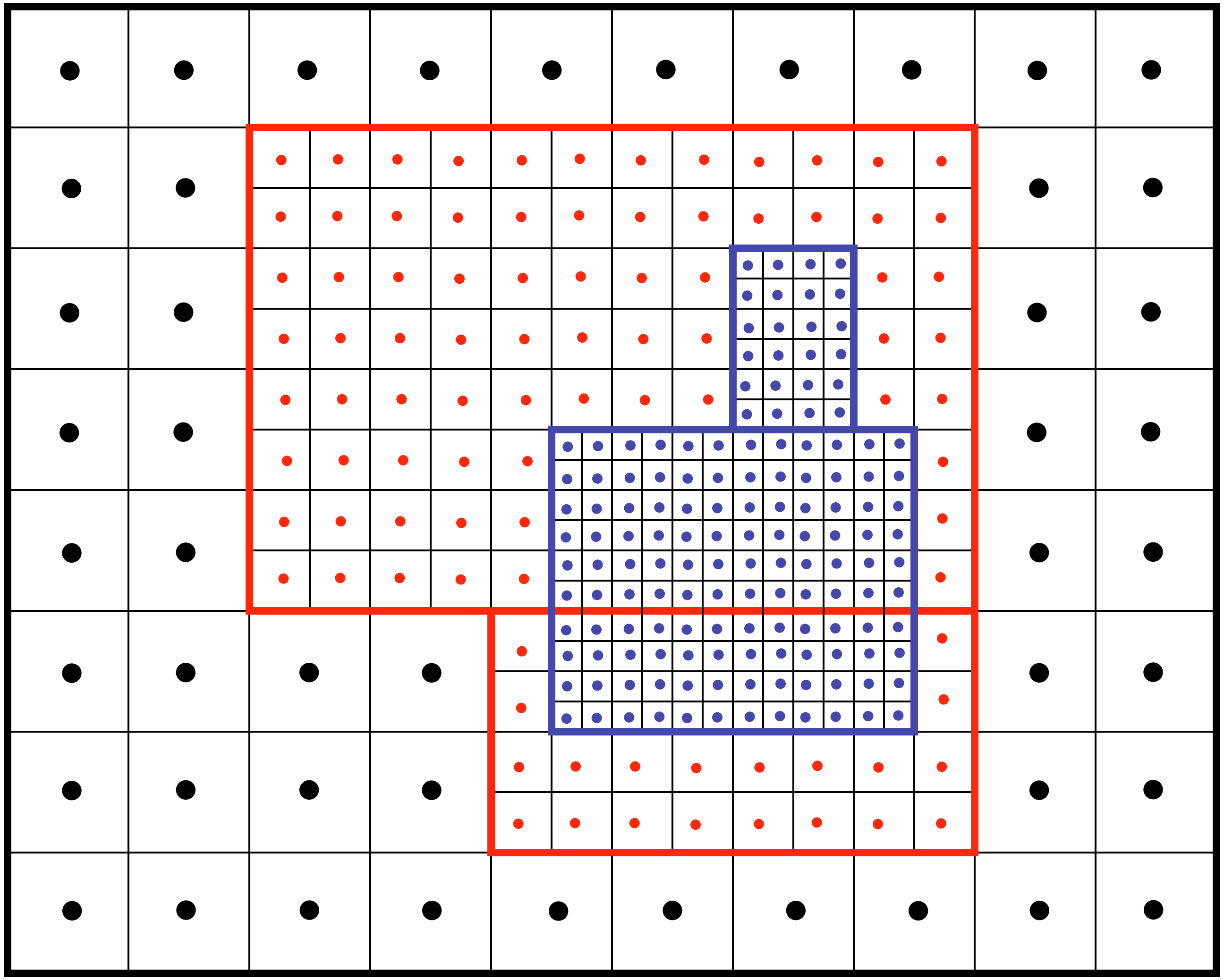
Outline

- Introduction to Berger-Colella AMR
- Visualization of Scalar AMR Data
- Specialized AMR Visualization Tools
- Visualization Tools with AMR Support
- Short overview of VisIt

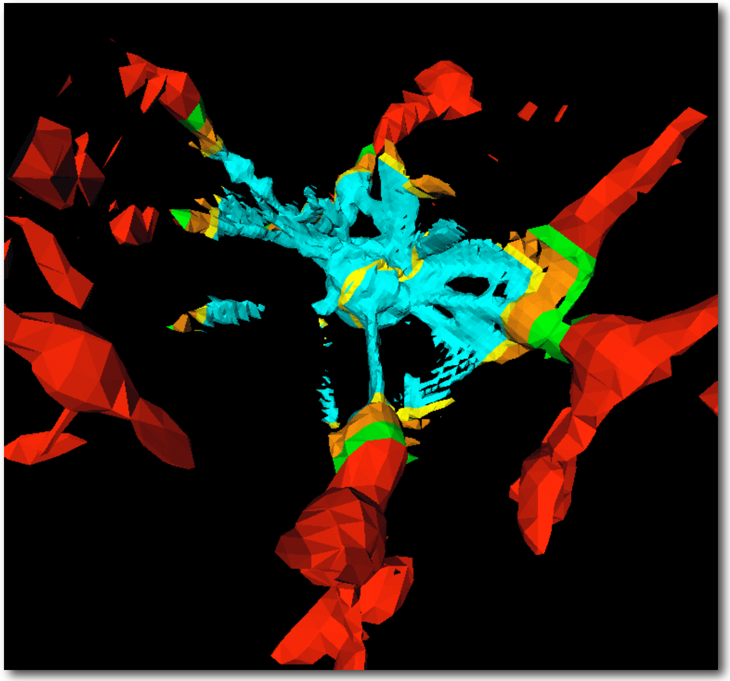
Adaptive Mesh Refinement

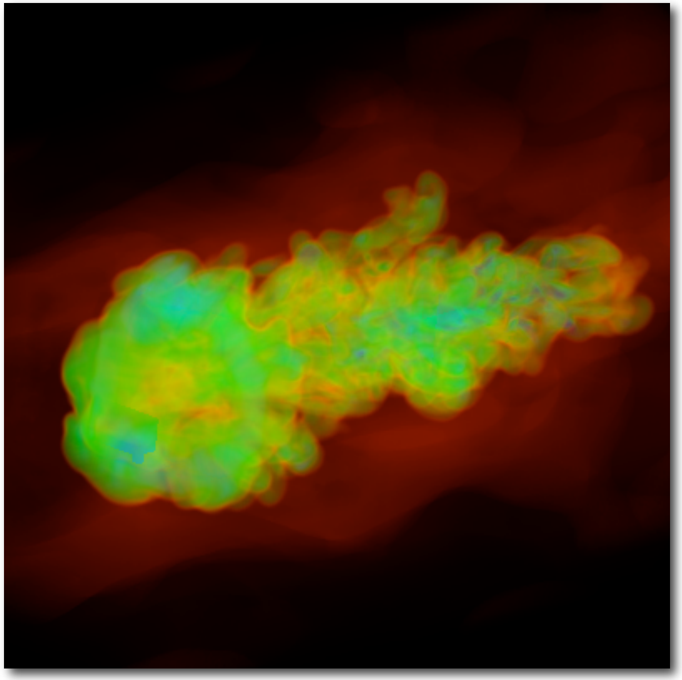
- Computational fluid dynamics technique
- Topological simplicity of regular grids
- Adaptivity of unstructured meshes
- Nested rectilinear patches, increasing resolution
  - Reduce simulation time
  - Reduce storage space
- Berger-Colella AMR: axis-aligned patches
- Very often: Cell centered data

Berger-Colella AMR Format



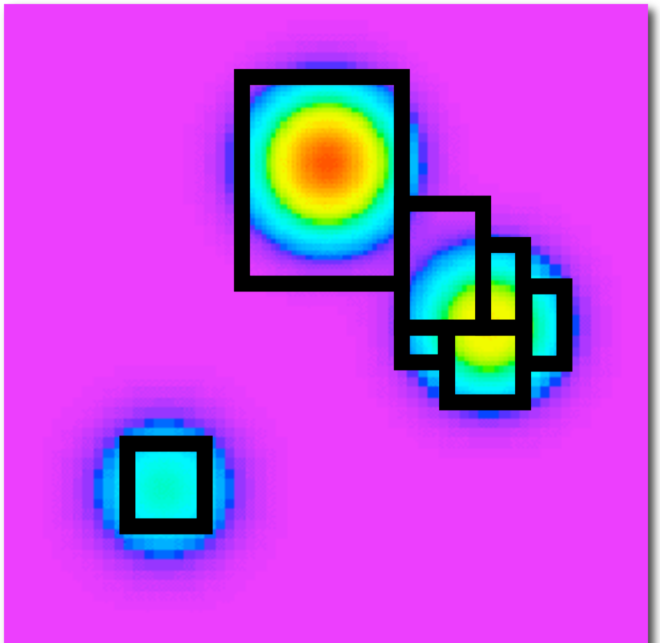
Scalar Field Visualization





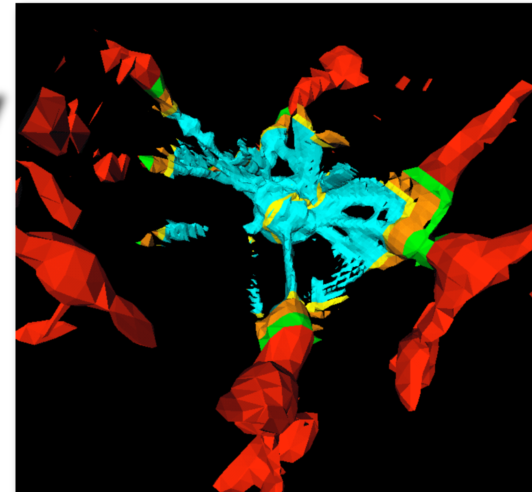
- Isosurface Extraction
  - Main challenge: Consistent “crack-free” surfaces
- Direct Volume Rendering
  - Effective utilization of hierarchy for efficient rendering

Effective Visualization of Scalar AMR Data



Hierarchical AMR simulation

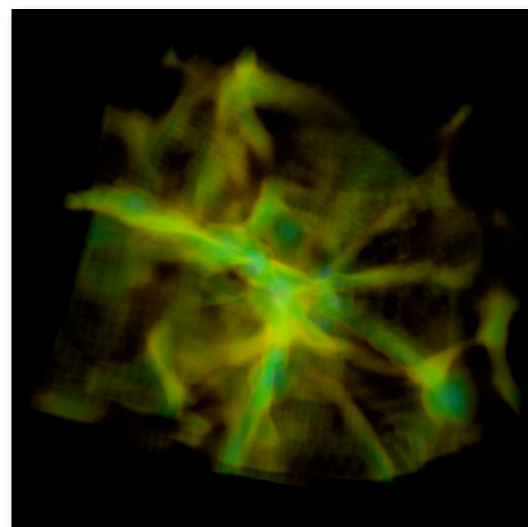
Visualization



Isosurfaces

- Extraction of continuous “crack-free” isosurfaces

Visualization



“Direct Volume Rendering”

- Effective utilization of the hierarchy for efficient rendering
- Good interpolation functions

Aim: Use inherently hierarchical structure for efficient visualization

AMR Visualization – In the Beginning

- Translation of AMR to unstructured meshes [Norman et al. 1999]
  - Visualization with standard tool (VTK, IDL, AVS)
  - Ineffective utilization of computational resources
- Direct Volume Rendering
  - Mention AMR data without further details [Max 1993]
  - PARAMESH [Ma 1999]
    - Resampling
    - Block-based

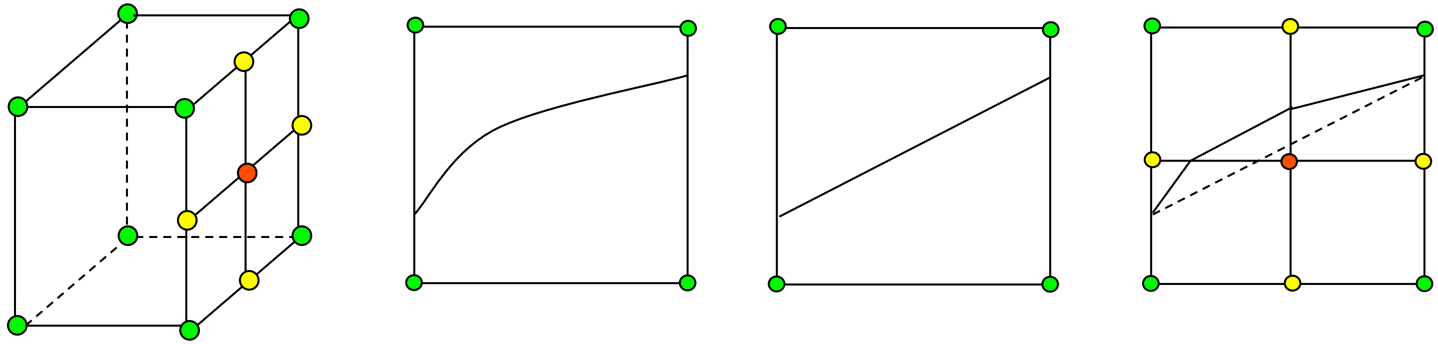
Isosurfaces

Marching Cubes and Dangling Nodes

- Marching cubes needs vertex centered data
  - *Resample* data set *to vertex centered* case

➨ Dangling nodes := only present in fine level (yellow + red)

  - Choice of consistent values to avoid problems?

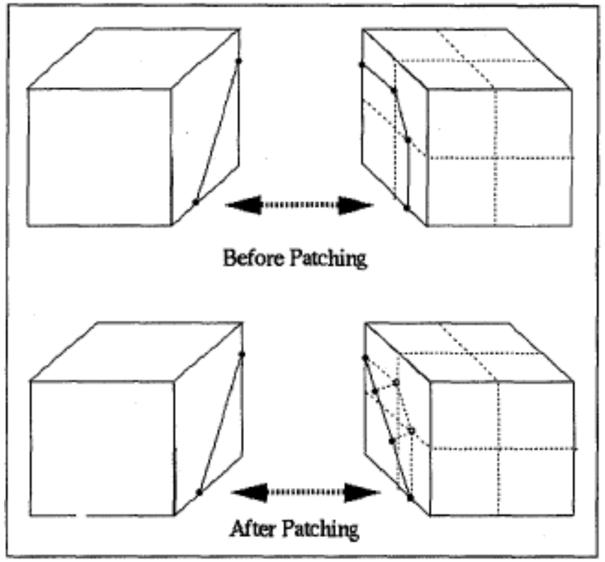


- Same in coarse and fine grid
- Linear interpolation avoids problems
- No unique value avoids problems

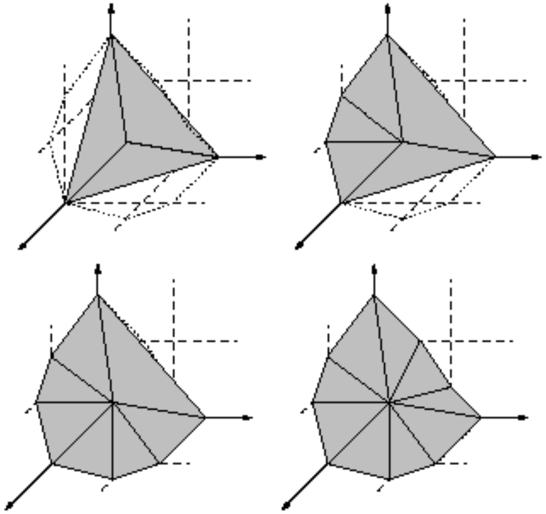
➨ Compare [Westermann, Kobbelt, Ertl 1999]

Previous Crack-fixing Solutions

- Mostly in context of Octree-based hierarchies
- [Shu et al., 1995]: Create polygon to fit crack
- [Shekhar et al., 1996]: Collapse polyline to line
- [Westermann et al., 1999]: Create triangle fan



[Shekar et al., 1996]



[Westermann et al., 1999]

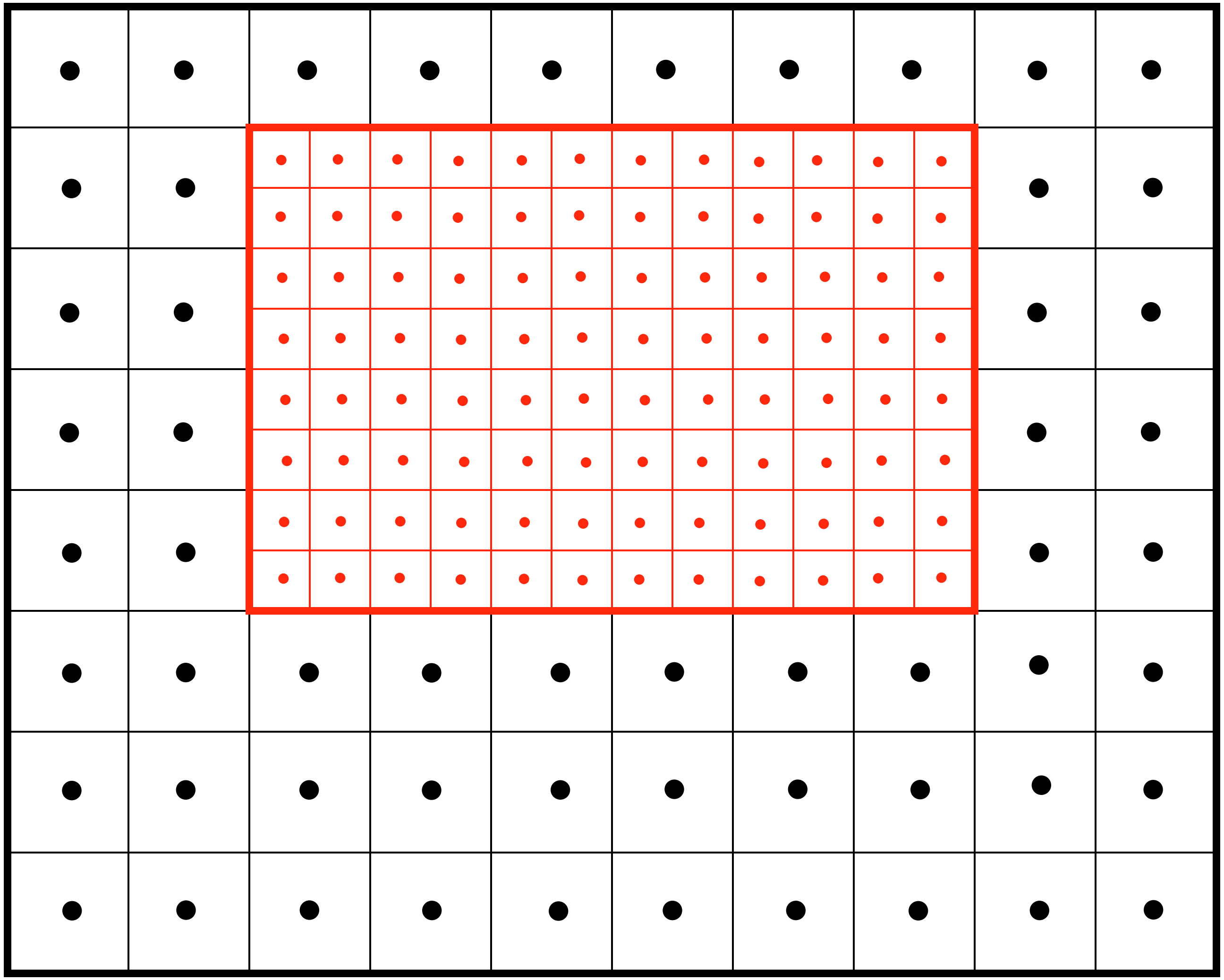
First Approach: Use of Dual Grids

- *"Avoid interpolation whenever possible!"*

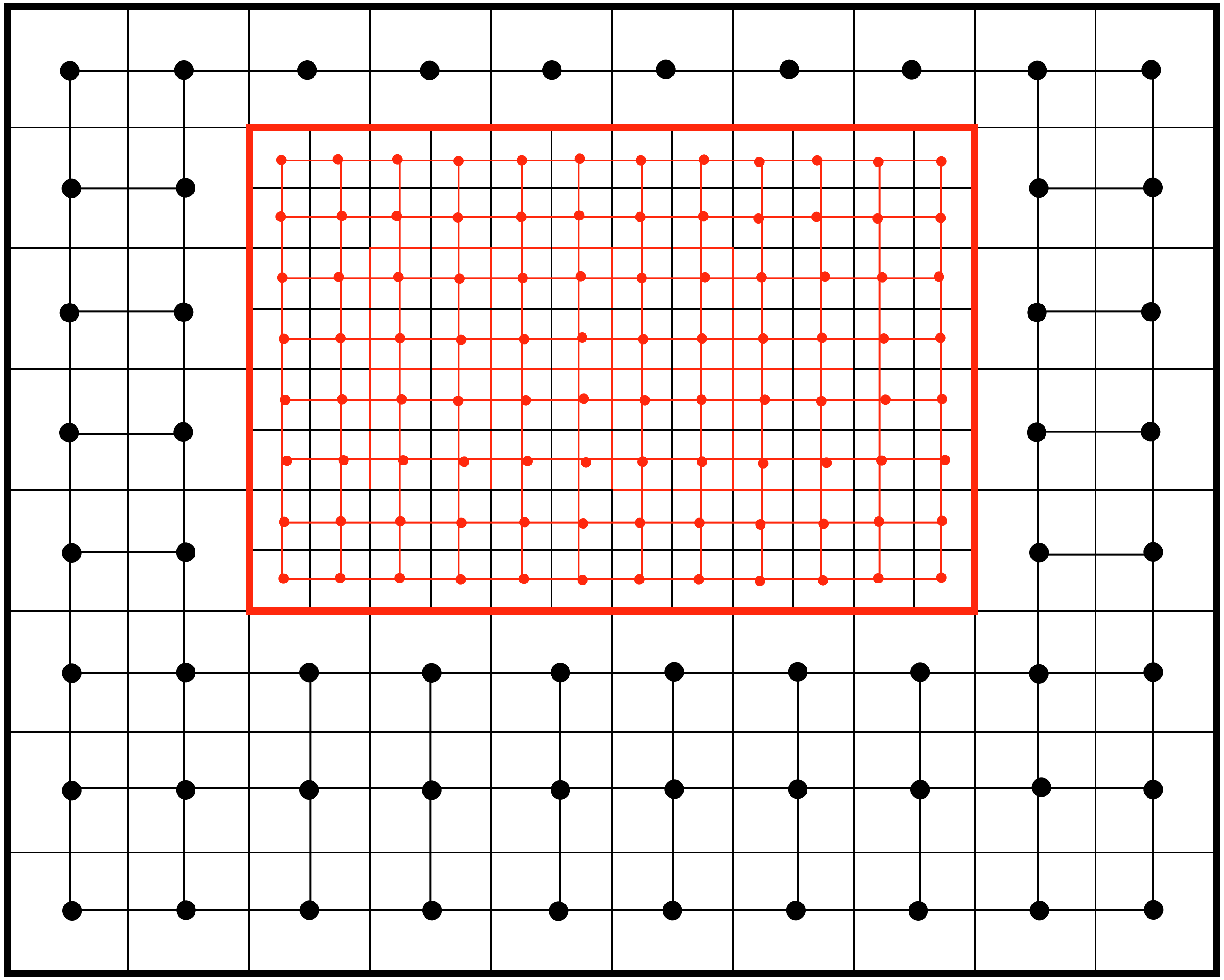
➡ Avoid interpolation apart from linear interpolation along edges, which is part of marching cubes

- Use *dual grid* := grid formed by connecting cell centers

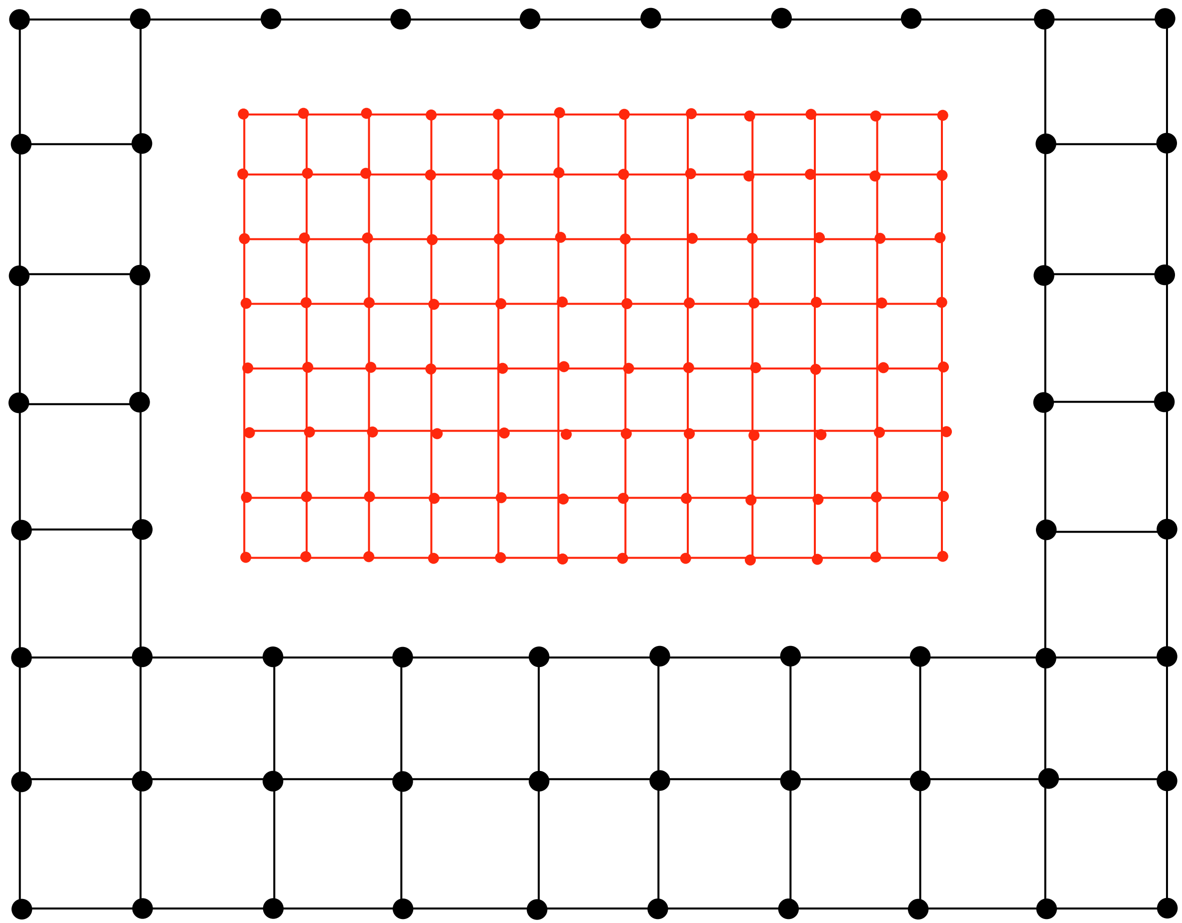
Dual Grid – Original Grid



Dual Grids – Both Grids



Dual Grids



Advantages of Dual Grid Approach

- Use of *values original data* for *marching cubes*
- No dangling nodes
- Instead: *Gaps* between hierarchy levels!

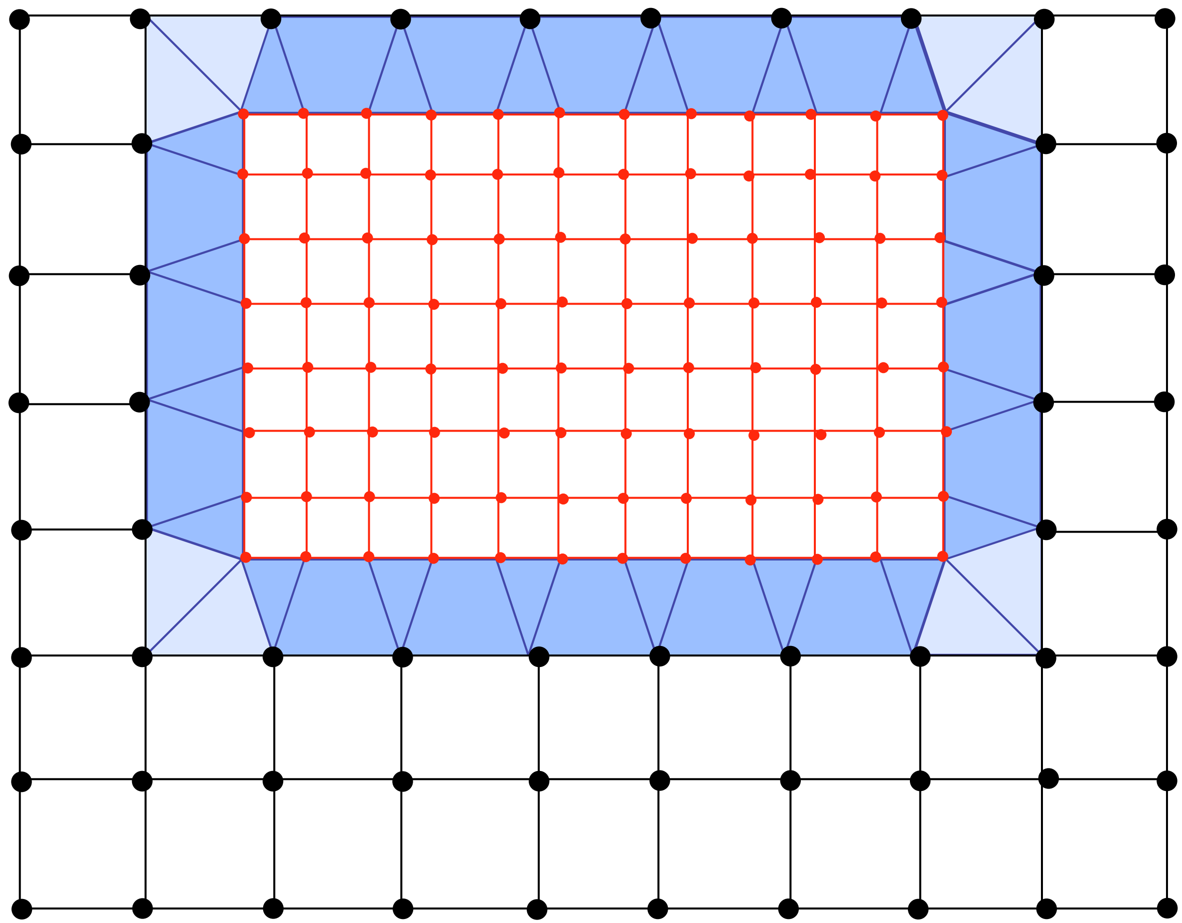
➡ Fill those gaps with *stitch cells*

Stitching the Gaps

- Tessellation scheme for filling the gap between two hierarchy levels
- Constraints
  - Only gap region is tessellated
  - The complete gap region is tessellated
  - Only vertices, edges and *complete* faces are shared

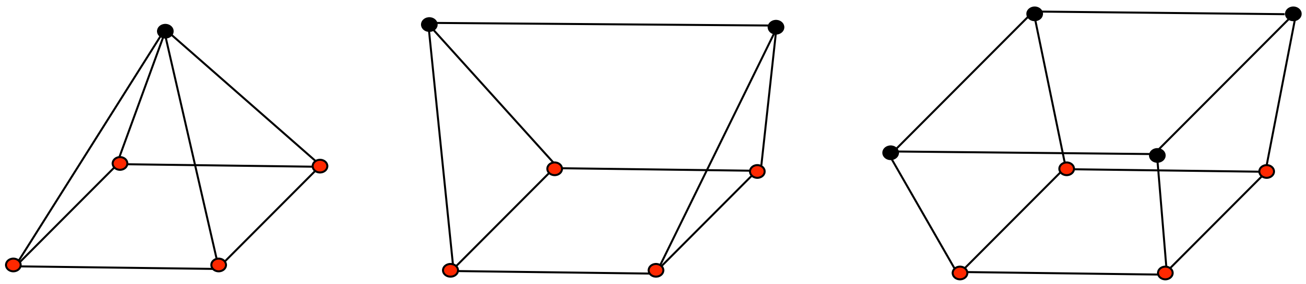
➨ In 3D space: Cannot use tetrahedra because cells must share quadrilaterals as faces

Stitching Process

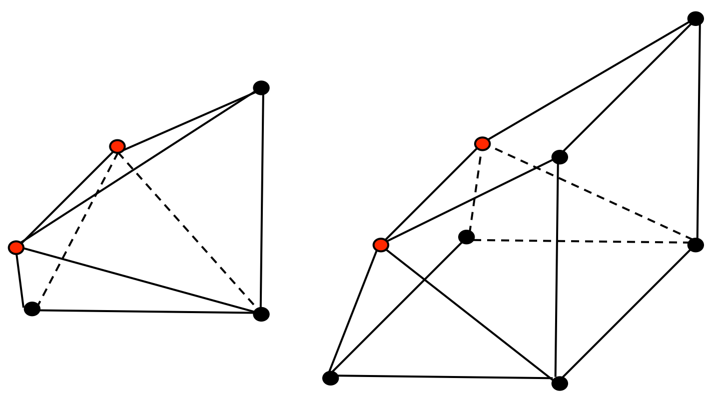


Stitch Cells – 3D Case

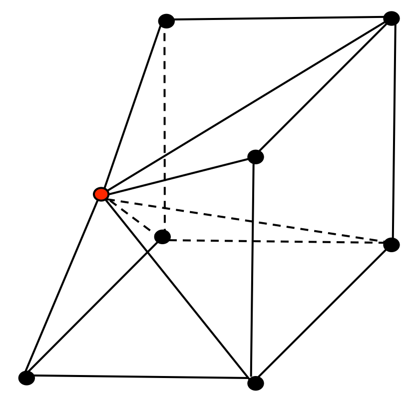
- Fine patch
- Coarse patch



Cell Faces

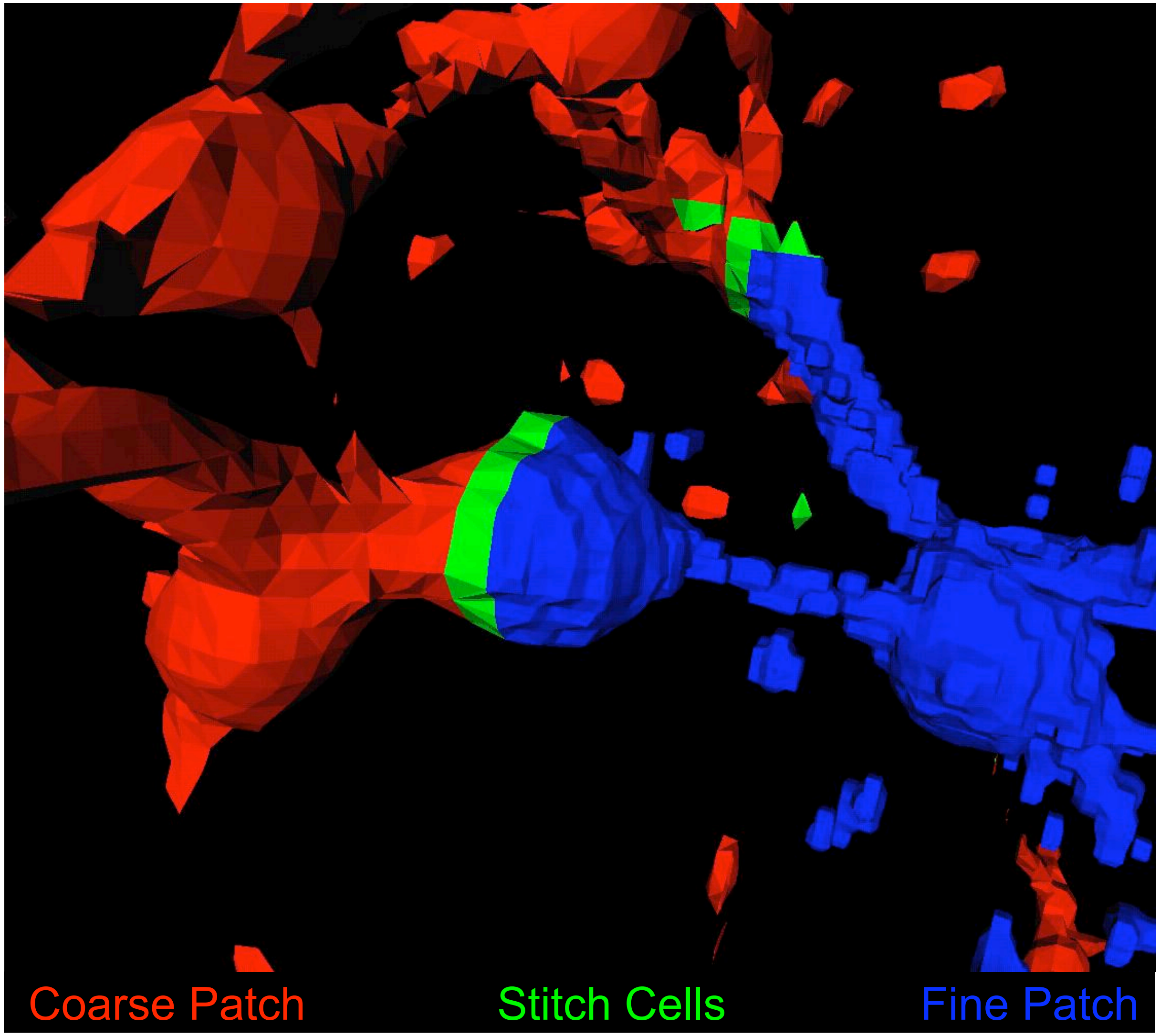


Cell Edges



Cell Vertices

First Results



AMR simulation of star cluster formation

Root level 32x32x32

[Data set: Greg Bryan, Theoretical Astronomy Group, MIT]

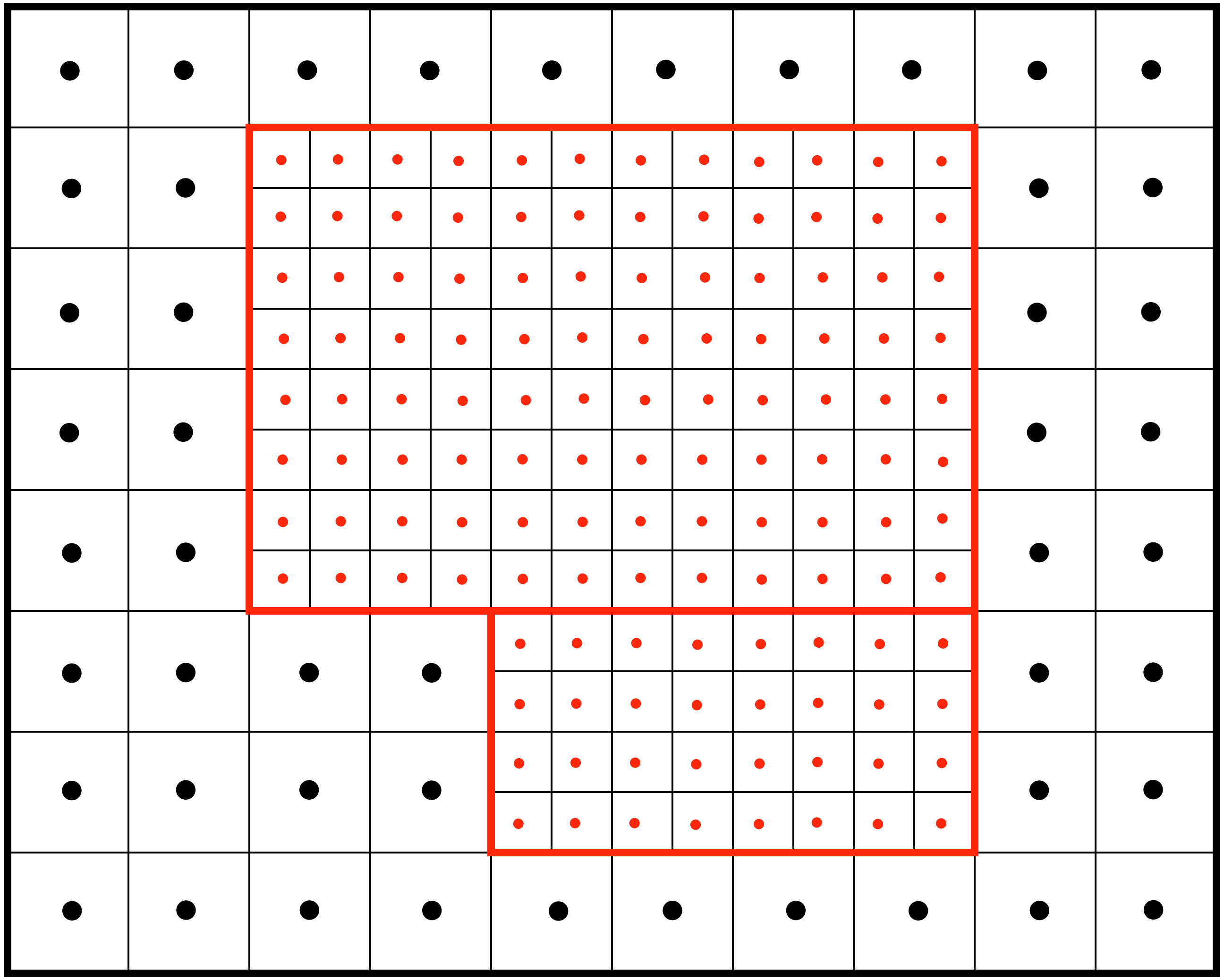
Multiple Patches

- Multiple patches can be connected using the same scheme

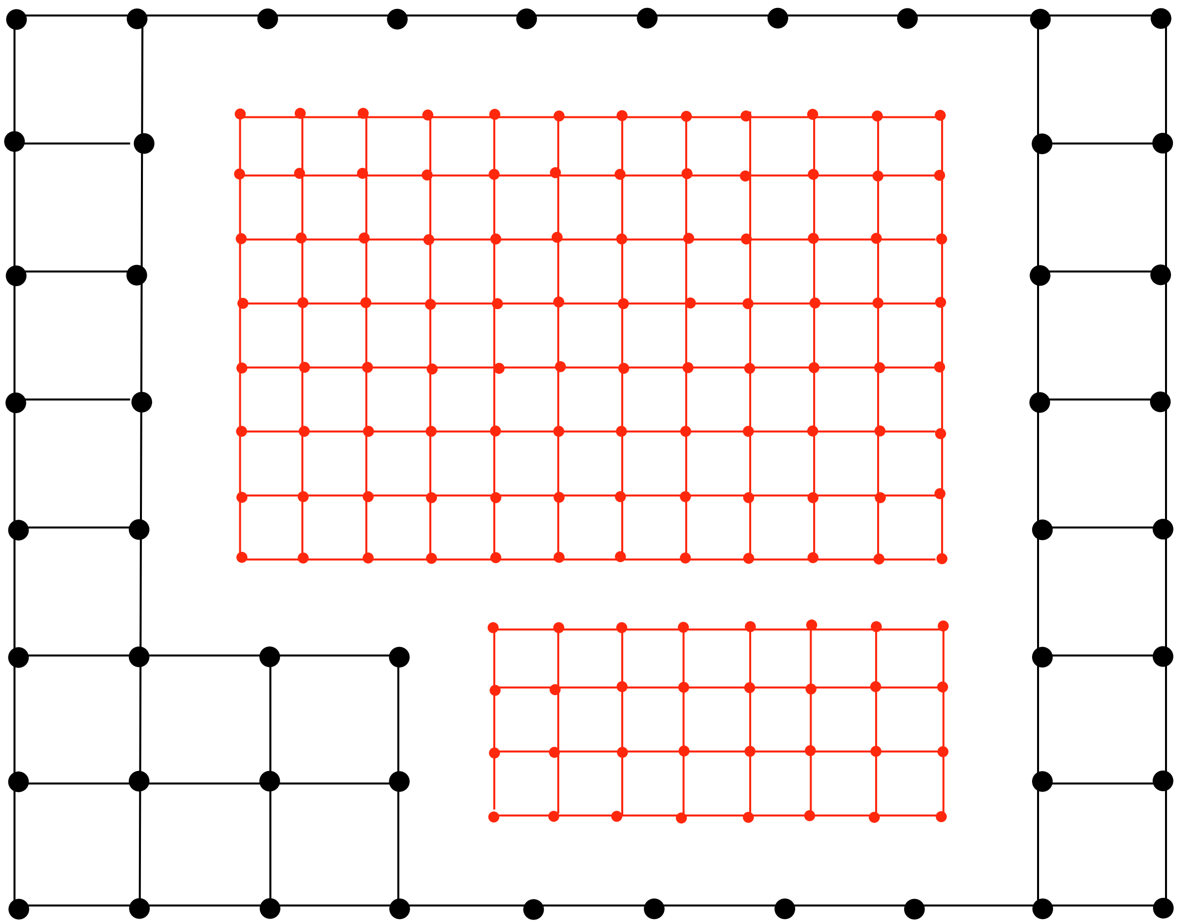
- However: Special care must be taken with adjacent fine patches.

- Must “merge” adjacent grids (i. e., “upgrade” edges to quadrilaterals and vertices to edges)

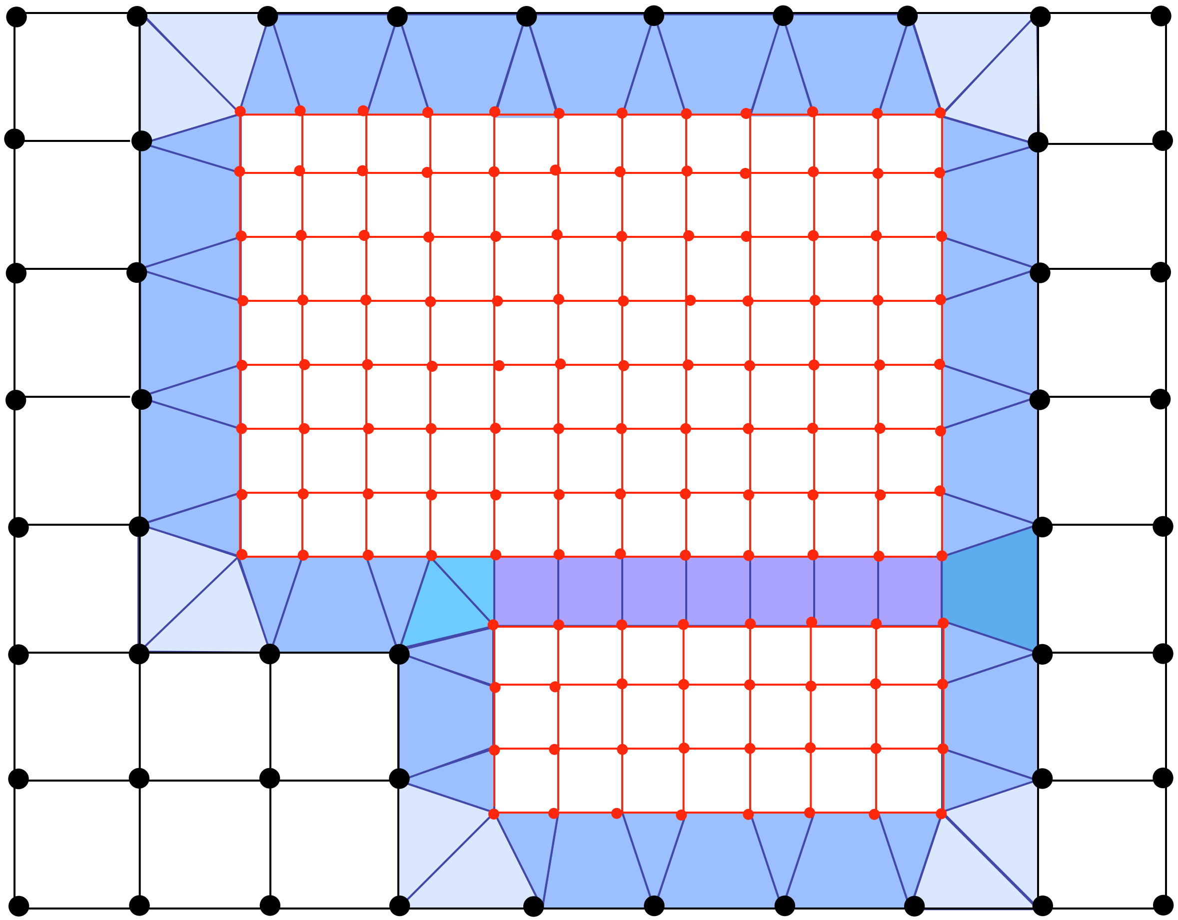
Multiple Patches – Example



Multiple Patches – Example



Multiple Patches – Example



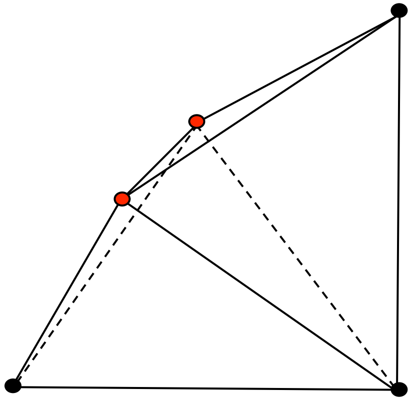
Multiple Patches – Cell Faces

- **Pyramid (2 basis configurations):**
  - **Unrefined coarse grid point → No change**
  - **Refined coarse grid point → Becomes cuboid**

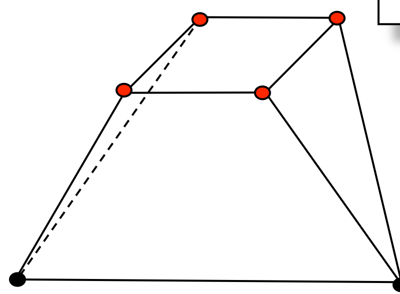
- **Triangle prism (3 basis configurations):**
  - **All coarse grid points unrefined → No change**
  - **One refined coarse grid point**
  - **Both coarse grid points refined → Becomes cuboid**

Multiple Patches – Fine Edge to Coarse Edges

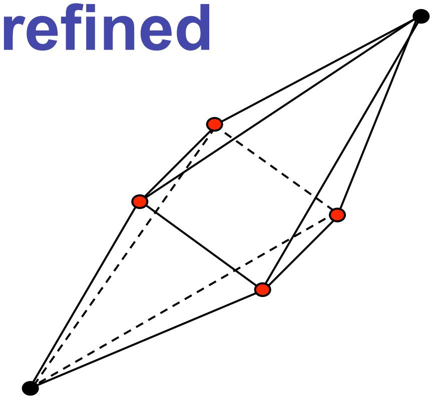
- Fine patch
- Coarse patch



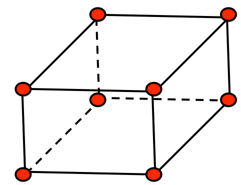
All coarse grid points unrefined



Two neighboring coarse grid points refined



Two diagonally opposed coarse grid points refined



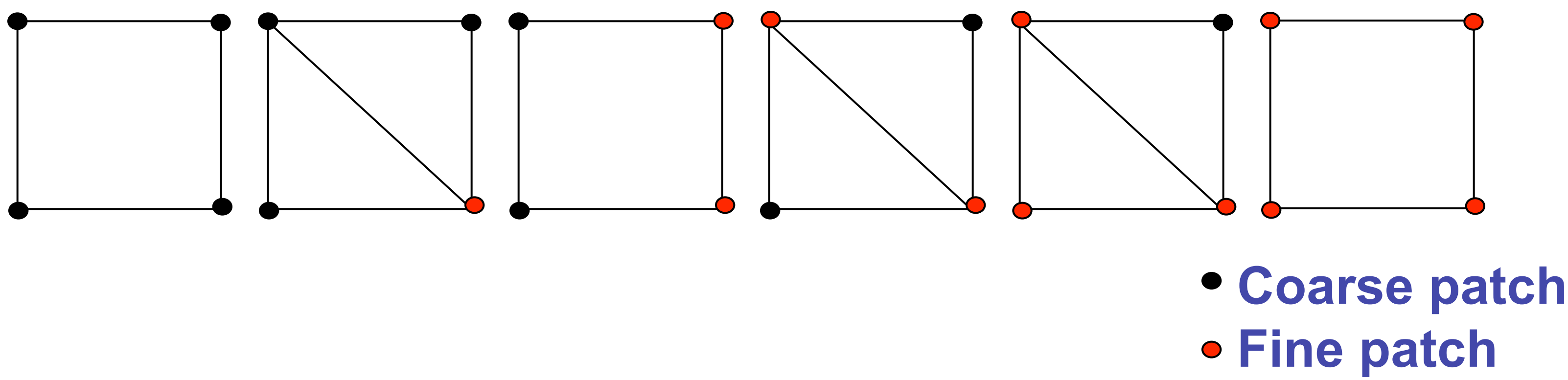
All coarse grid points refined

Multiple Patches (3D) – Remaining Cases

- All remaining cases consider 8 vertices
  - ➡ Quadrilateral Cell
- Actual vertex positions irrelevant!
- Information per vertex: refined or unrefined?

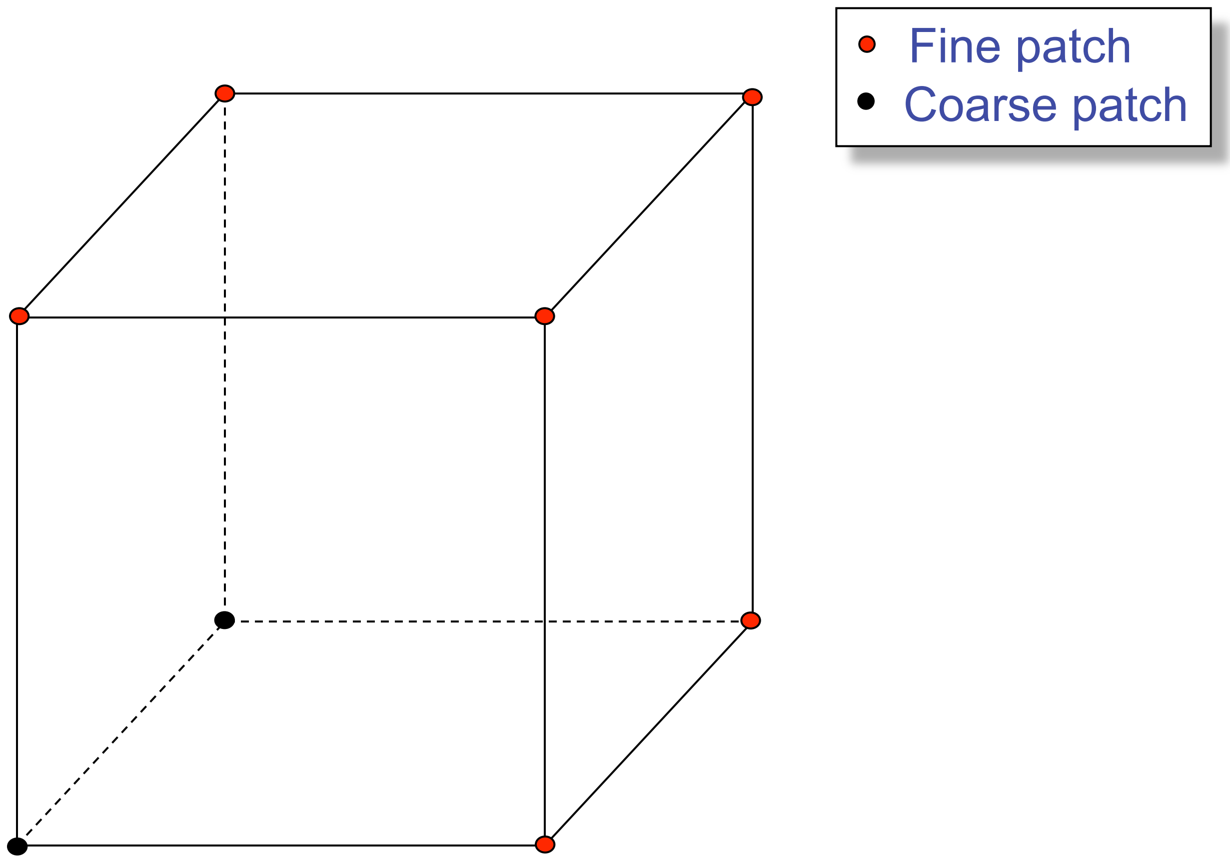
Multiple Patches (3D) – Generating Tessellations

- Draw cell to tessellate as a cube
- Mark each vertex as refined or unrefined
- Use canonical subdivisions for boundary faces

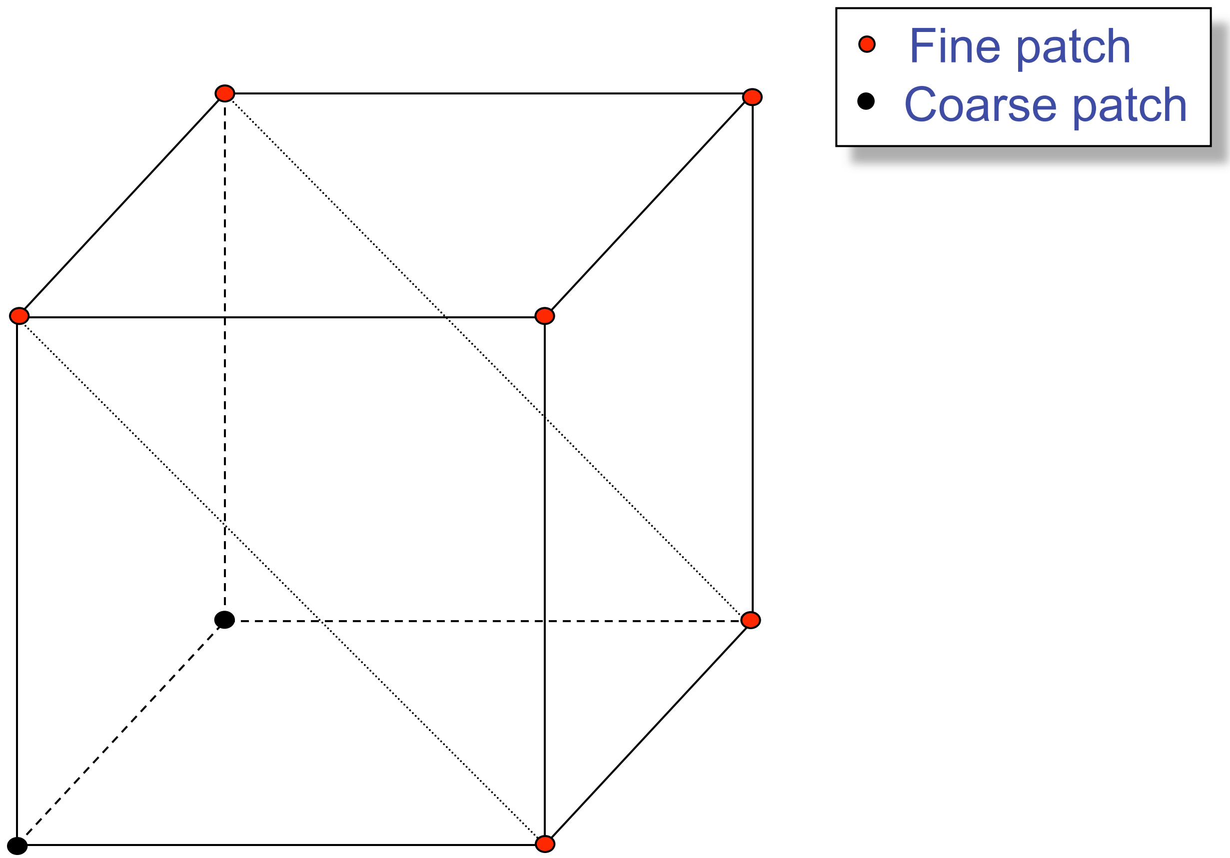


- Use implied tessellation for cell
  - If more than one tessellation is possible, use arbitrary one

Multiple Patches – Example Tessellation



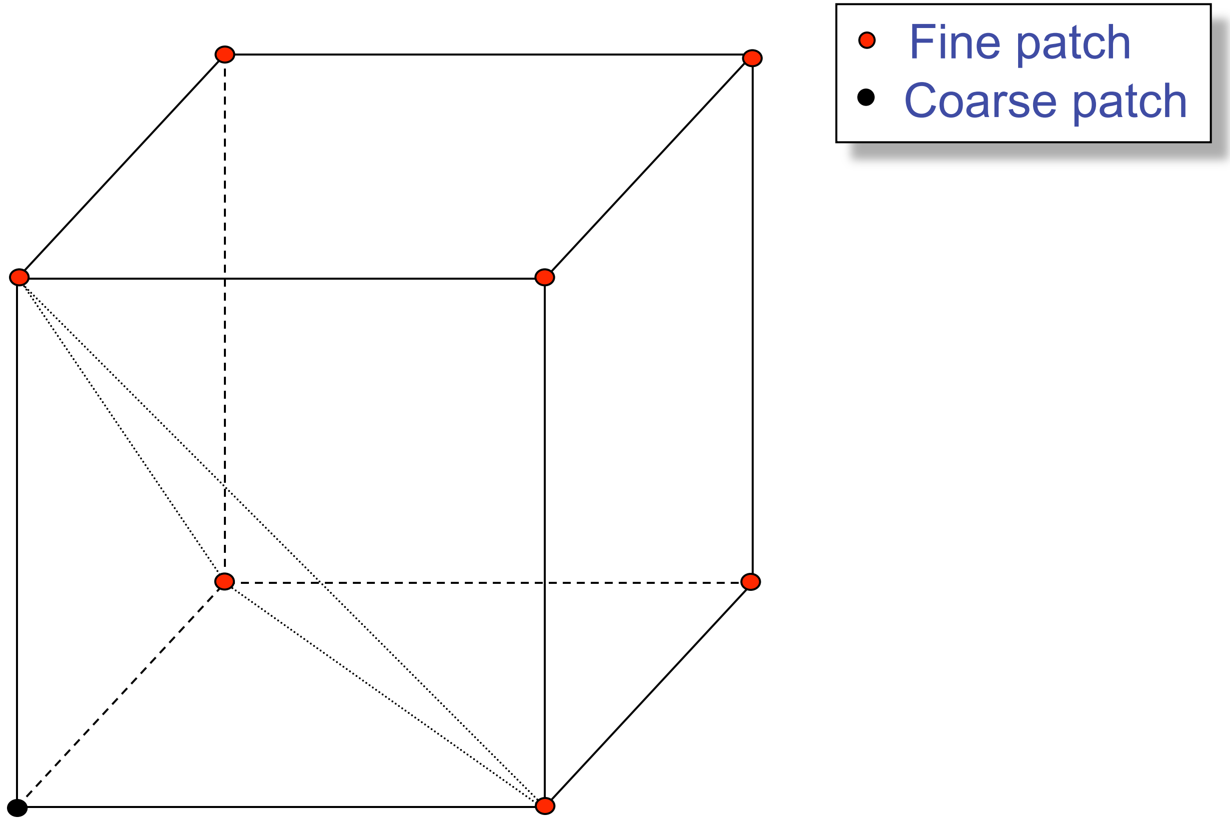
Multiple Patches – Example Tessellation



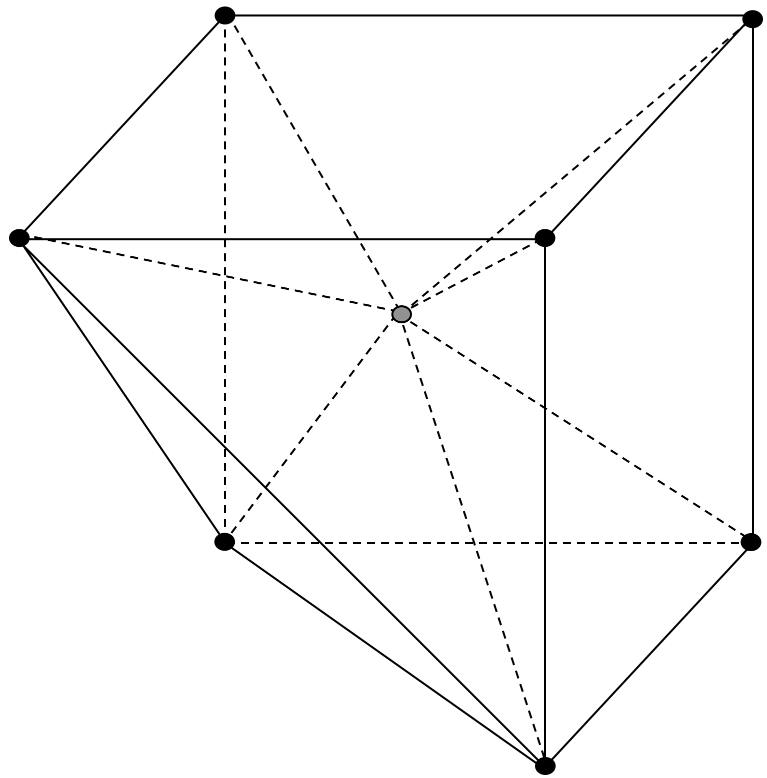
Reducing Amount of Cases

- Quadrilateral to quadrilateral (16 cases)
  - No reduction necessary
- Edge to quadrilaterals (64 cases)
  - Upgrade to quadrilateral case (-24 cases)
  - In certain cases: Can consider two independent triangular prisms (- 14 cases)
  - ➨ 26 cases (- further symmetry considerations)
- Vertex to Quadrilaterals
  - Either upgrade to edge case or consider three pyramids independently

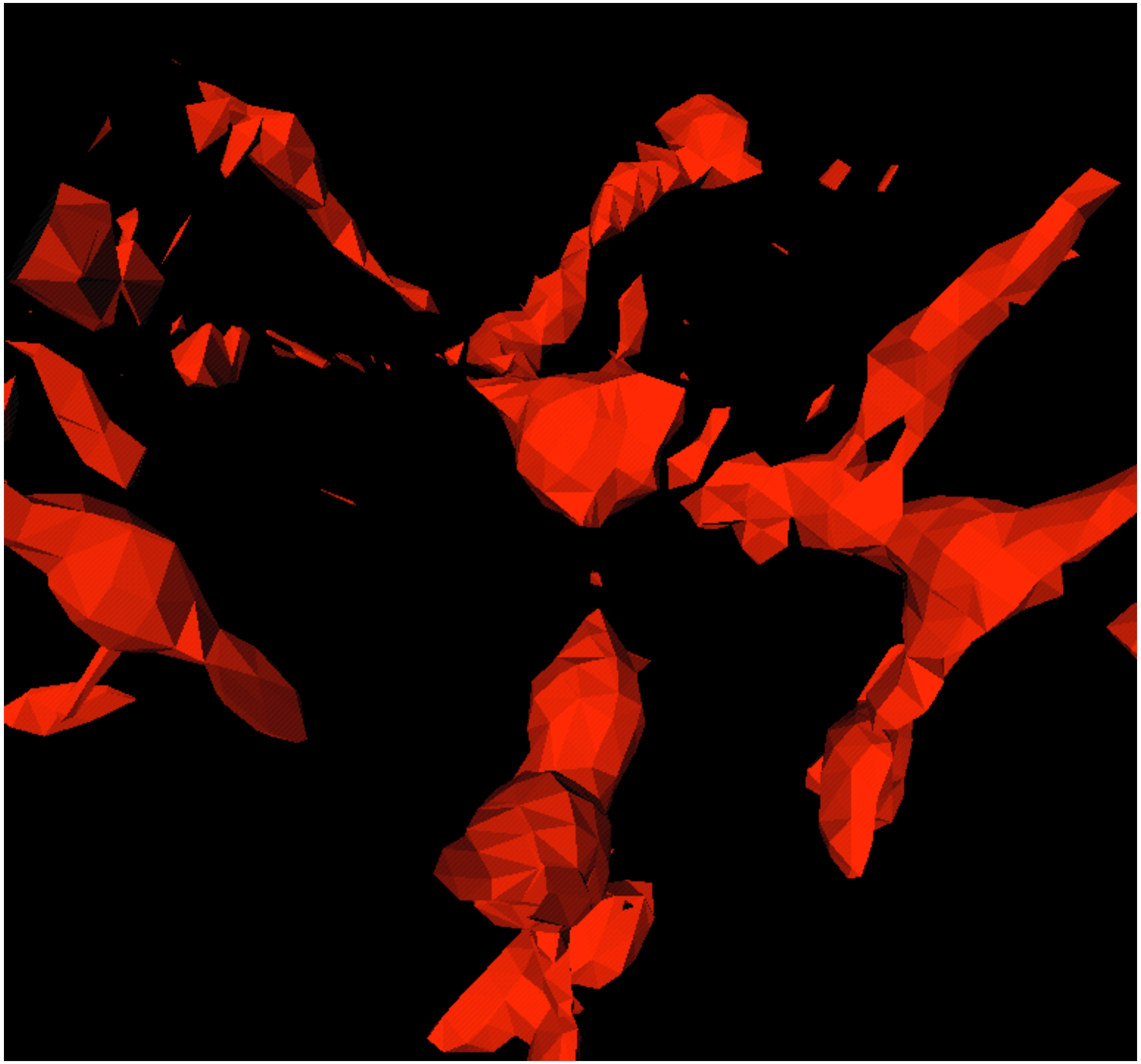
Problem Case



Problem Case



Isosurface - One Level

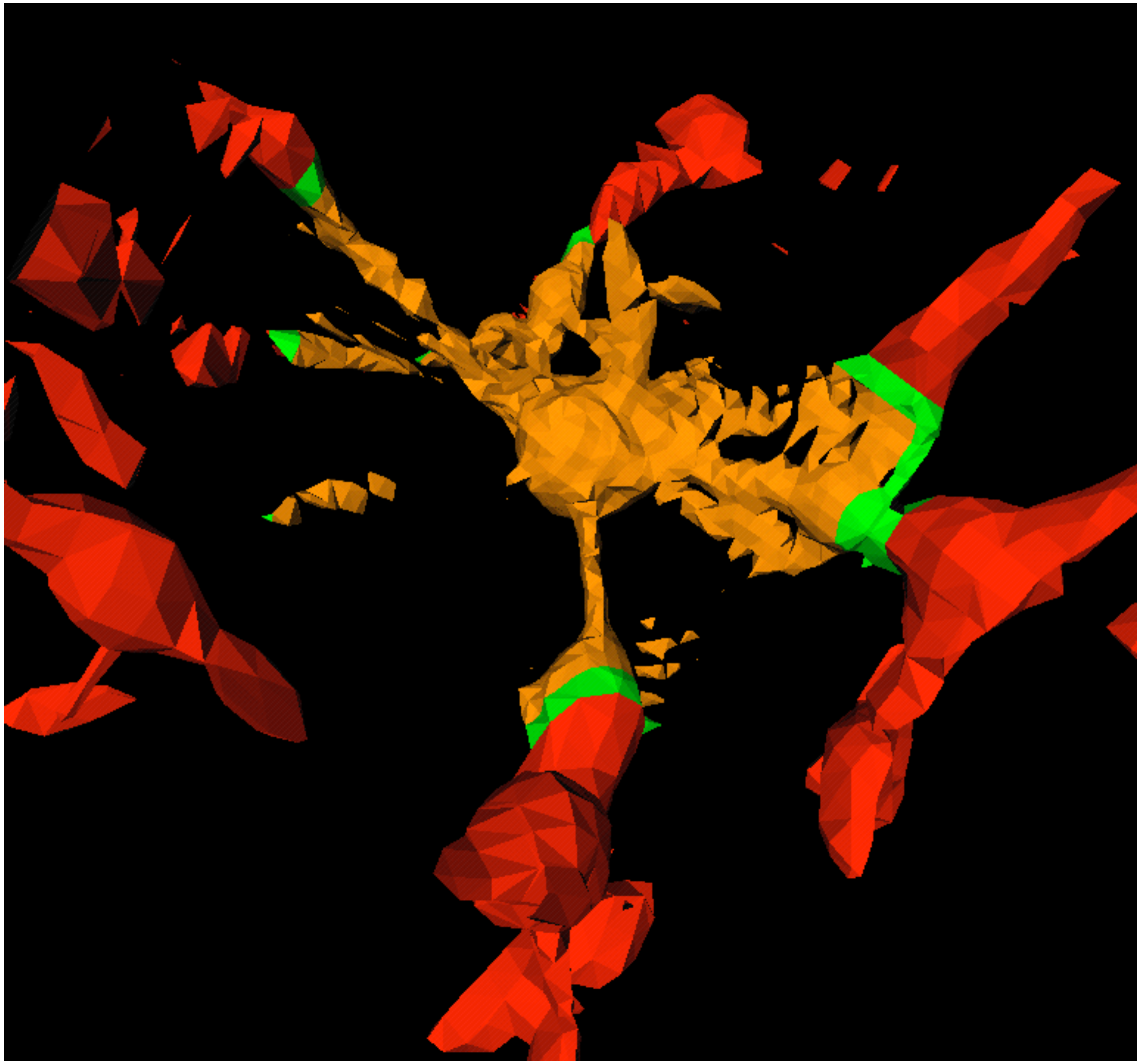


AMR simulation of star cluster formation

Root level 32x32x32

[Data set: Greg Bryan, Theoretical Astronomy Group, MIT]

Isosurface - Two Levels



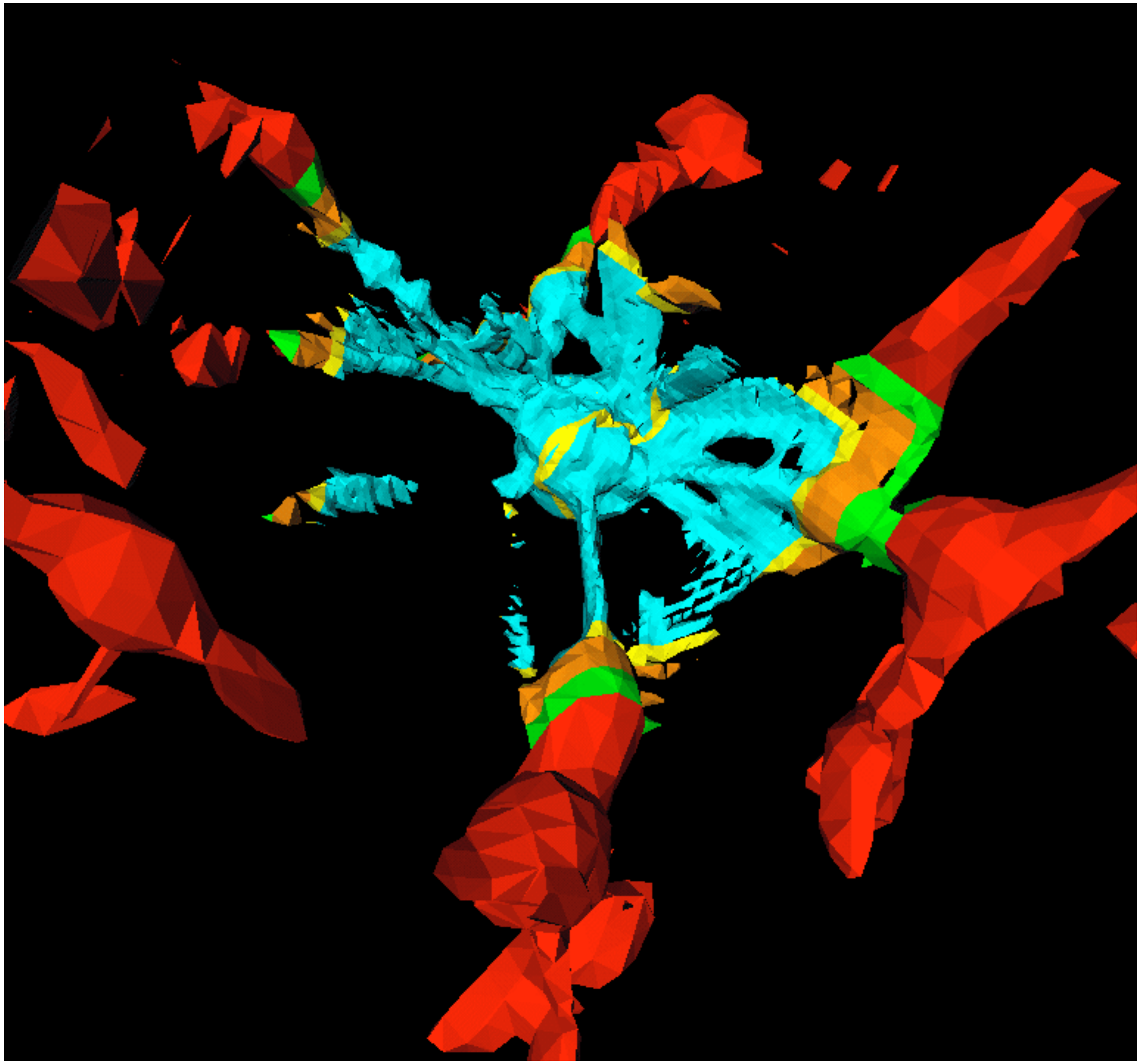
AMR simulation of star cluster formation

First level

Stitch cells (1/2)

Second level

Isosurface - Three Levels



AMR simulation of star cluster formation

First level

Stitch cells (1/2)

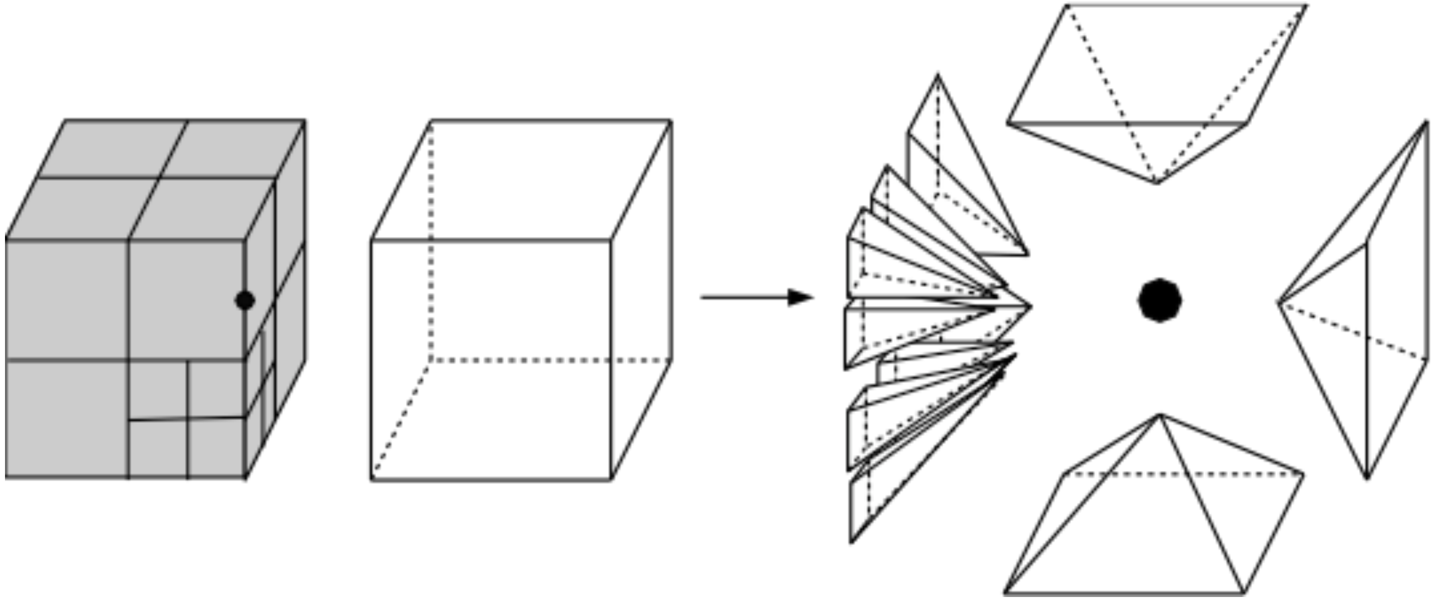
Second level

Stitch cells (2/3)

Third level

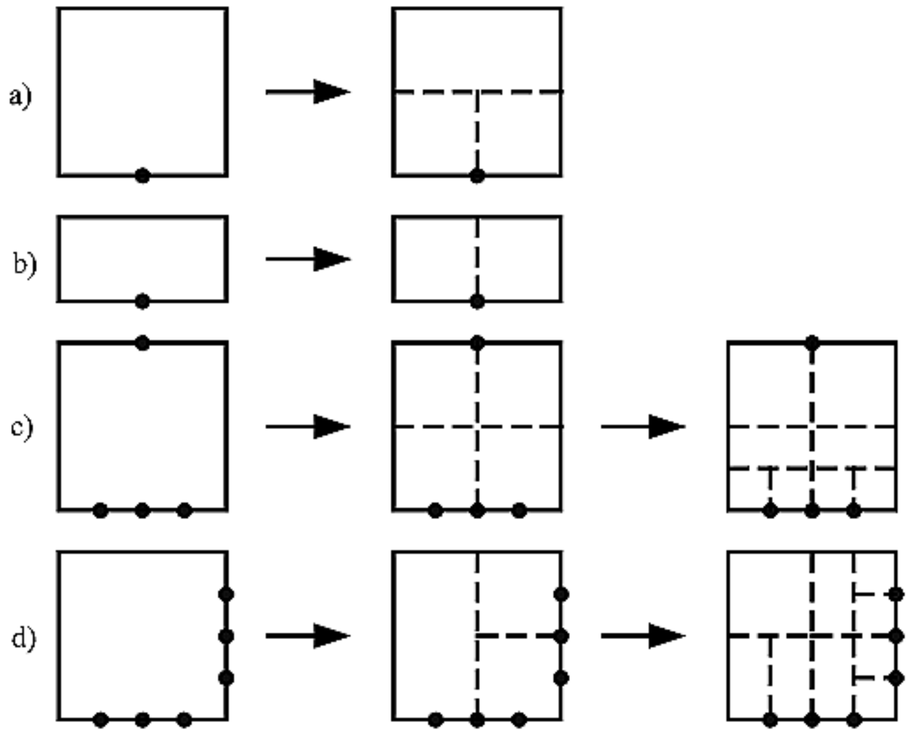
Second Approach: Keep Grid

- Vertex/node centered data
- Retain "identity" of cells (debugging)
- Subdivide boundary cells into pyramids
  - Eliminates "non-linear" hanging nodes
  - Standard isosurface techniques for pyramids

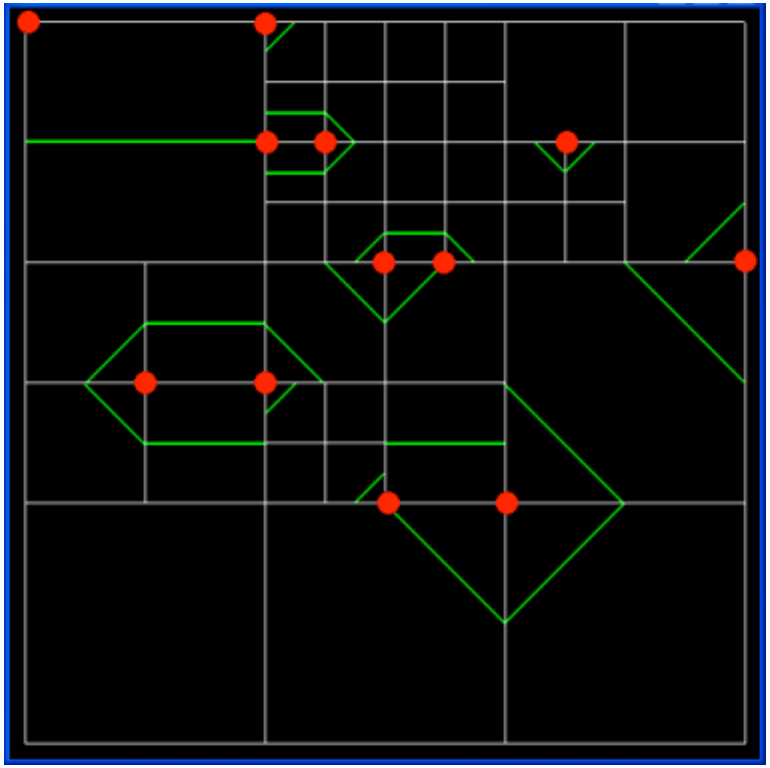


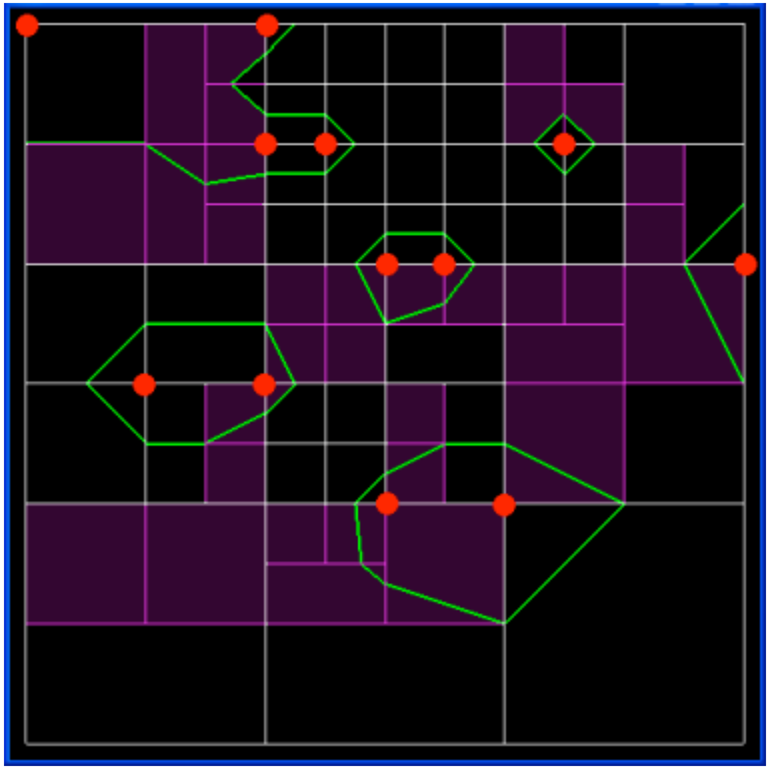
2D Case

- Forms basis of 3D case
- Split cell faces to eliminate hanging nodes along edges
- Obtain values at newly created hanging by linear interpolation



2D Results



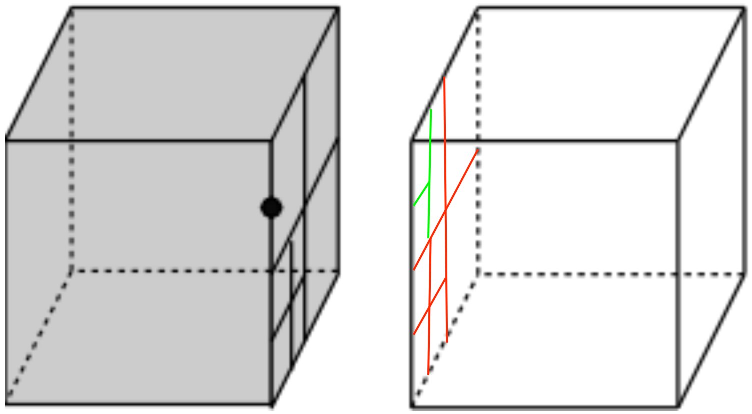


Extracted contour

Cells due to added samples

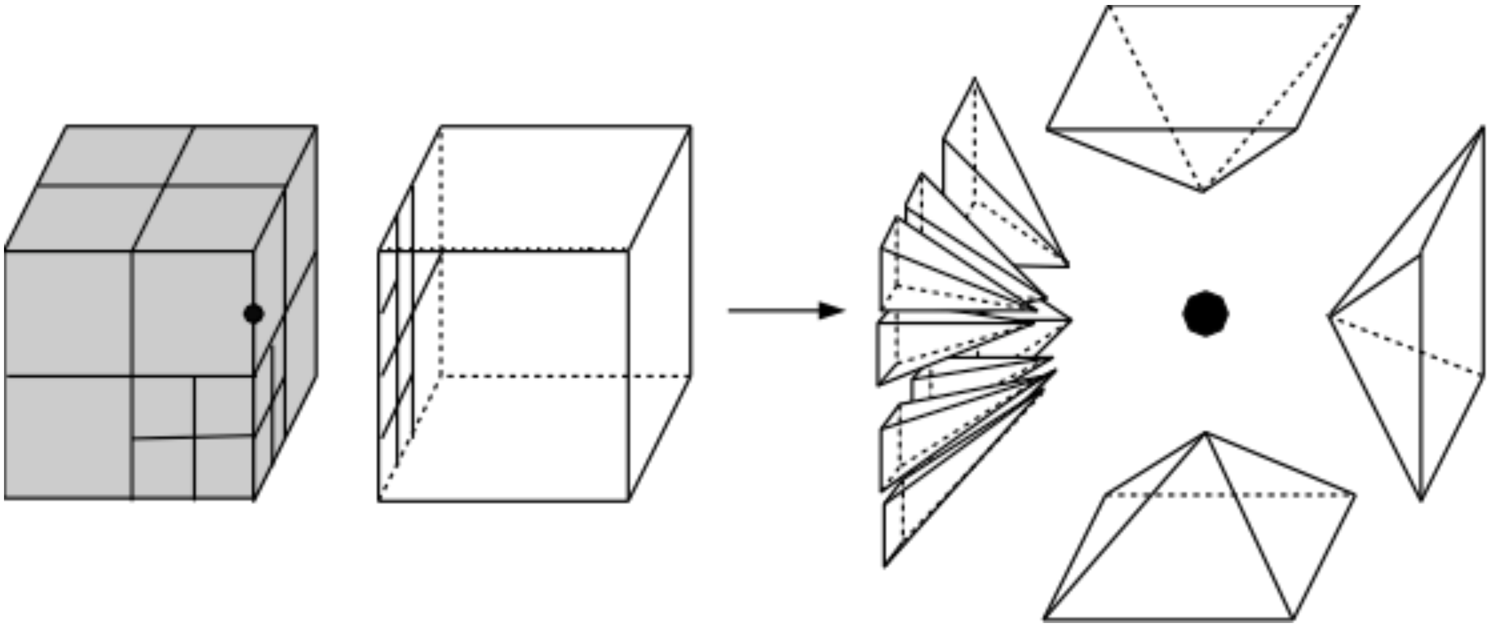
3D Cell Face Subdivision

- Subdivide lower-resolution cell face to match higher resolution face
- Subdivide cell face to eliminate hanging nodes

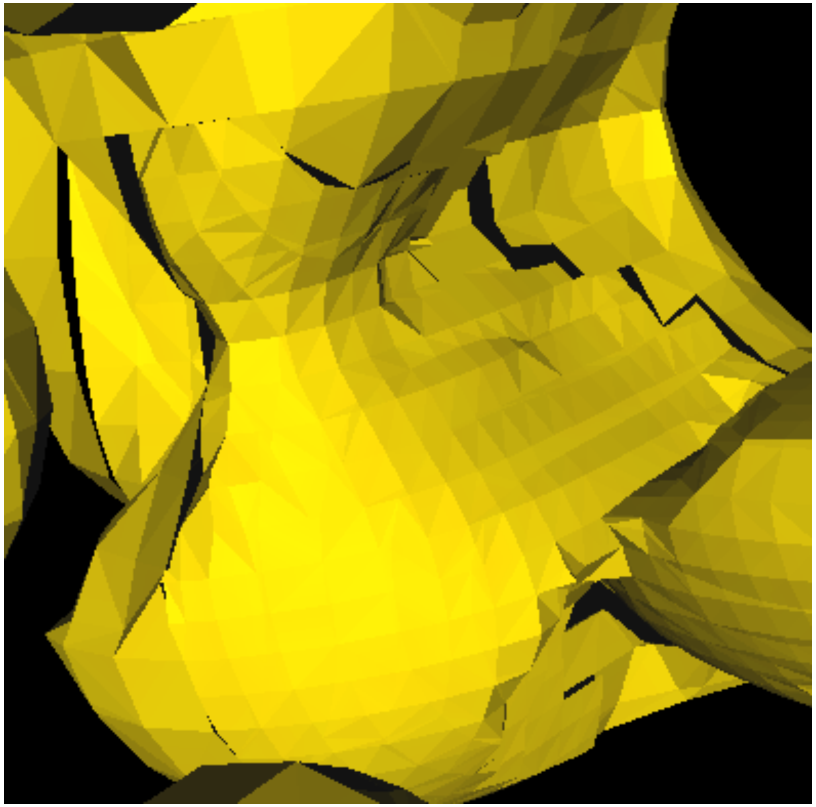


3D Cell Subdivision

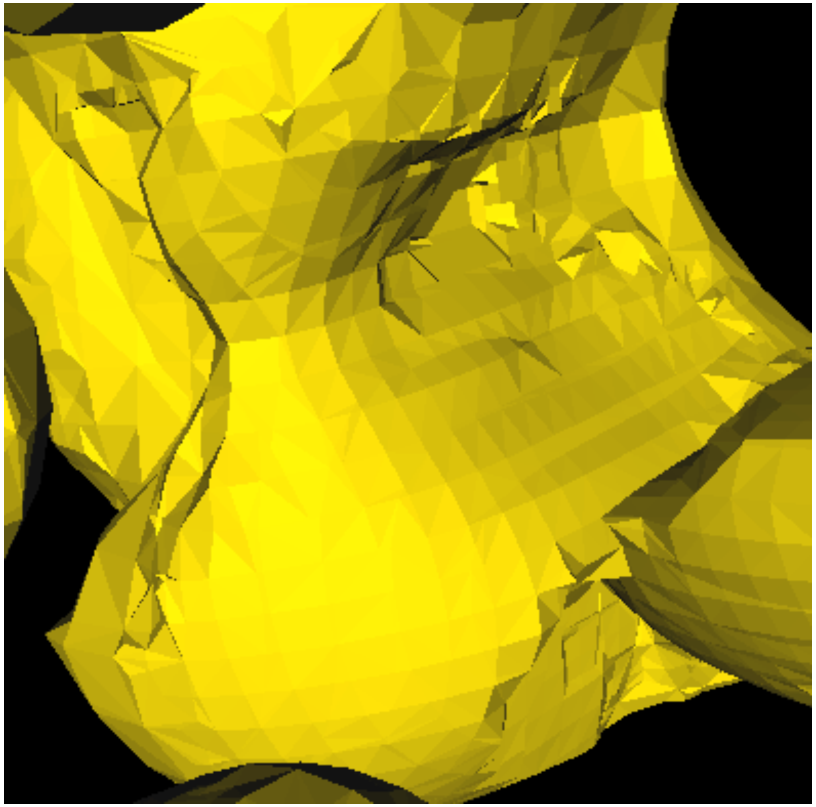
- Subdivide cell into pyramids with common apex point



Second Approach – Results



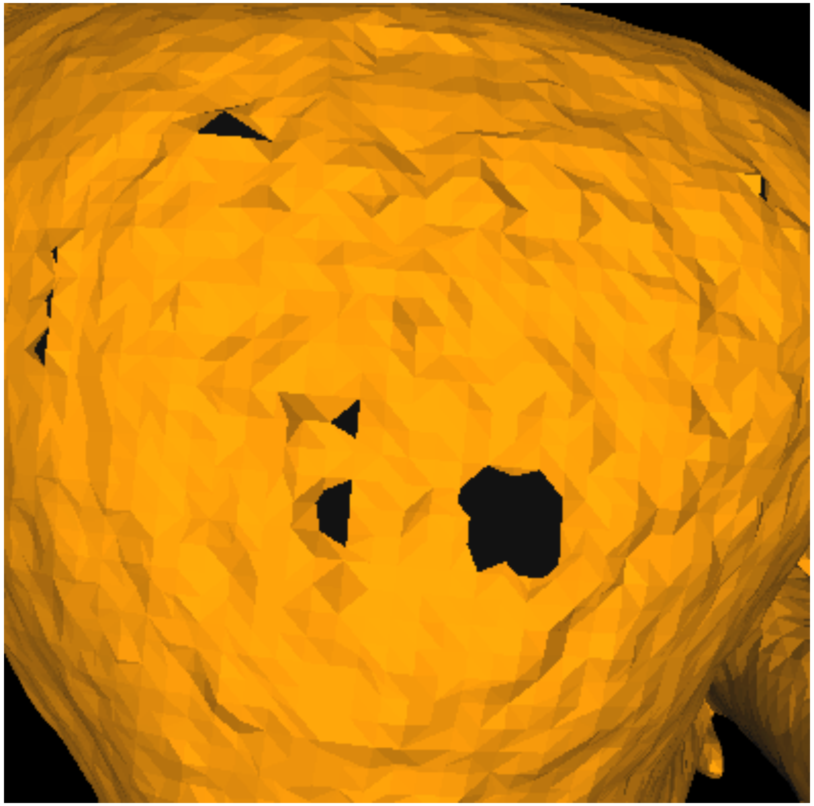
Cells: 44,332
Triangles: 10,456



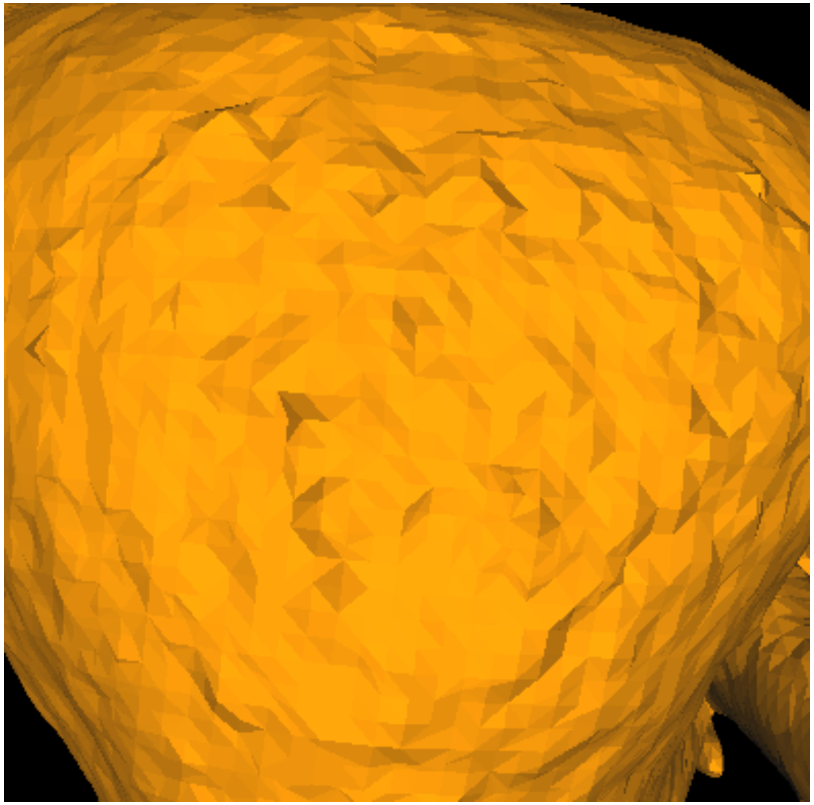
Cells: 74,358
Triangles: 14,332

Time: 2.30 sec

Second Approach – Results



Cells: 303,759
Triangles: 77,029



Cells: 680,045
Triangles: 78,127

Time: 7.73 sec

Volume Rendering

Hardware-accelerated Preview of AMR Data

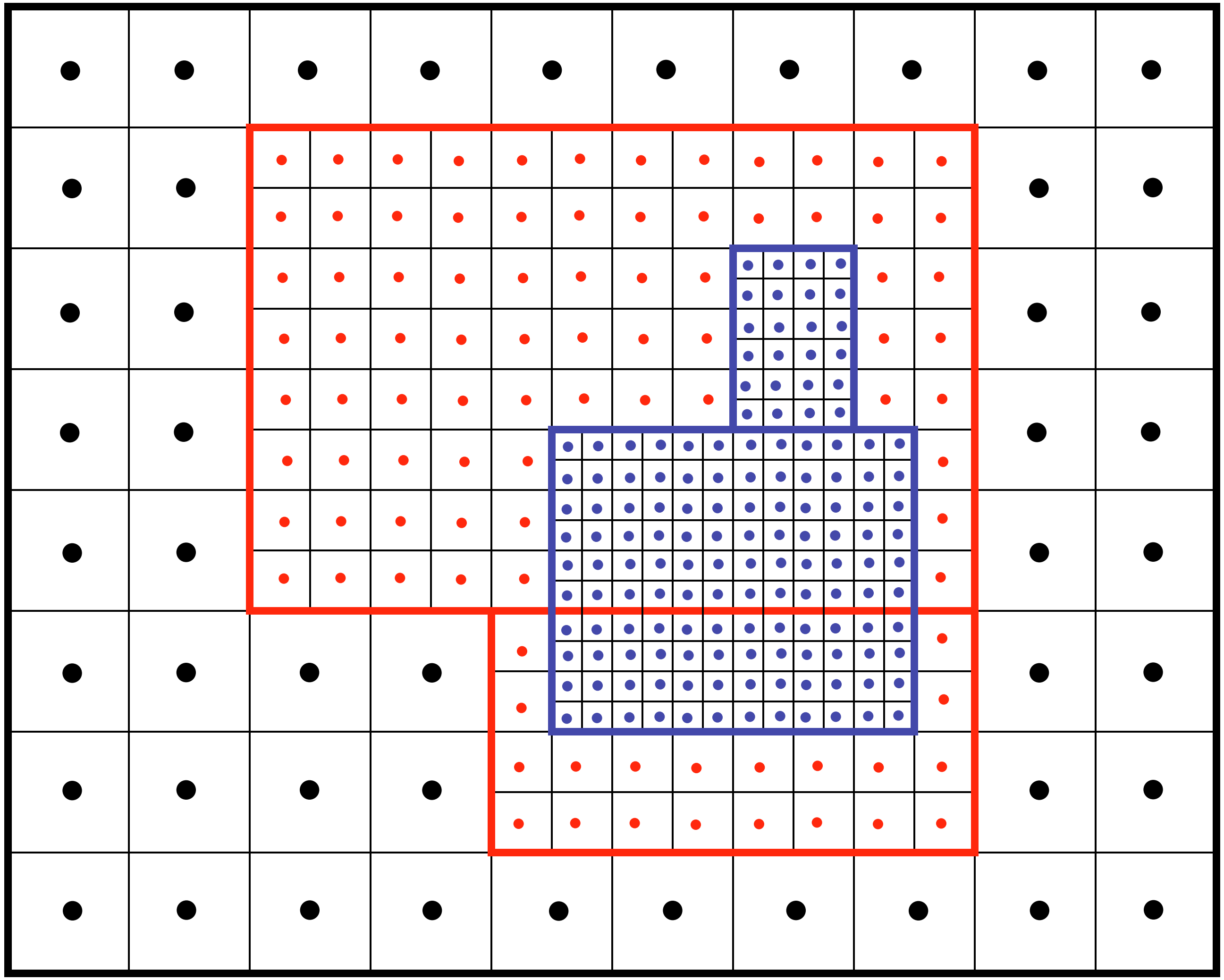
- Interactive DVR for choosing view point and transfer function

- Subdivide data set in regions of constant resolution

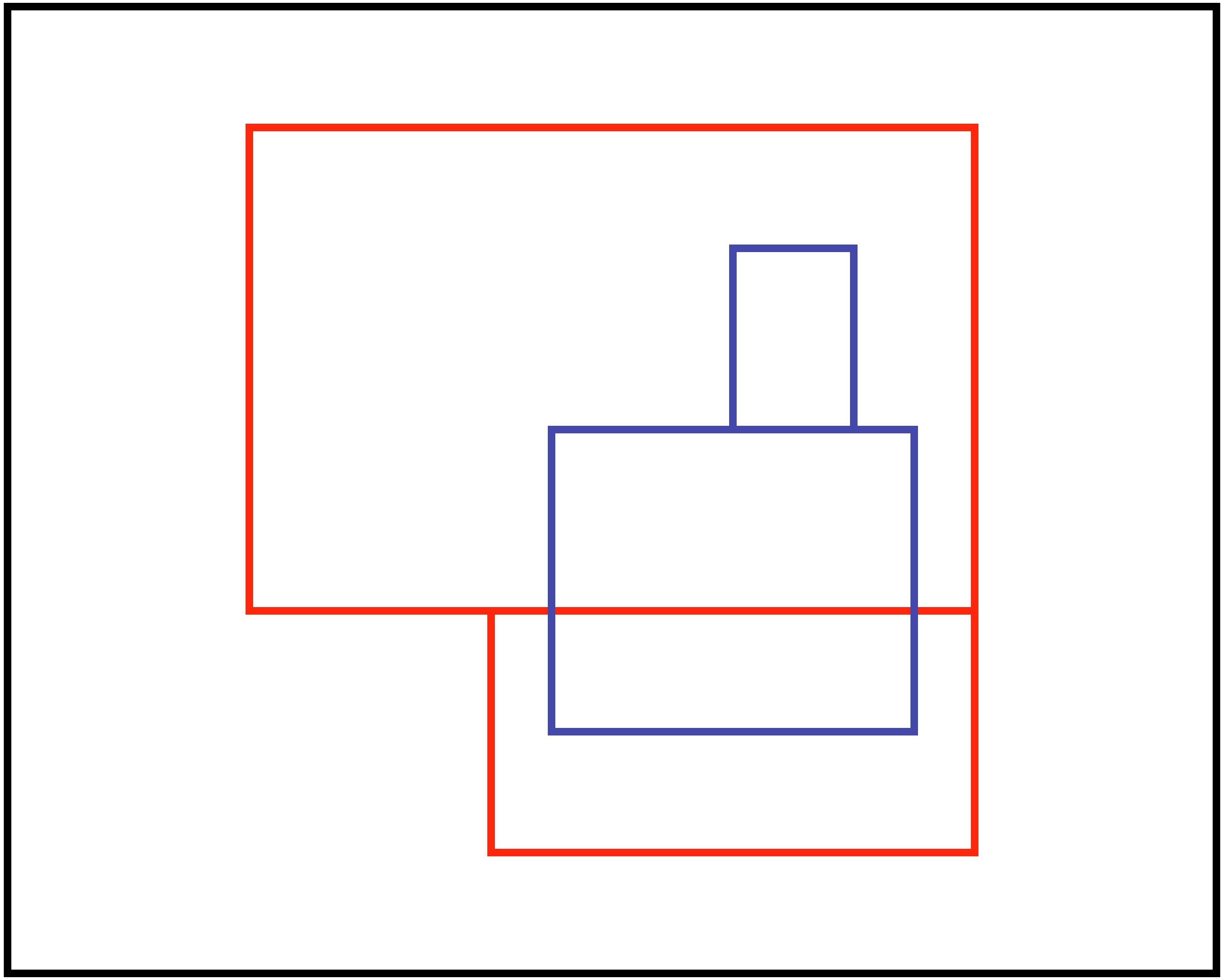
➔ AMR Partition Tree (generalized kD-tree)

- Traverse “AMR Partition tree” and render regions using hardware-accelerated DVR

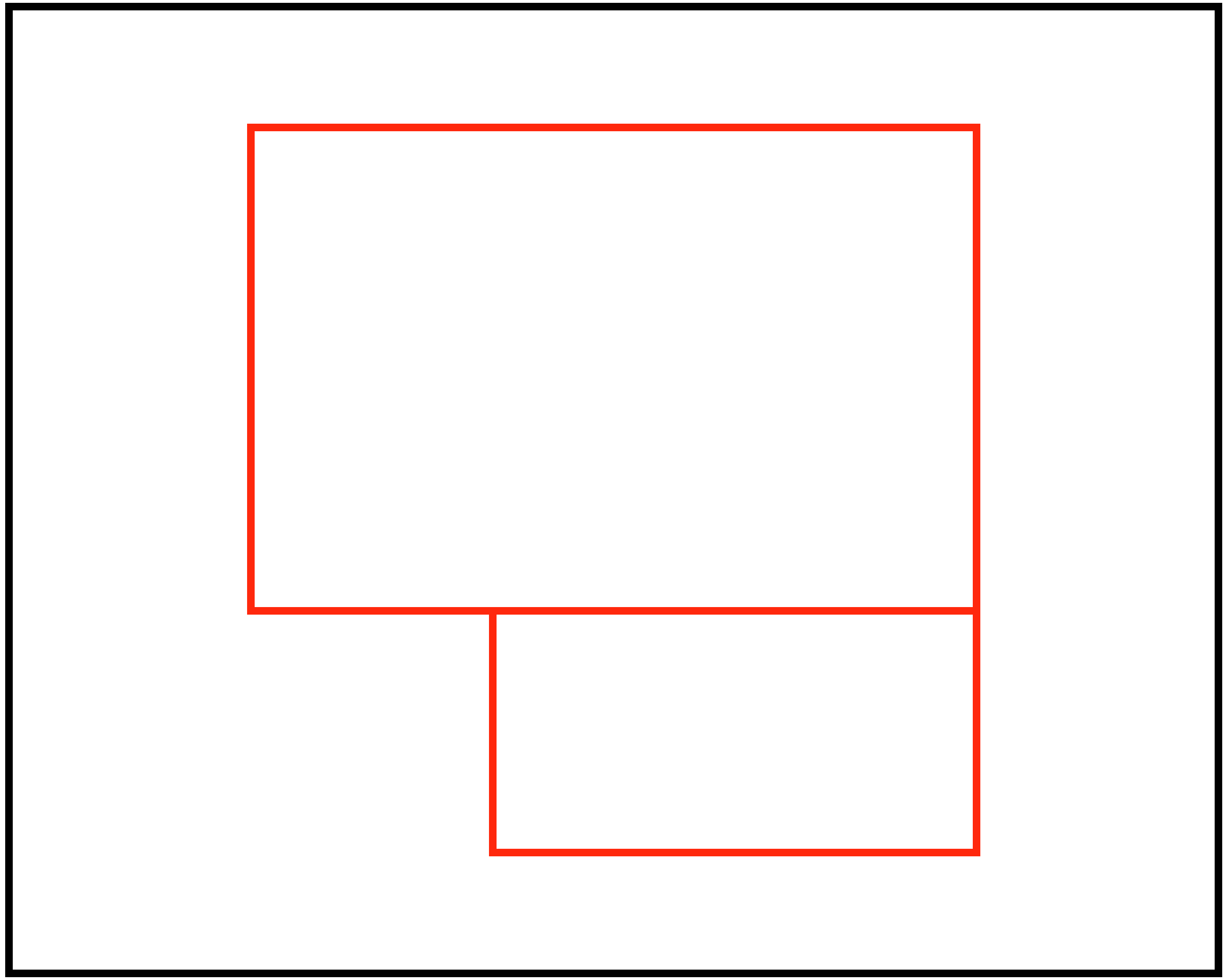
Homogenization



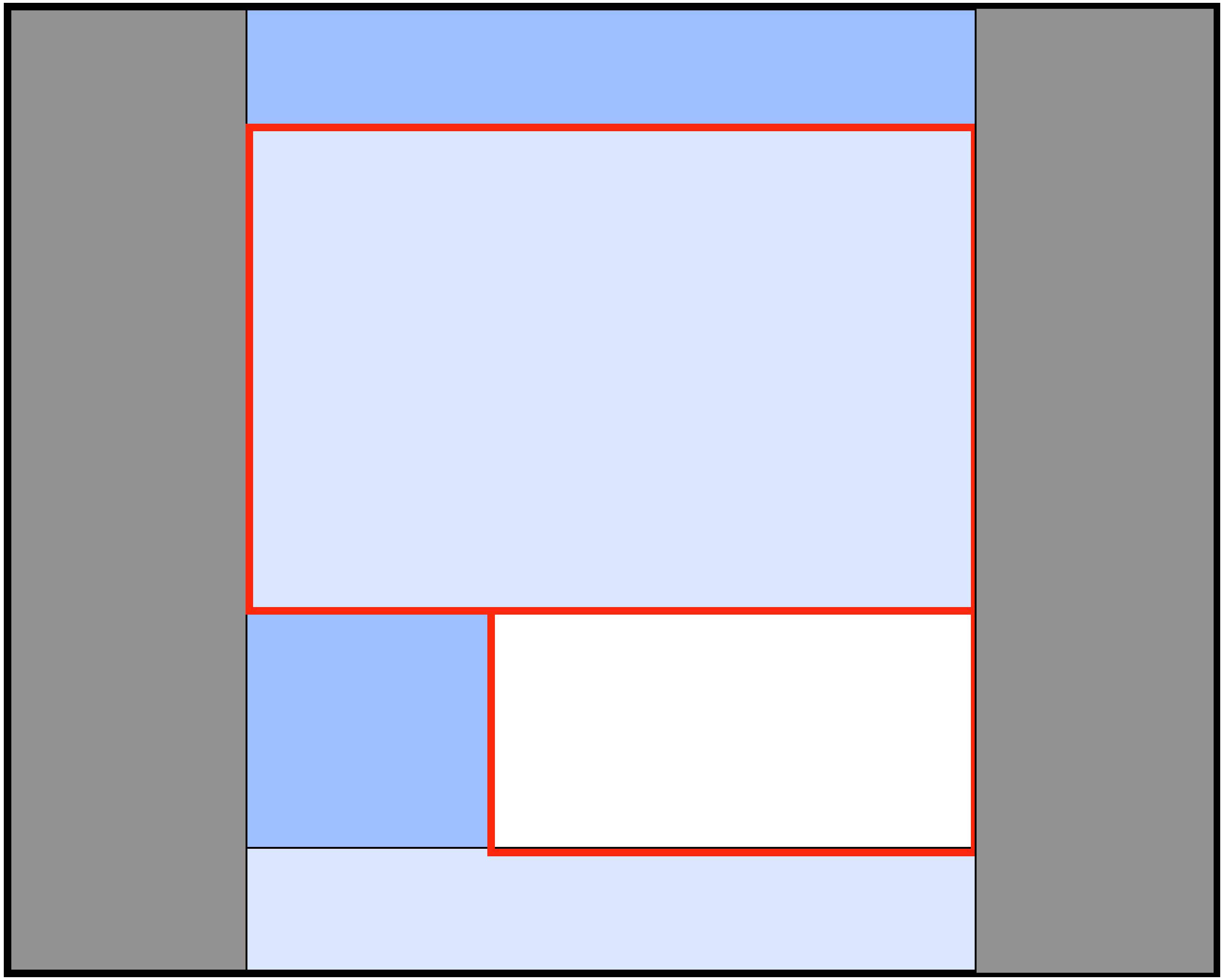
Homogenization



Homogenization



Homogenization

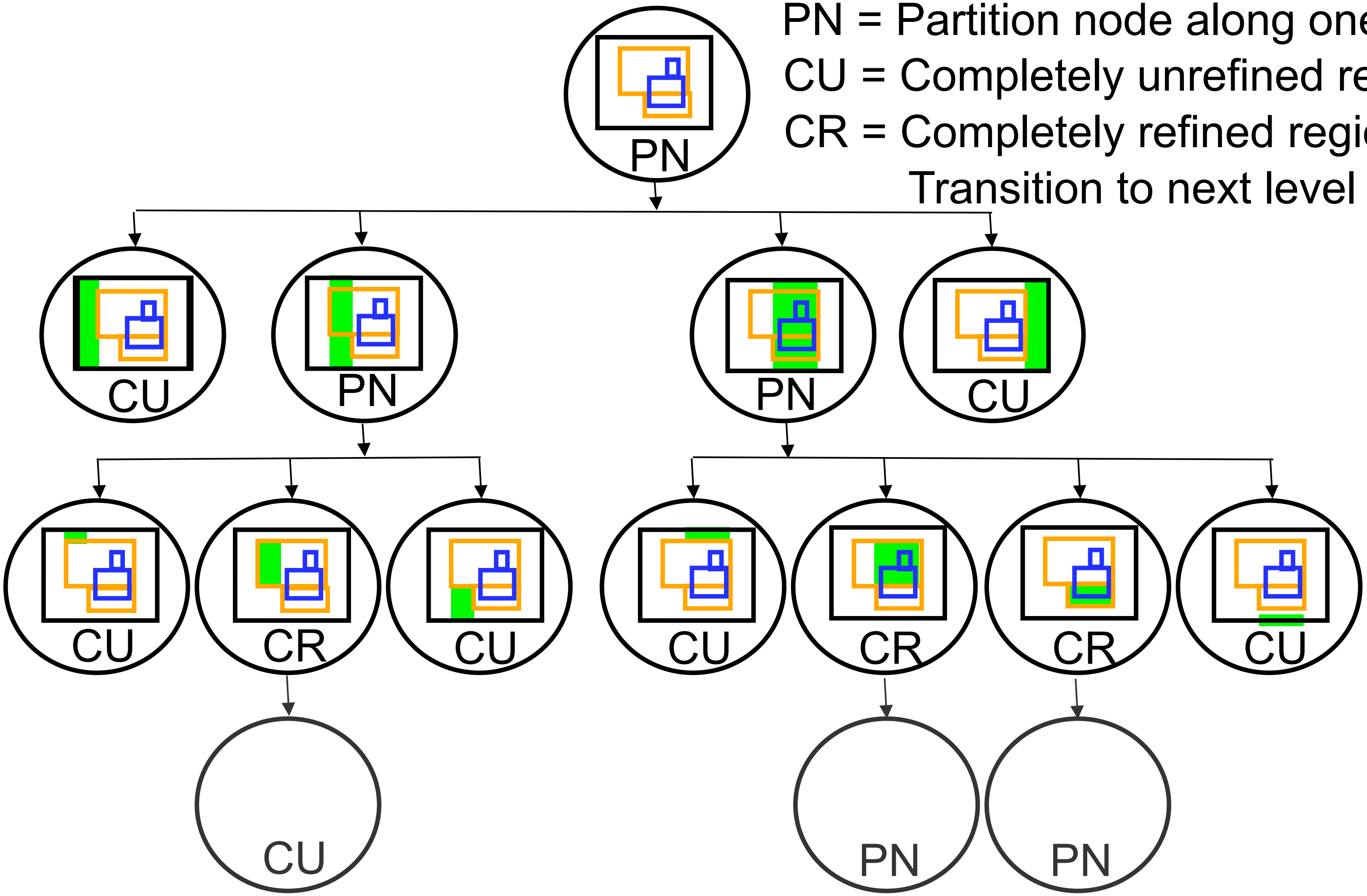


AMR Partition Tree

- Generalized kD-tree
- Partitions data-set into regions of constant resolution
- Node types:
  - Unrefined grid part (CU): Region is only available at resolution of current level
  - Completely refined grid part (CR): Region is completely available at next higher resolution
  - Partition node (PN): Partitions bounding box along one of the axes

Partition Tree – Example

PN = Partition node along one axis
CU = Completely unrefined region
CR = Completely refined region = Transition to next level

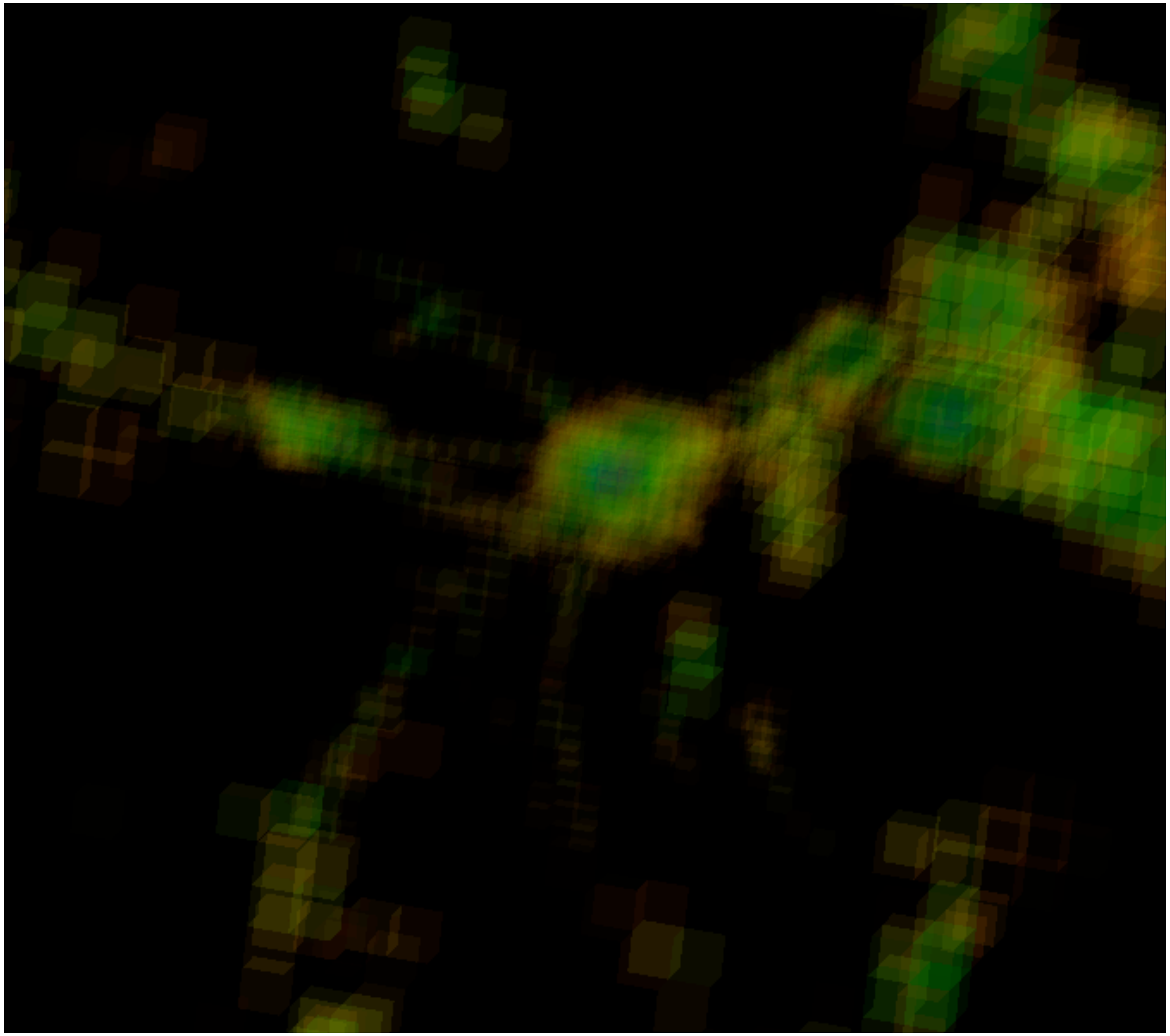


Adaptive Tree Traversal

- View-dependent criteria:
  - Avoid unnecessary computation time
  - No quality loss

- Time-dependent criteria:
  - Sacrifice render quality to obtain specified frame rate

Hardware-accelerated Rendering – Interactive

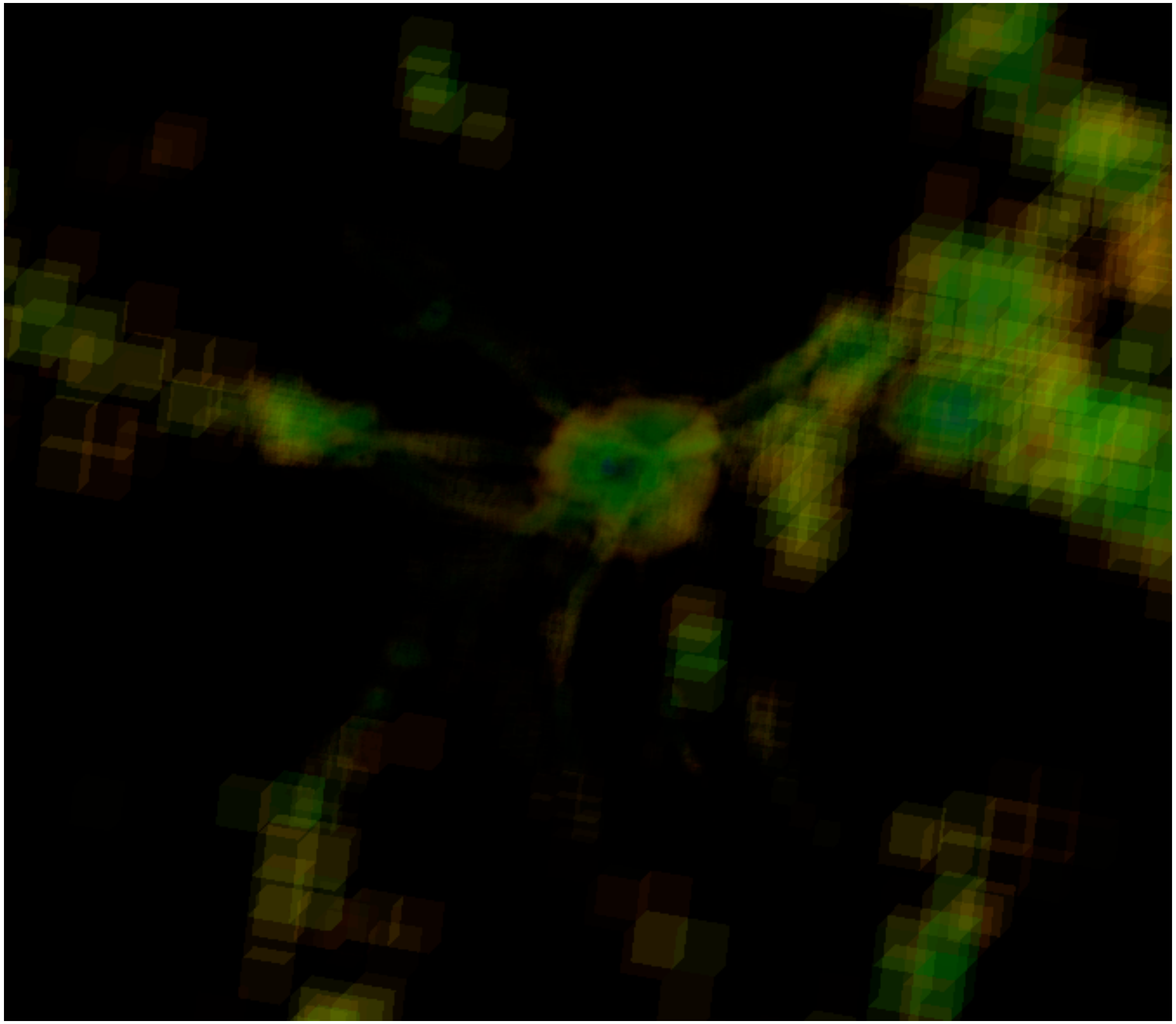


AMR simulation of star cluster formation

Root level 32x32x32

[Data set: Greg Bryan, Theoretical Astronomy Group, MIT]

Hardware-accelerated Rendering – Maximum Quality



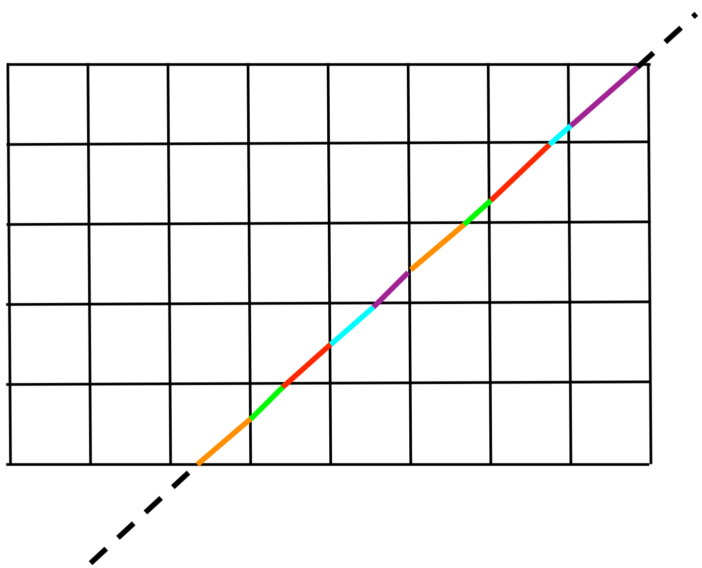
AMR simulation of star cluster formation

Root level 32x32x32

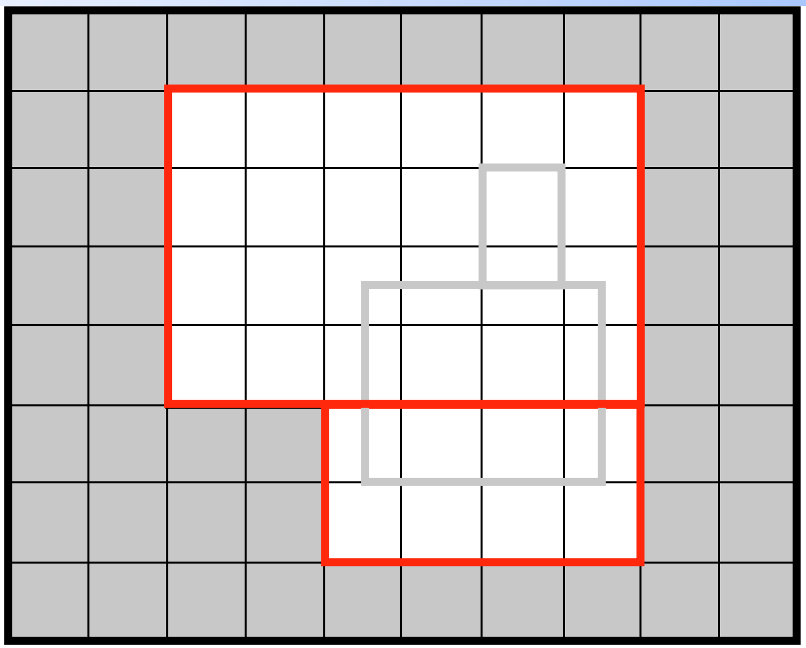
[Data set: Greg Bryan, Theoretical Astronomy Group, MIT]

High-quality DVR of AMR Data

- **Use "cell projection" [Ma & Crockett 1997] to display individual patches**
  - **Traverse patches and construct ray segments [object space based]**
  - **Ma & Crockett: Sort ray segments**

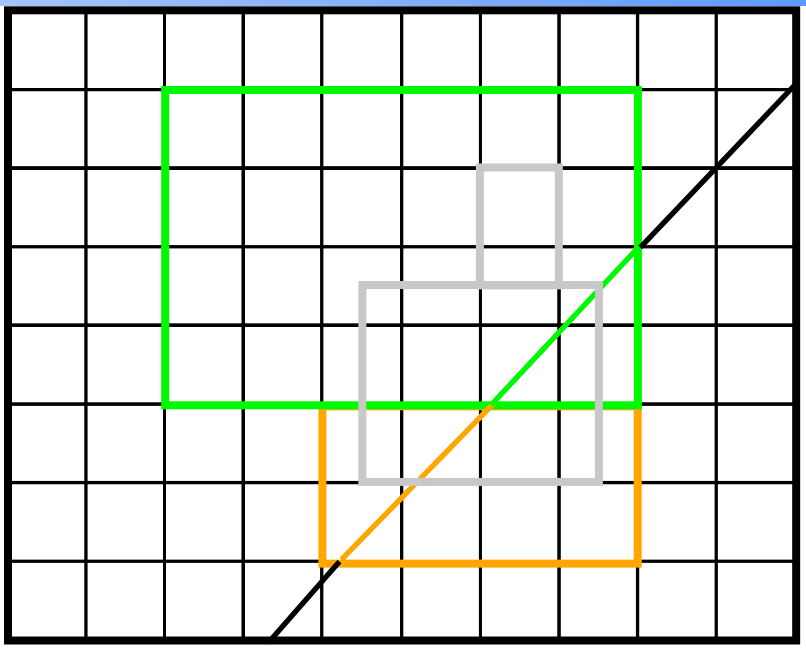


Progressive DVR of AMR Data



Bottom-up

Render fine grids, fill gaps with coarse grid data

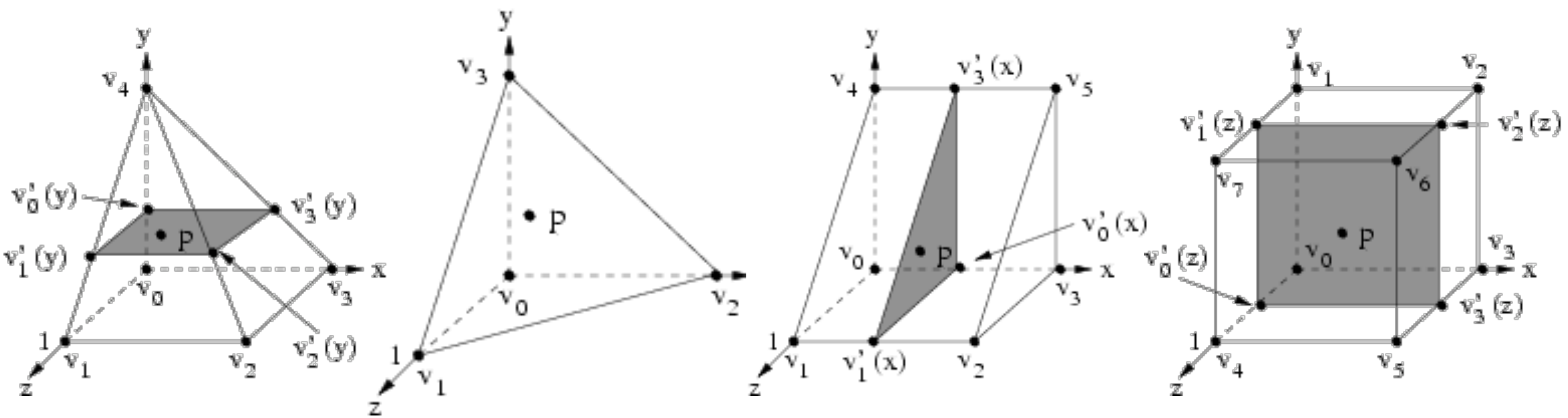


Top-down

Render coarse grids (preview), replace data with finer representation

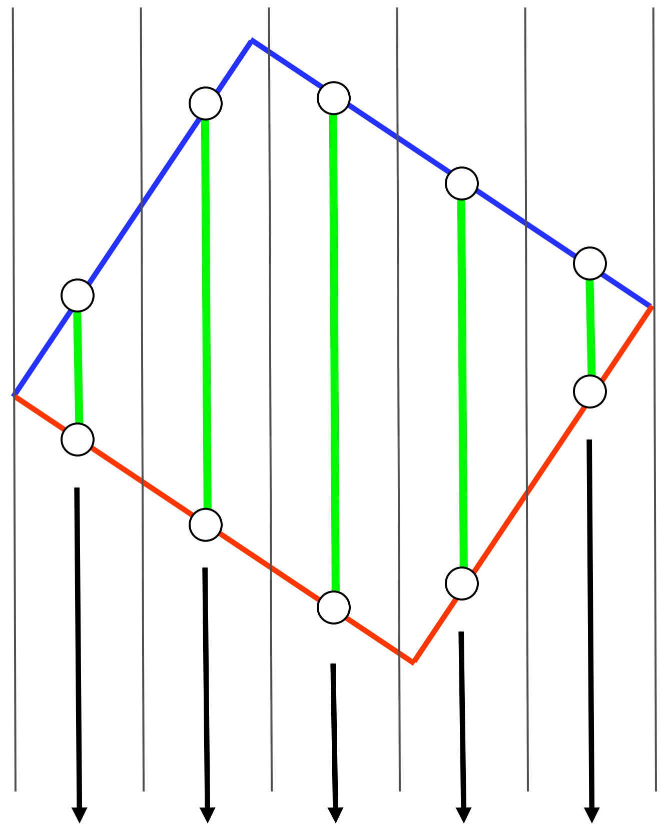
Interpolation

- **Nearest neighbor (constant) interpolation → debugging**
- **Piecewise Linear Method (PLM) → Discontinuities**
- **Dual grids (trilinear) and stitch cells**

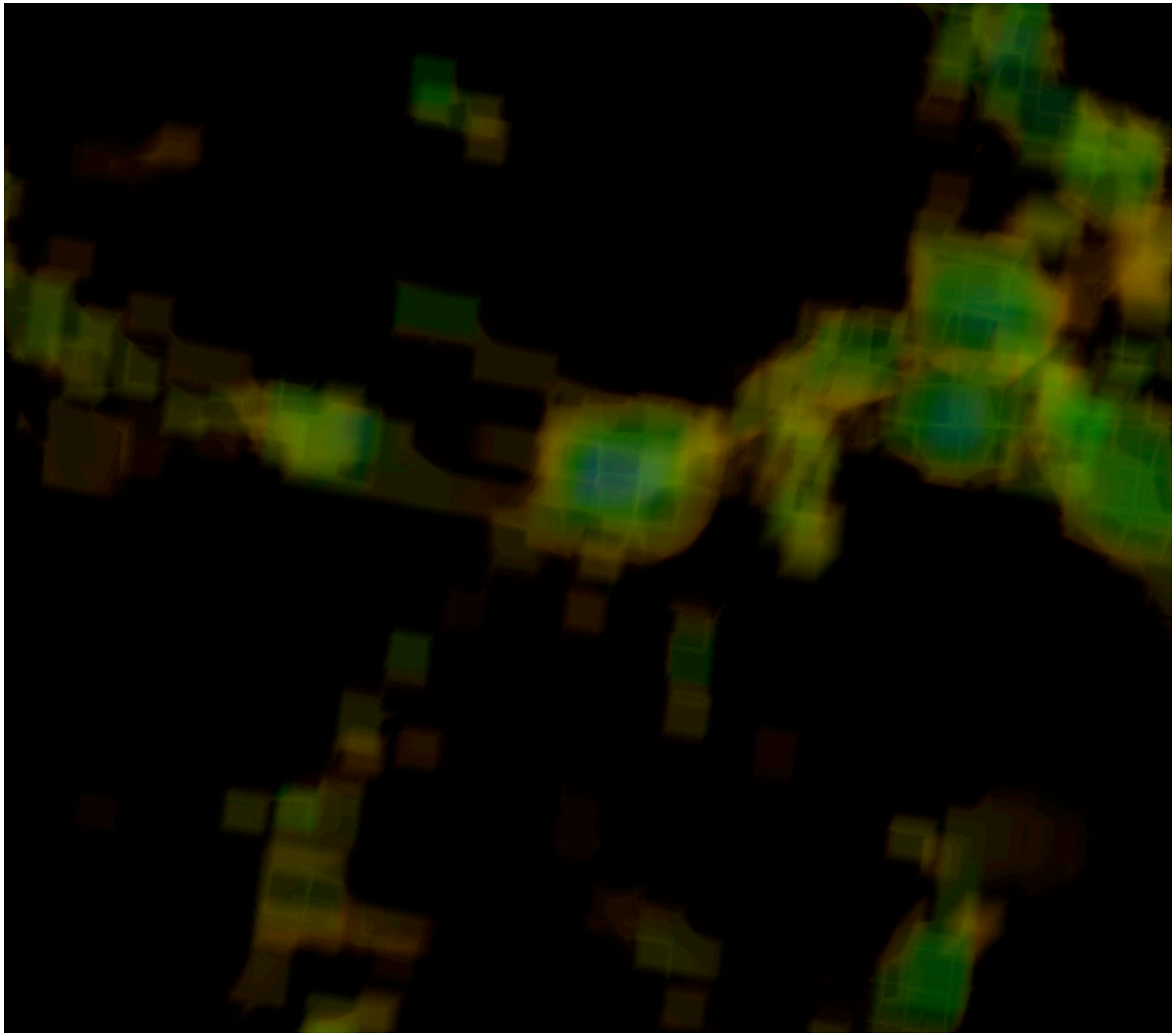


□ Bilinear △ Linear

Cell-projection — Scan Convert Front Facing Boundaries



Piecewise Linear Method – One Hierarchy Level

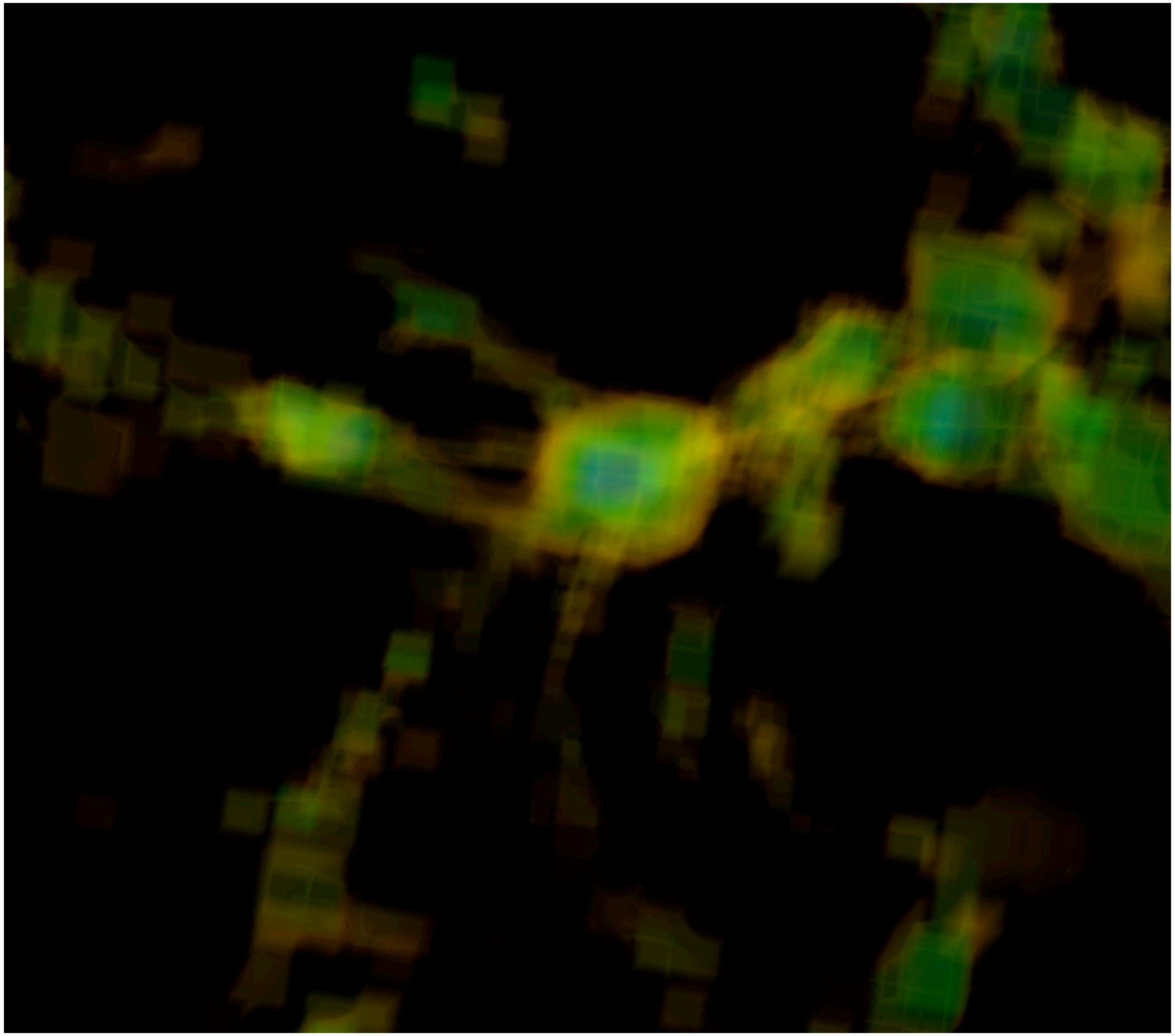


AMR simulation of star cluster formation

Root level 32x32x32

[Data set: Greg Bryan, Theoretical Astronomy Group, MIT]

Piecewise Linear Method – Two Hierarchy Levels

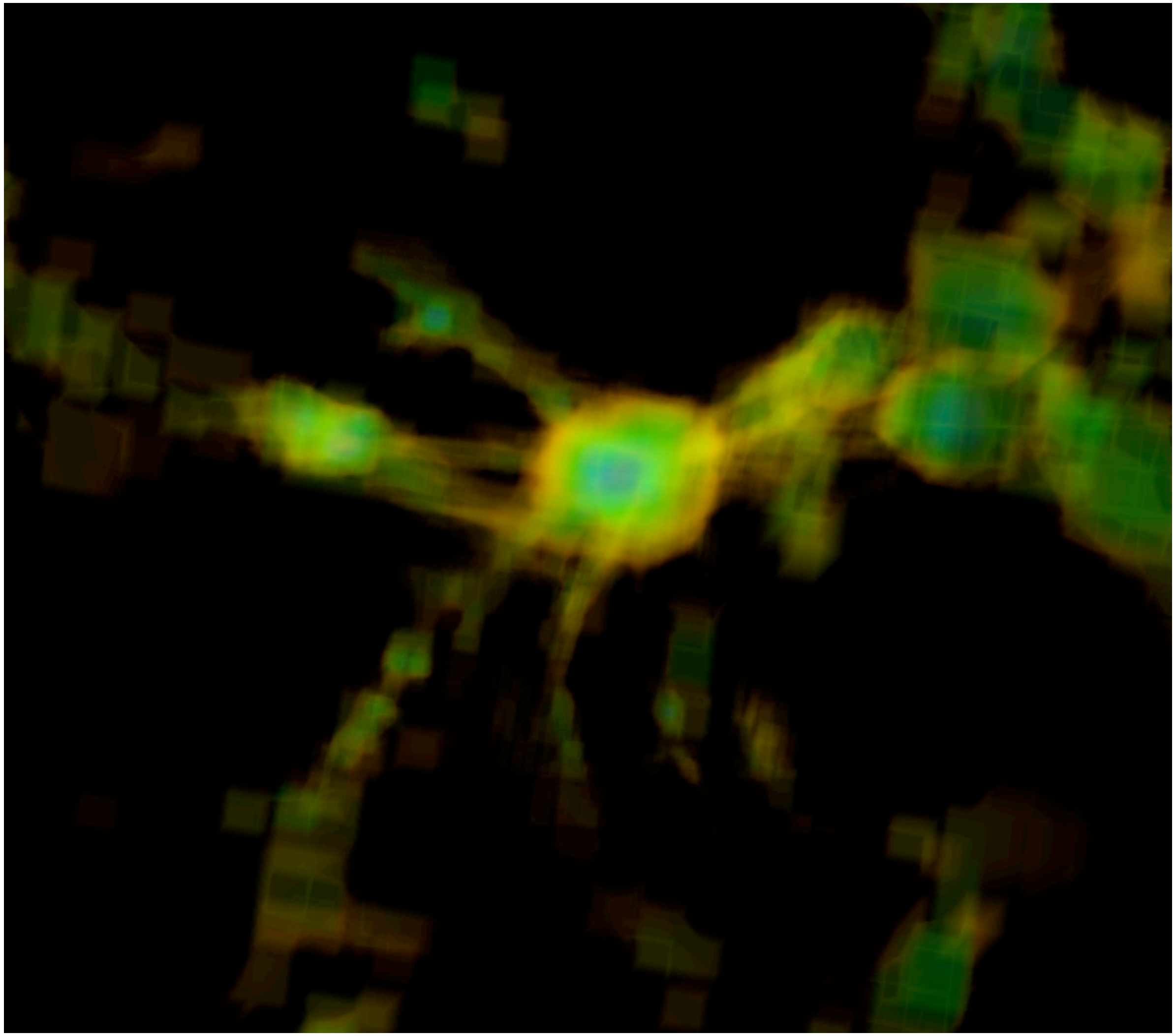


AMR simulation of star cluster formation

Root level 32x32x32

[Data set: Greg Bryan, Theoretical Astronomy Group, MIT]

Piecewise Linear Method – Three Hierarchy Levels



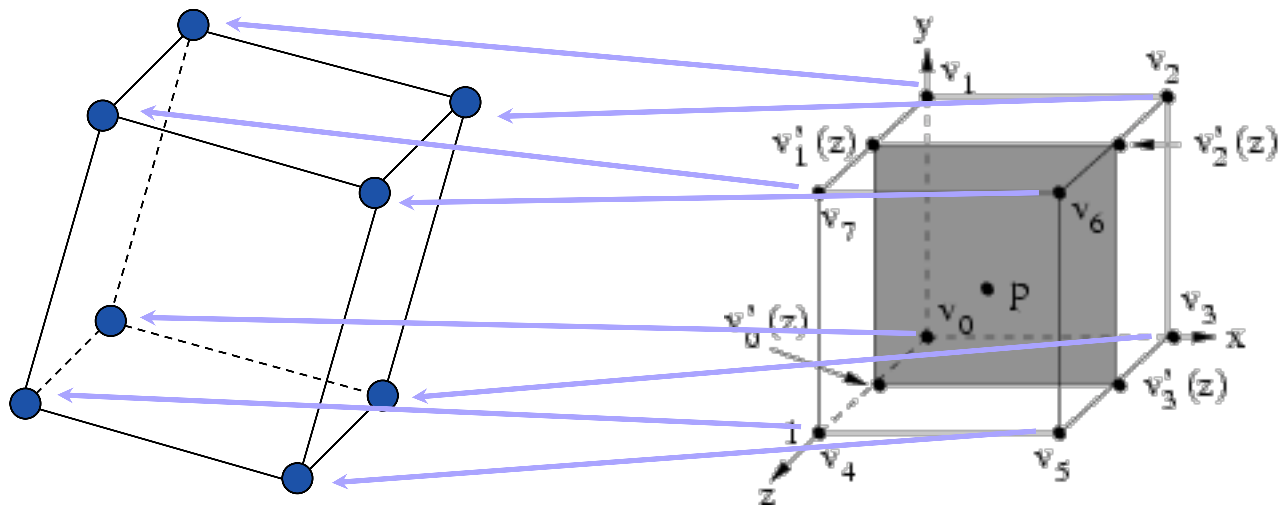
AMR simulation of star cluster formation

Root level 32x32x32

[Data set: Greg Bryan, Theoretical Astronomy Group, MIT]

Mapping to Standard Elements (1/3)

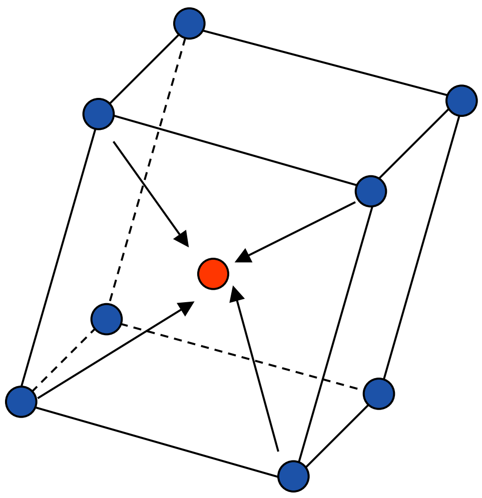
- Save standard element coordinates in cell vertices



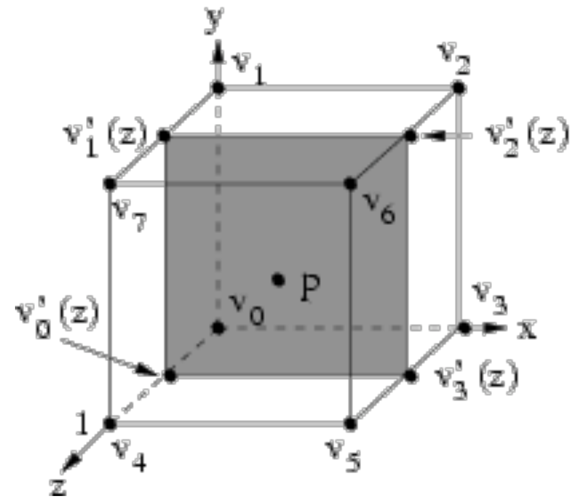
World coordinates **Standard element coordinates**

Mapping to Standard Elements (2/3)

- Interpolate standard element coordinates during rasterization



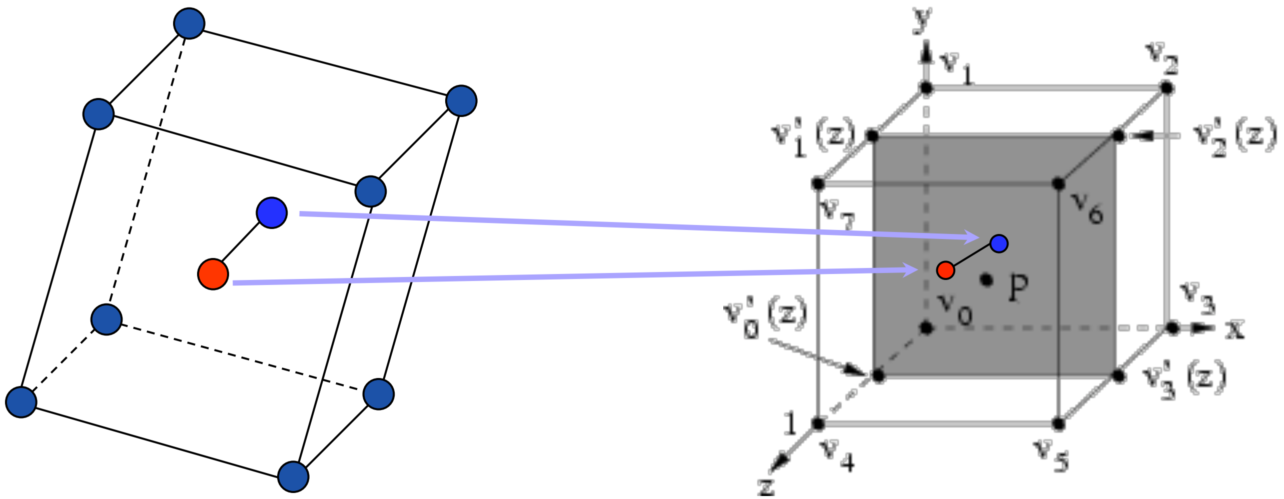
World coordinates



Standard element coordinates

Mapping to Standard Elements(3/3)

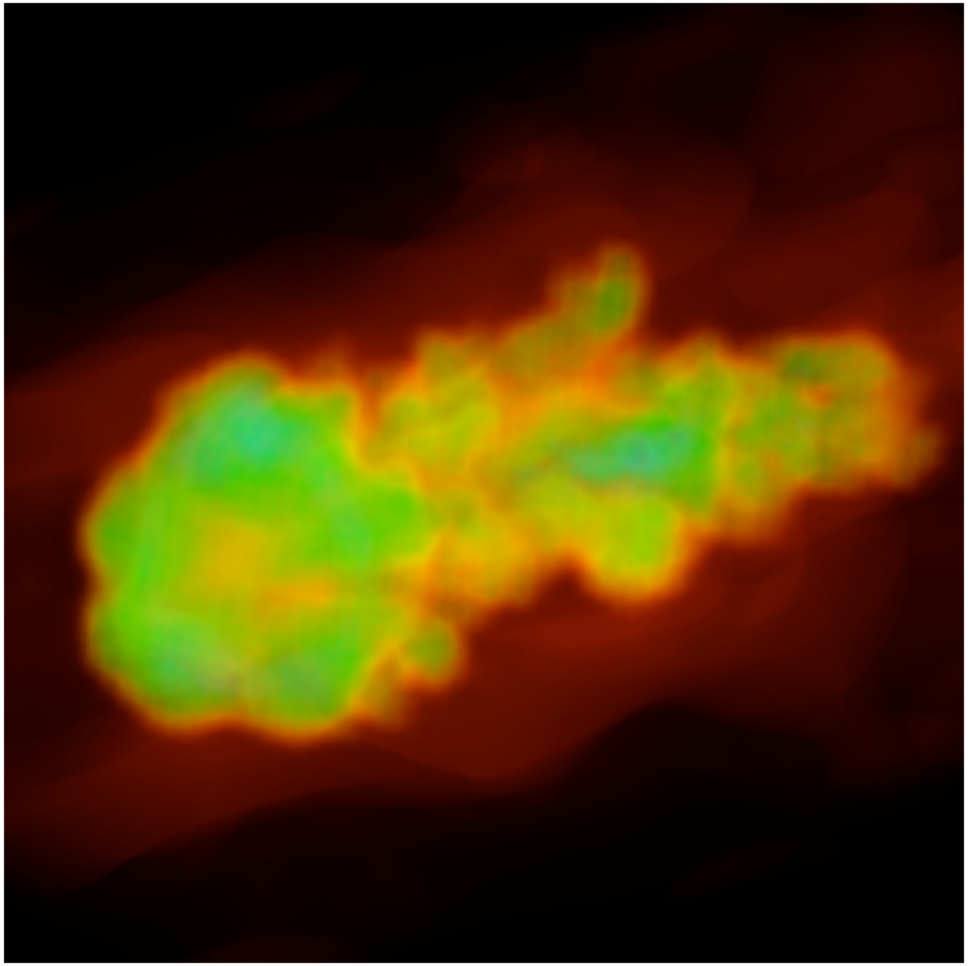
- Use standard element coordinates for interpolation along ray segment



World coordinates

Standard element coordinates

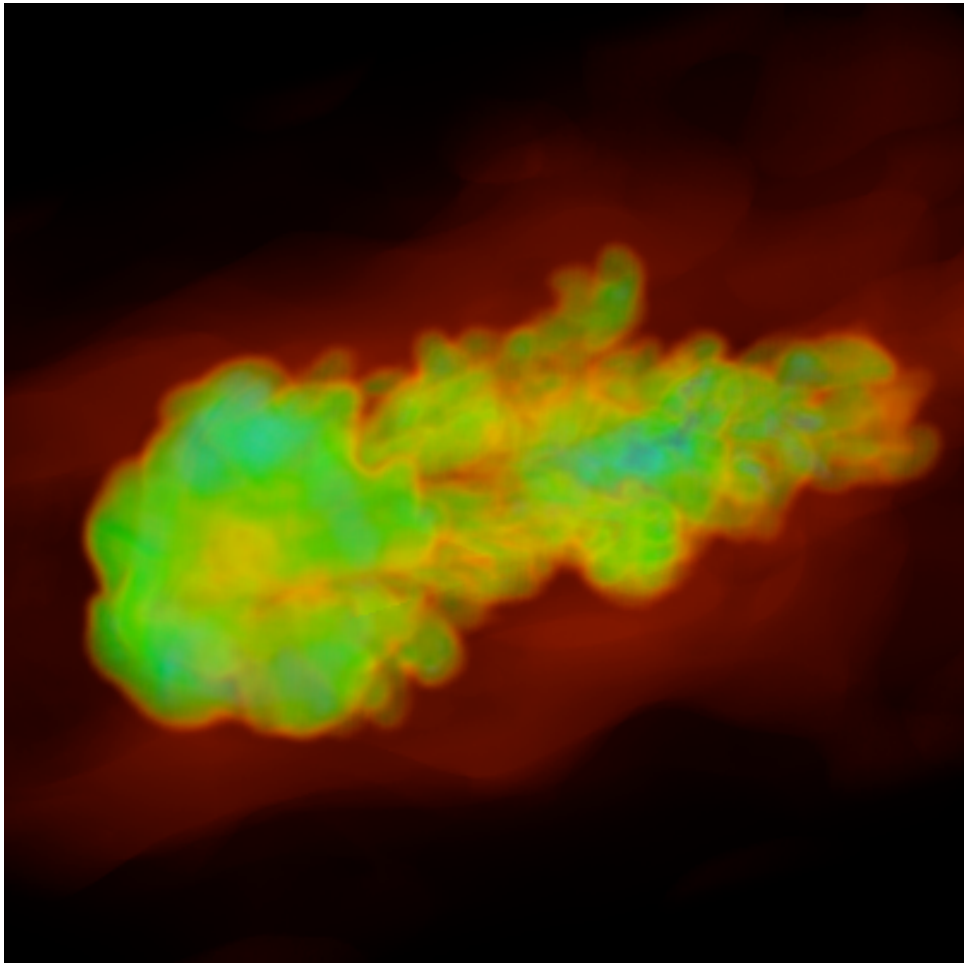
Interpolation with Stitch Cells – One Hierarchy Level



Simulation of an Argon bubble in a surrounding gas hit by a shockwave

[Data set: Center for Computational Sciences and Engineering (CCSE), Lawrence Berkeley National Laboratory]

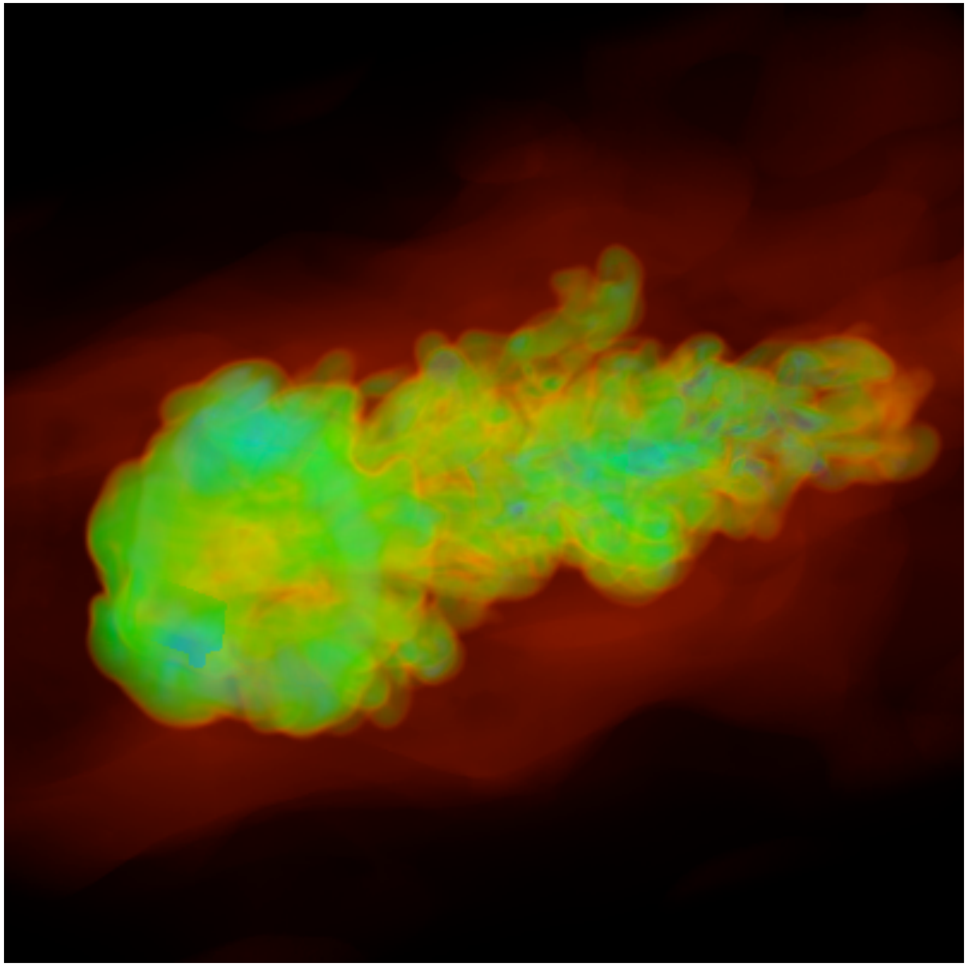
Interpolation with Stitch Cells – Two Hierarchy Levels



Simulation of an Argon bubble in a surrounding gas hit by a shockwave

[Data set: Center for Computational Sciences and Engineering (CCSE), Lawrence Berkeley National Laboratory]

Interpolation with Stitch Cells – Three Hierarchy Levels



Simulation of an Argon bubble in a surrounding gas hit by a shockwave

[Data set: Center for Computational Sciences and Engineering (CCSE), Lawrence Berkeley National Laboratory]

Level-dependent Transfer Functions

- Problem case: A fine level is completely enclosed within a coarse level

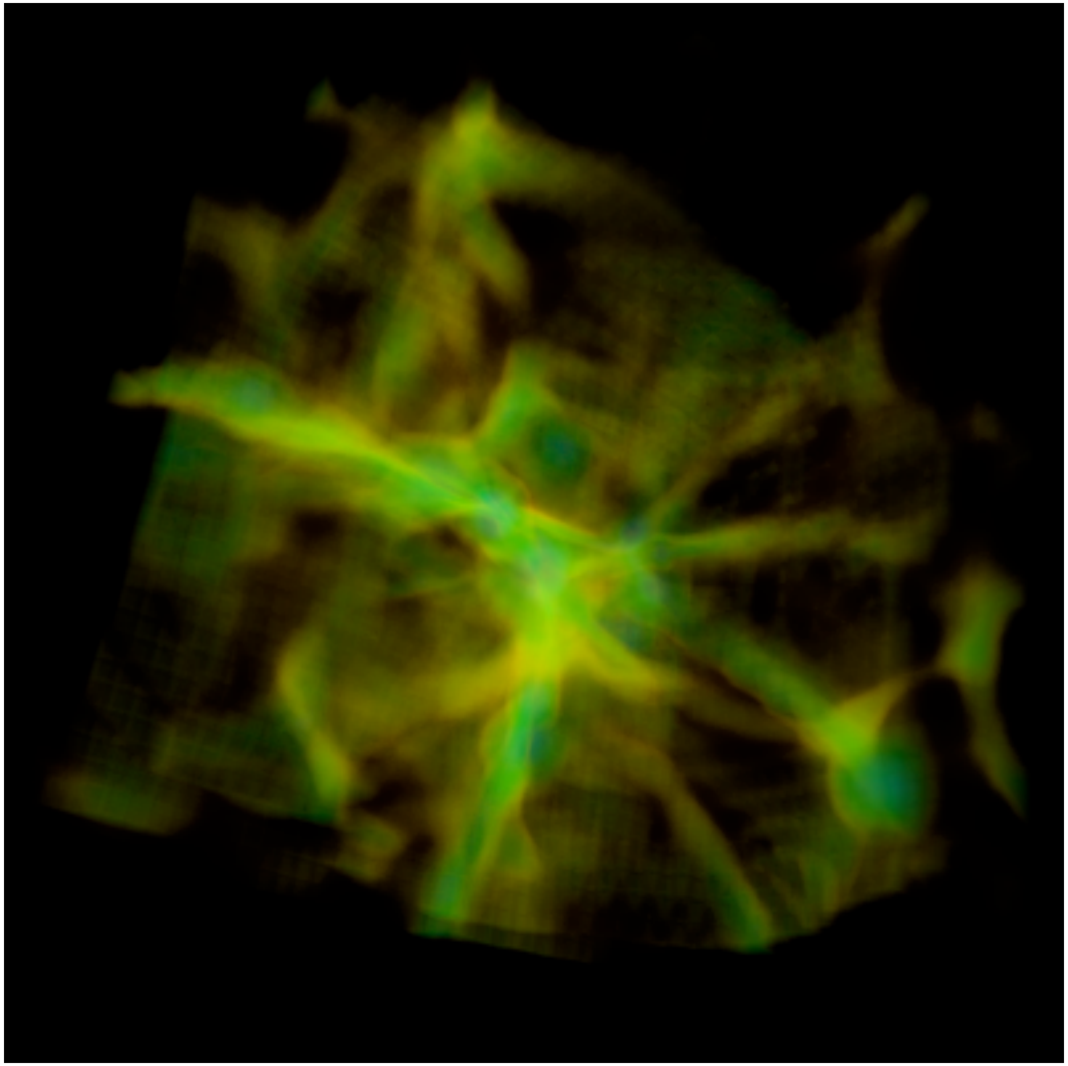
➨ The coarse level can hide interesting regions of the fine level

- Coarse level necessary to provide „context“ (orientation aid) for fine level

➨ Cannot completely discard coarse level

➨ Scale opacity and/or color saturation of coarse level

No Transfer Function Scaling

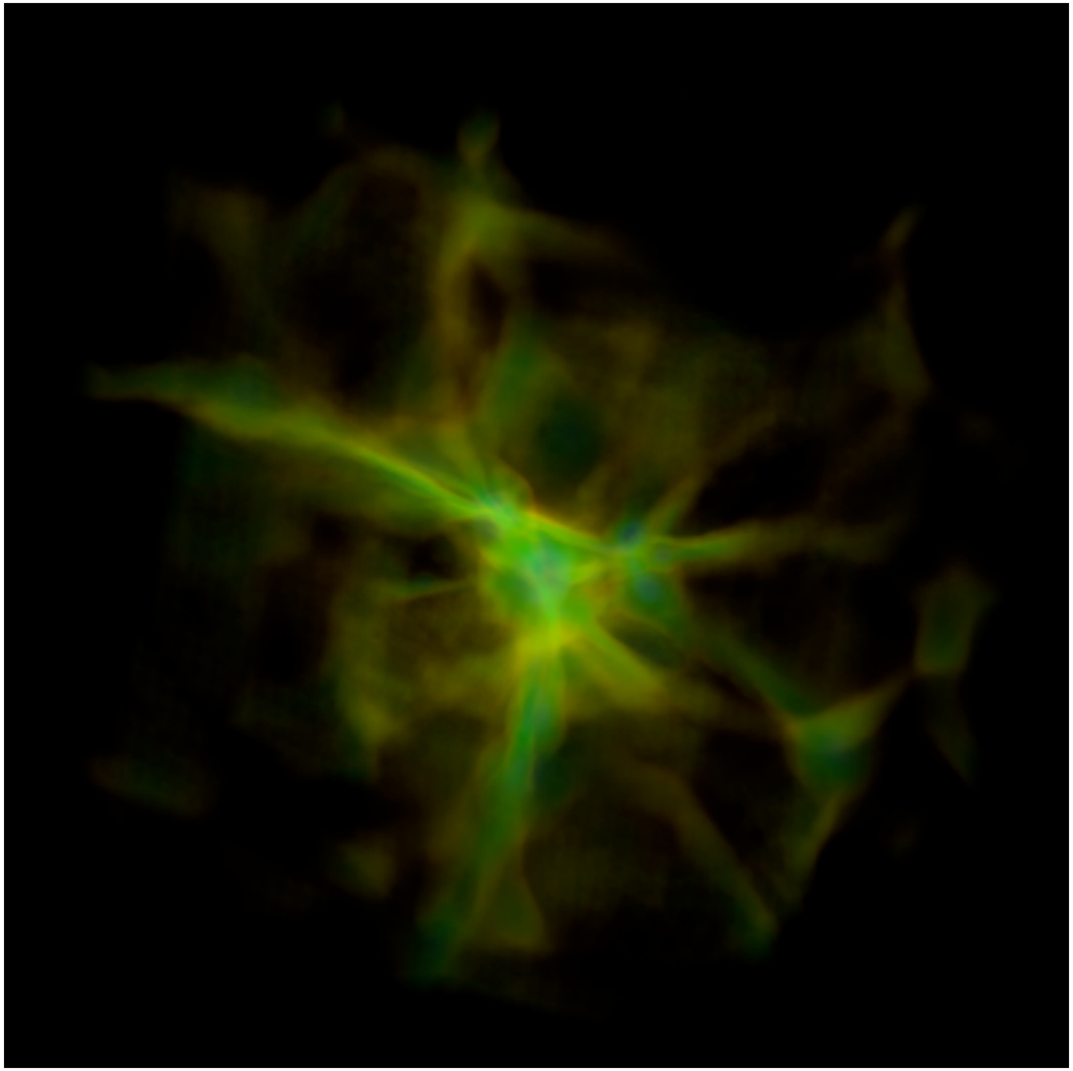


AMR simulation of star cluster formation

Root level 32x32x32

[Data set: Greg Bryan, Theoretical Astronomy Group, MIT]

Opacity Scaling

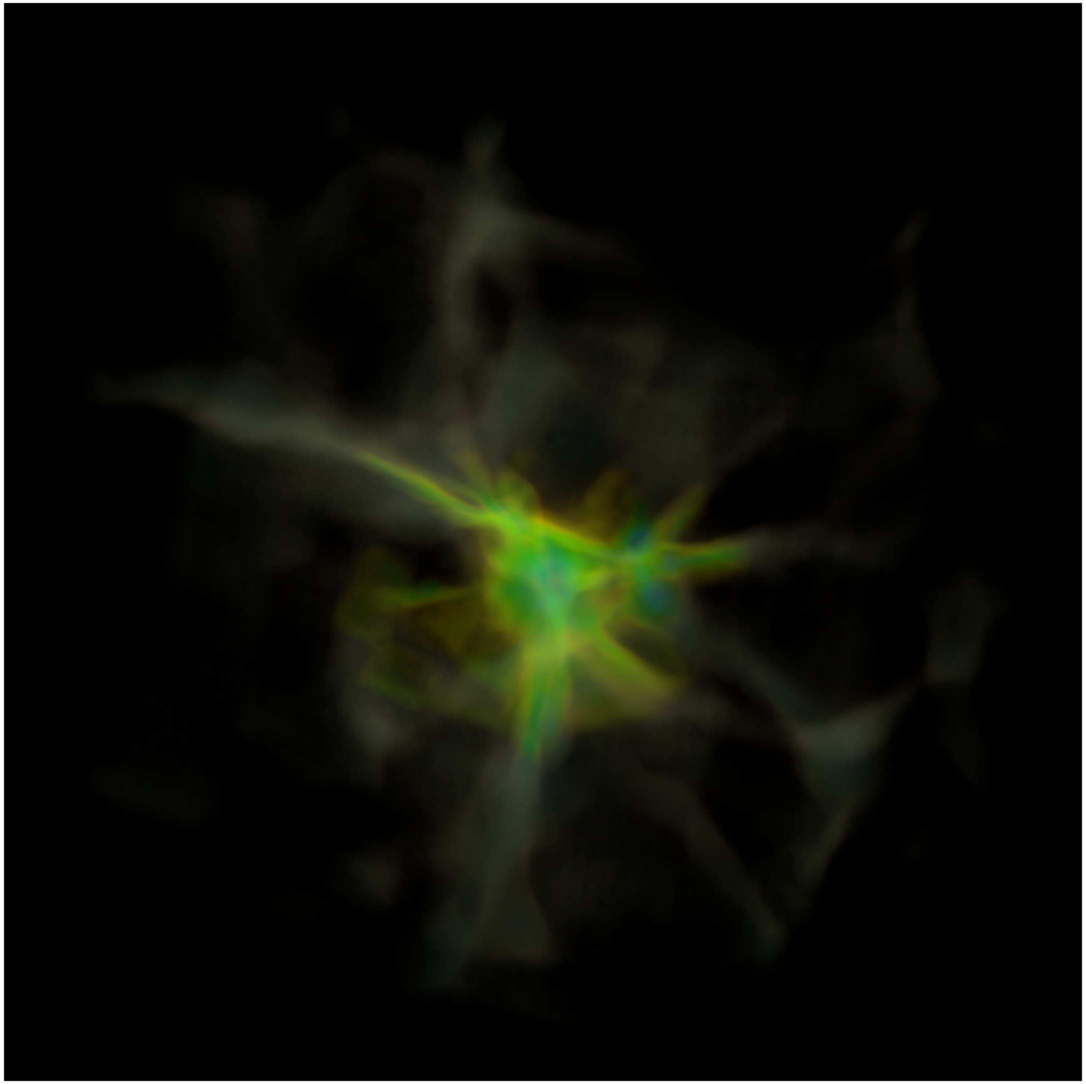


AMR simulation of star cluster formation

Root level 32x32x32

[Data set: Greg Bryan, Theoretical Astronomy Group, MIT]

Opacity and Saturation Scaling



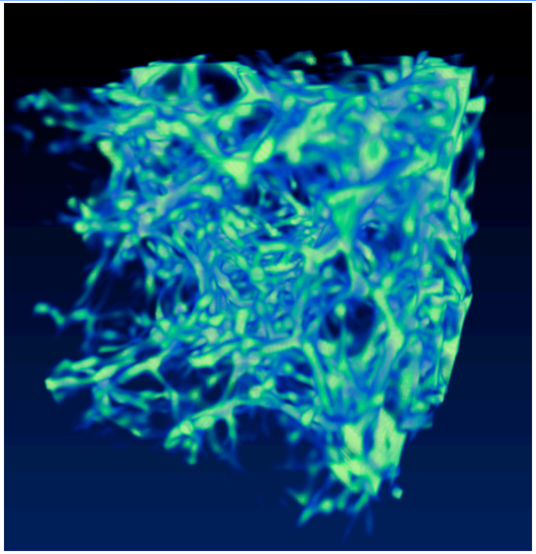
AMR simulation of star cluster formation

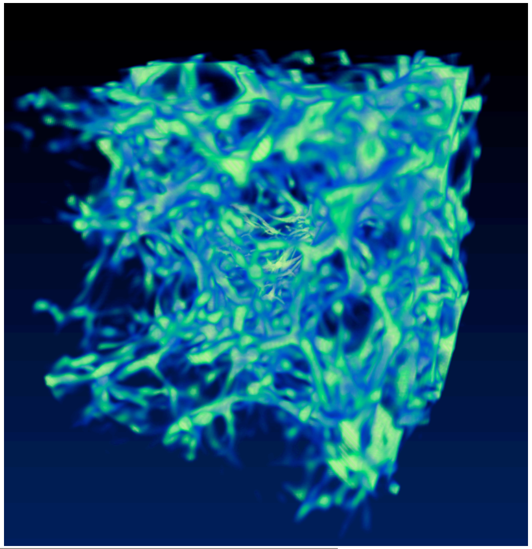
Root level 32x32x32

[Data set: Greg Bryan, Theoretical Astronomy Group, MIT]

Texture-based AMR Volume Rendering

- [Kähler & Hege, 2001 / 2002]
- Resample to node centered
- Subdivide in homogenous resolution regions (kD-tree)
- Minimize number of blocks using information about AMR grid placement algorithm
- Texture/Slicing-based volume rendering
- Optimized texture packing
- Adapt slice spacing & correct opacity



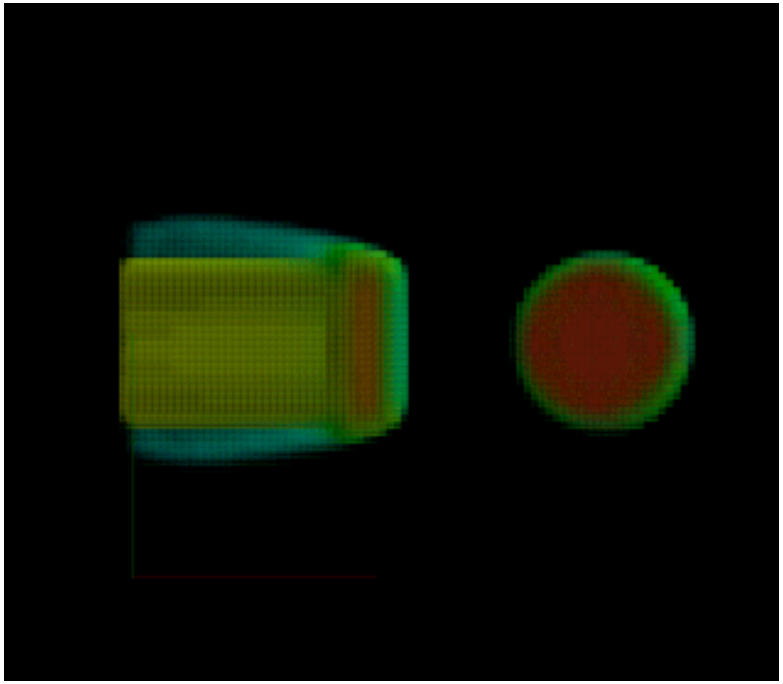


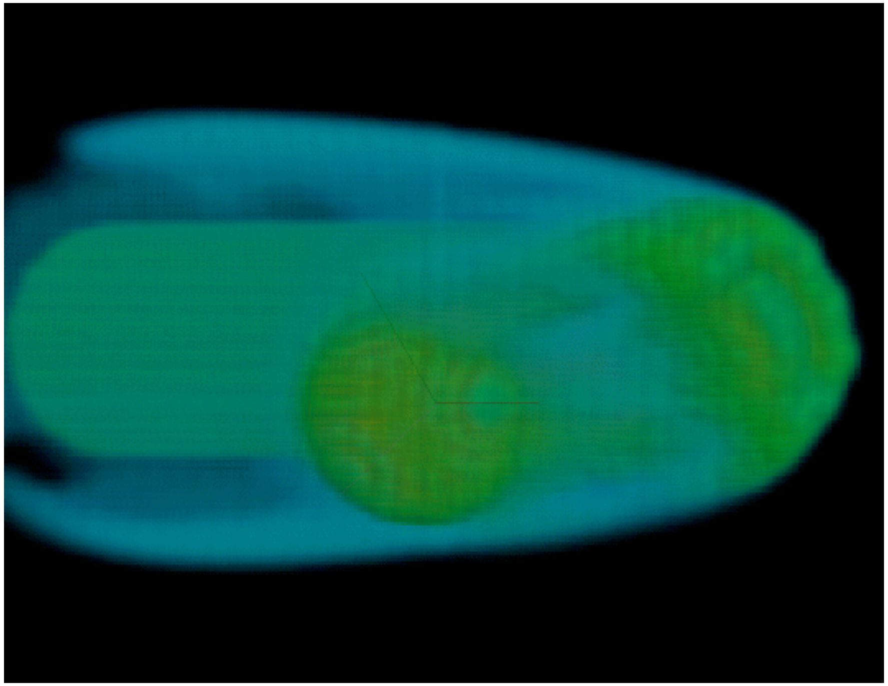
Rendering the First Star of the Universe

- [Kähler et al., 2002]: Application to astrophysical data set
  - Texture-based volume renderer, Virtual Director, CAVE
- Aired on Discovery Channel

Splatting-based Volume Rendering of AMR Data

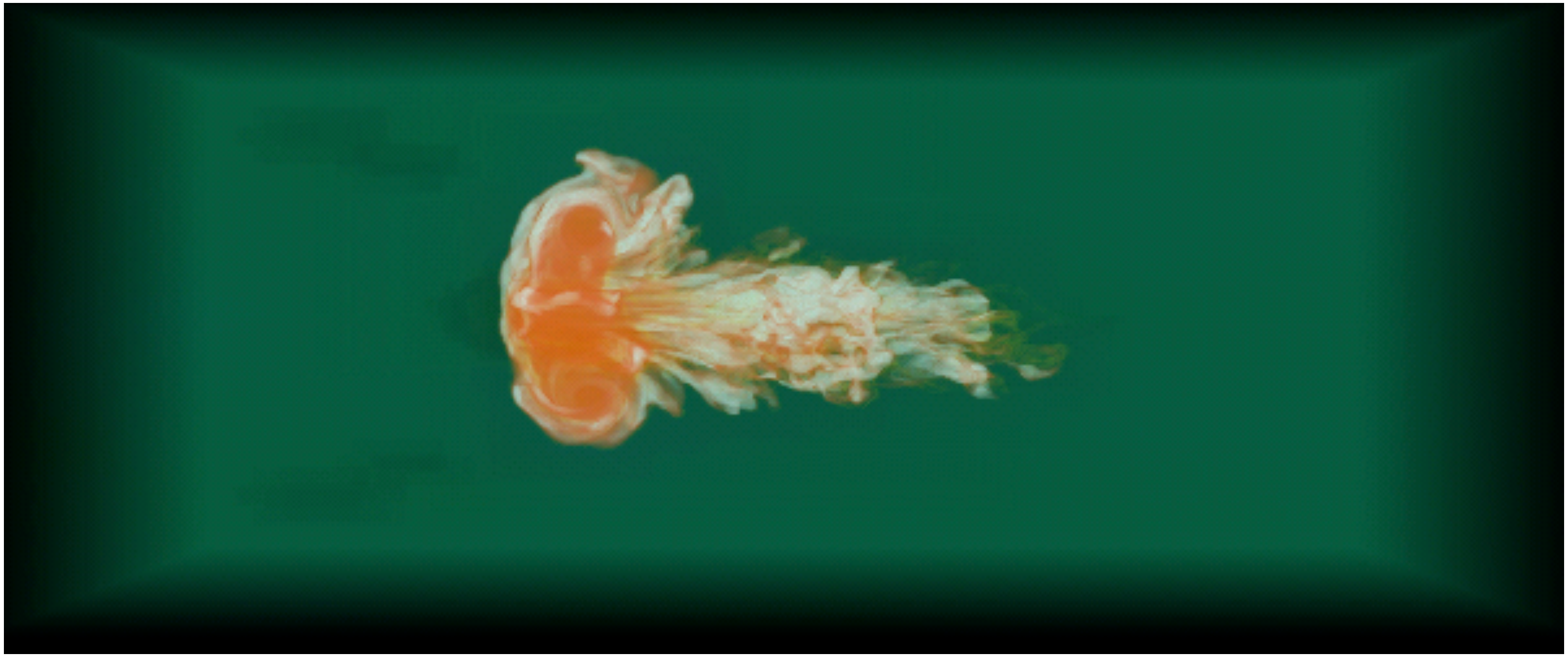
- [Park et al., 2002]
- kD-tree- and Octree-based domain subdivision
- Specify isovalue range and transfer function
- Rendering using hierarchical splatting





Direct Volume Rendering of AMR Data

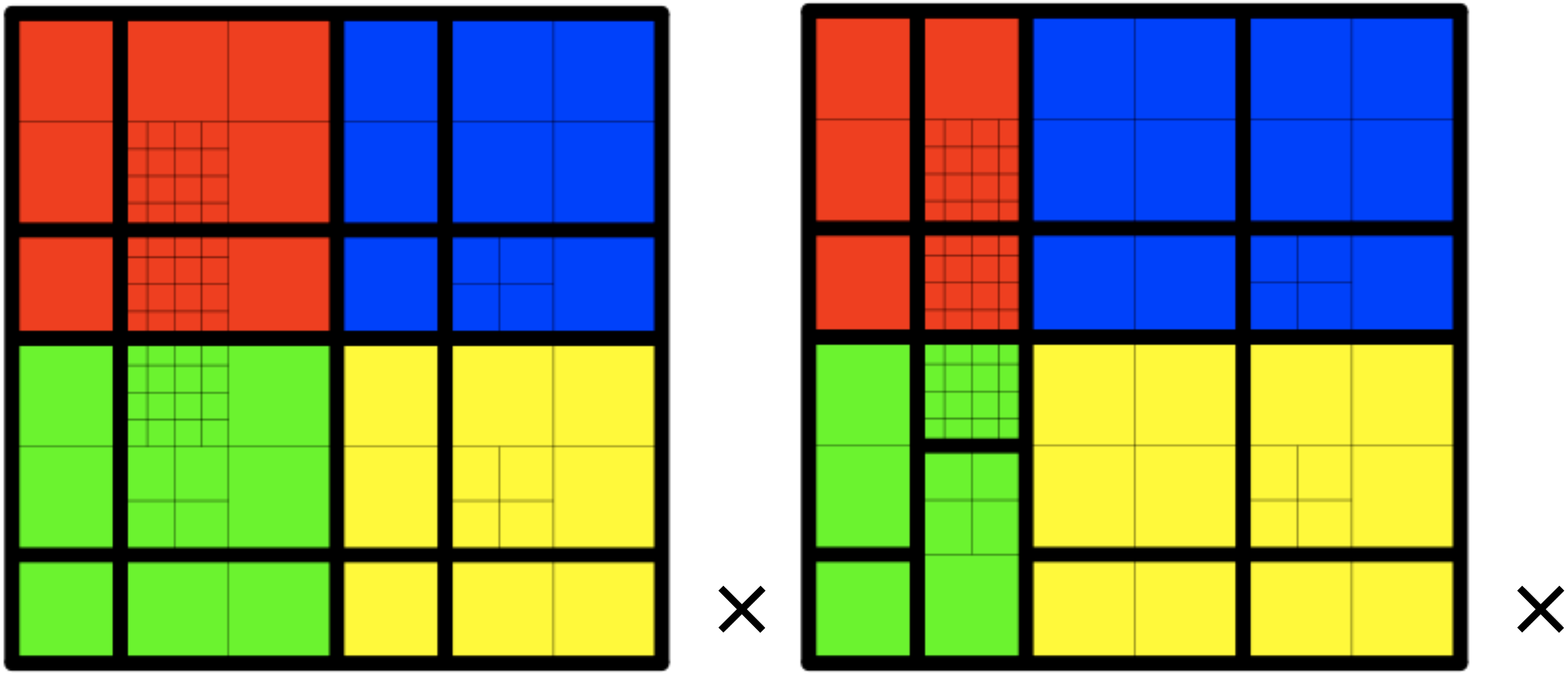
- [Kreylos et al., 2002]
- Homogenization using kD-tree
- Distributed rendering using texture-based slicing
- Cost-range decomposition



Framework for Parallel AMR Rendering

- Efficient reimplementation of cell projection
  - Sort cells [Williams, Max & Stein 1998]
- Subdivision in object space with kD-tree
- Subdivision of first hierarchy level
  - Uniform: Blocks of approximately equal size
  - Weighted: Blocks of similar computational effort
- Subdivision in blocks of constant resolution
  - Unweighted
  - Weighted

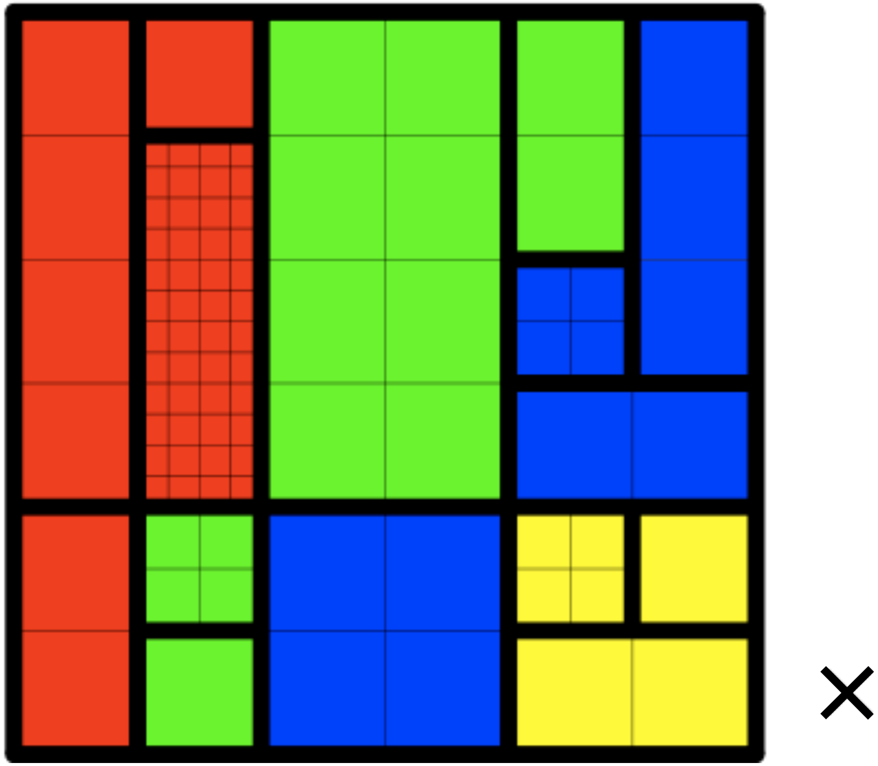
Subdivision Strategies – Subdivision of the First Hierarchy Level



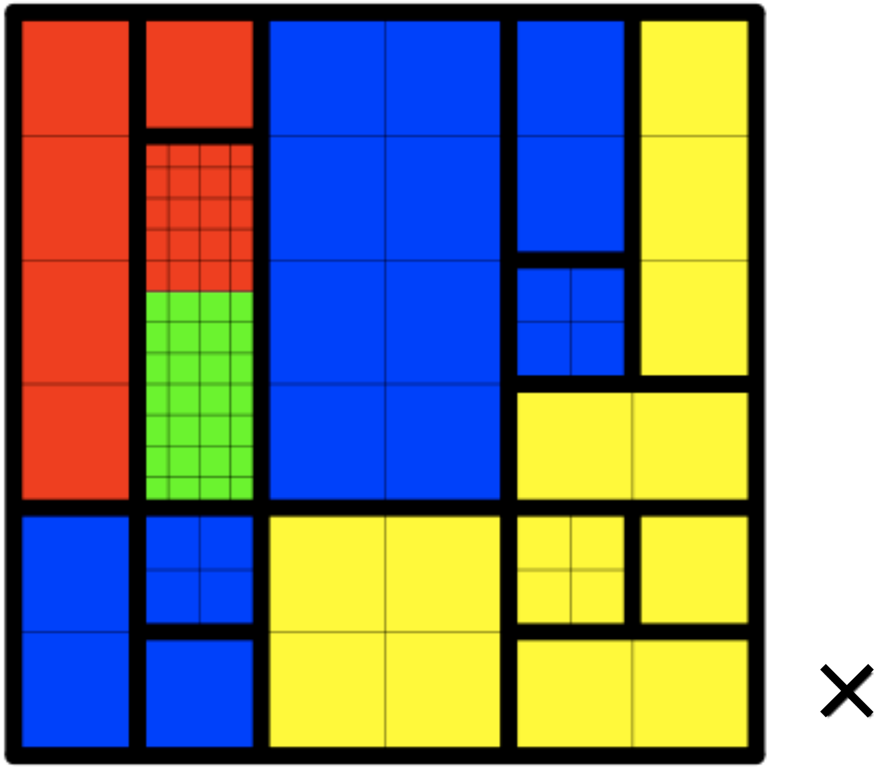
Uniform Weighted

X = Viewpoint; Color = Assigned processor

Subdivision Strategies – Homogenization



Unweighted

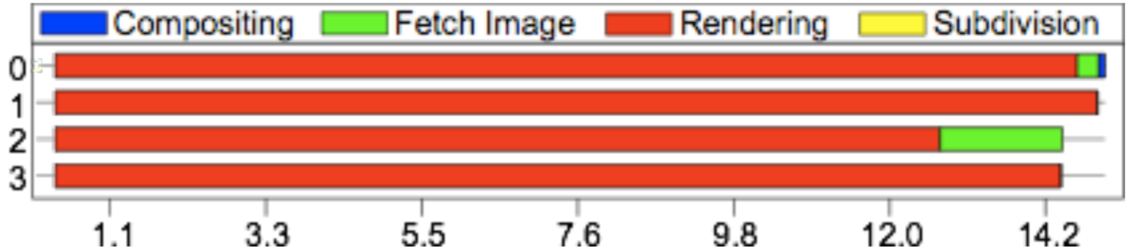


Weighted

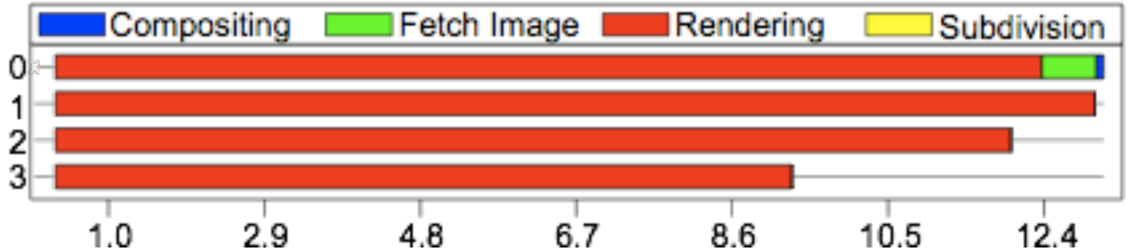
X = Viewpoint; Color = Assigned processor

Timing Results

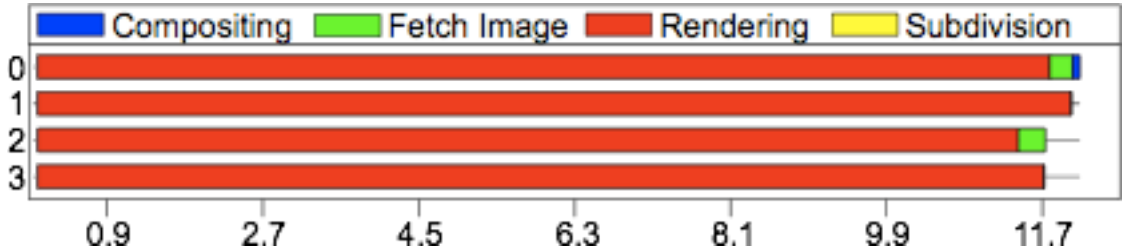
Uniform subdivision



Weighted subdivision of first hierarchy level



Homogeneous subdivision



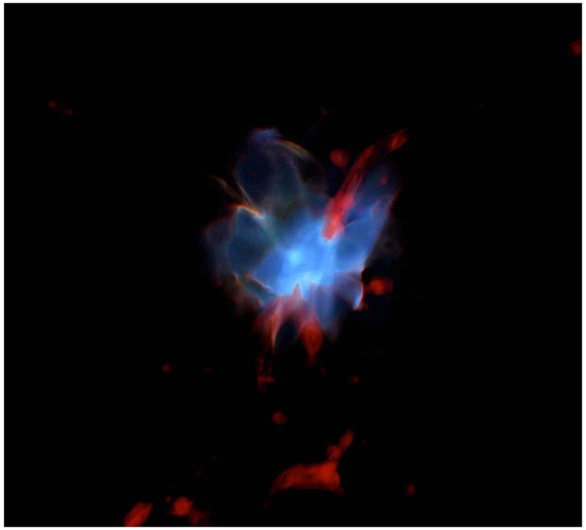
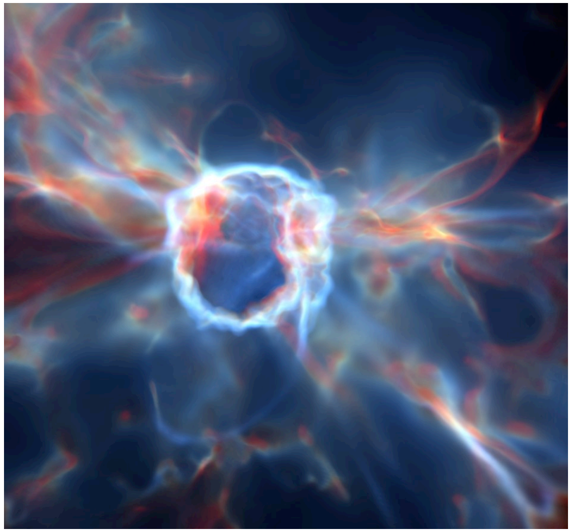
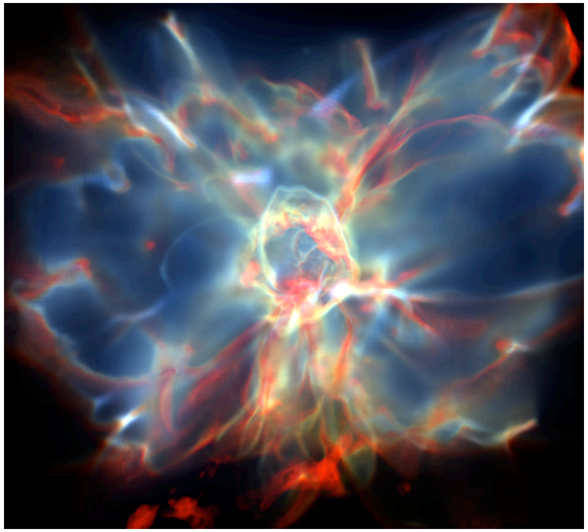
Weighted homogeneous subdivision

Observations

- Homogenization most efficient way to render AMR hierarchies
  - Computationally efficient
  - Use of standard methods
- Use of kD-tree currently standard way of describing subdivision
- Reasonable estimate of computational costs for rendering grid parts possible

GPU-Assisted Raycasting of AMR Data

- [Kähler et al., 2006]
- Use raycasting instead of texture slicing
- Higher quality (improved precision, avoid varying sample distances)
- Sophisticated light model with wavelength dependent absorption

Visualization of Time-varying AMR Data

- Feature-tracking
  - [Chen et al., 2003]
  - Isosurface visualization
  - Track connected components through time and AMR levels

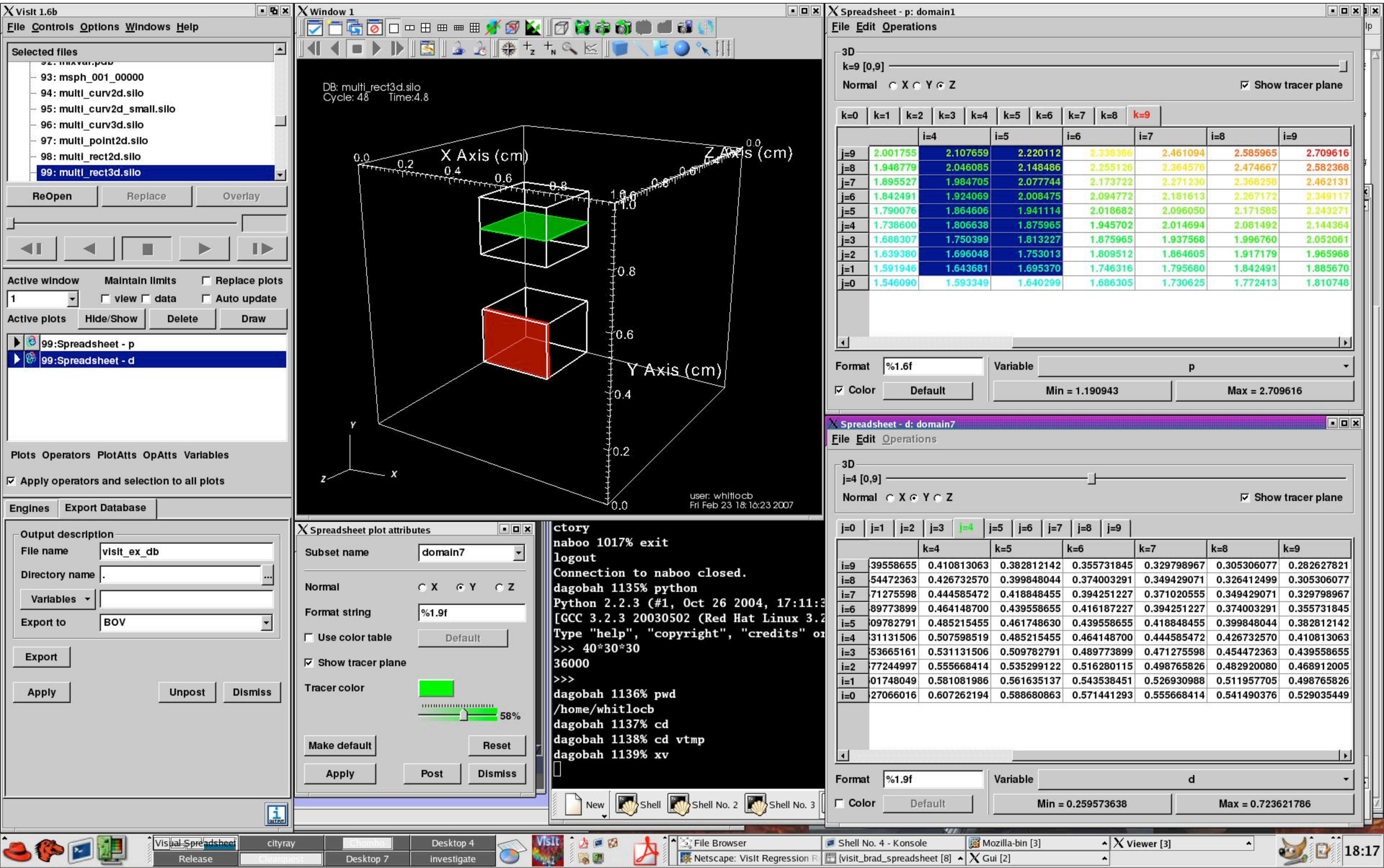
- Remote visualization of time-dependent AMR data
  - [Kähler et al., 2005]
  - Interpolation scheme for “in-betweening” of hierarchy levels evolving at different simulation rates
  - Access remote simulation over network

Specialized Tools for AMR Data

- **ChomboVis**
  - **LBNL Applied Numerical Algorithms Group**
  - **Slicing and *spreadsheets***
  - **Isosurfaces (w/ cracks)**
  - **Streamline computation (unpublished)**

- **AMR Vis**
  - **LBNL Center for Computational Sciences and Engineering**
  - **Shear-warp volume rendering (re-sampling)**
  - **Slicing and *spreadsheets***
  - **Streamlines**

Spreadsheets

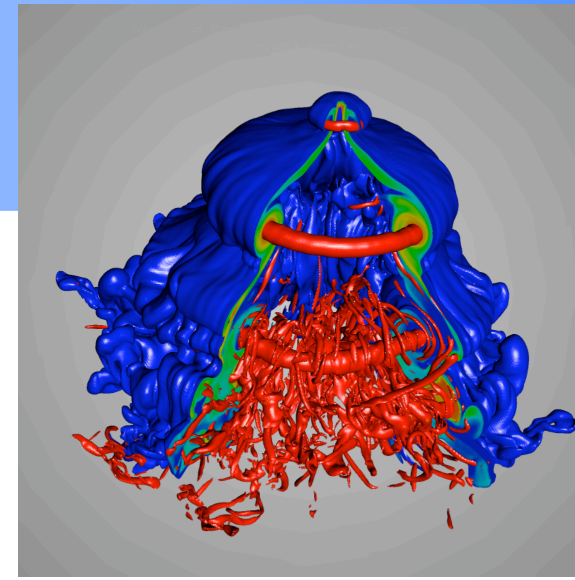


Visualization Tools with AMR Support

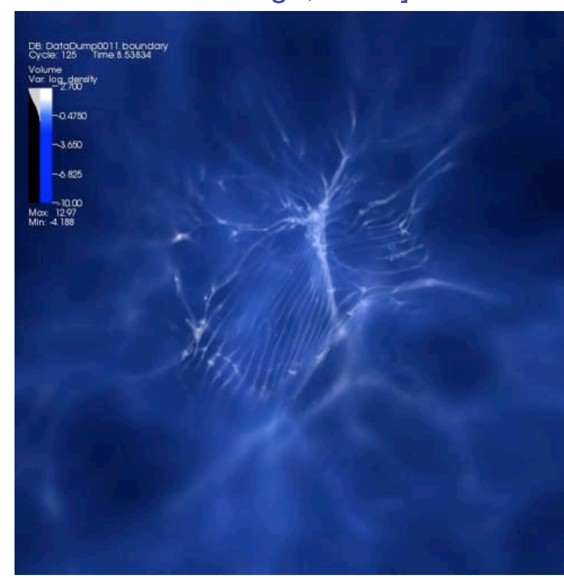
- ParaView
  - Support for reading AMR data sets (e.g., VTM)
  - Slicing, Isosurfaces (with cracks)
  - Volume rendering in development (commercial version)
- Amira
  - Some AMR support in internal collaboration version
  - Mainly volume rendering
- VisIt
  - Support for reading AMR data sets (e.g., Enzo, Boxlib, Chombo)
  - Wide range of visualizations including volume rendering, slices, isosurfaces (currently w/ cracks)

VISIT

- **Richly featured visualization and analysis tool for large data sets**
- **Data-parallel client server model, distribution on per patch-basis**
- **Use of meta-data / contracts to reduce amount of processed data**
- **Built for 5 use cases:**
  - **Data exploration**
  - **Visual debugging**
  - **Quantitative analysis**
  - **Presentation graphics**
  - **Comparative analysis**



[Argon bubble subjected to shock
Jeff Greenbough, LLNL]



[Logarithm of gas/dust density in Enzo star/galaxy simulation, Tom Abel & Matthew Turk, Kavli Institute]

VisIt and AMR Data

- Supported as "first-class" data type
- Handled via "ghost-cells": Coarse cells that are refined are marked "ghost" in the lower level
- Isocontouring via resampling, cracks possible at level boundaries
- Work on rectilinear grids and skip ghost cells or "remove" results produced in ghost cells later on

- AMR capabilities currently under rapid development (planned as ChomboVis replacement this FY)

- http://www.llnl.gov/visit

Acknowledgements

- Members of the NERSC Visualization Group, the Applied Numerical Algorithms Group (ANAG) and the Center for Computational Sciences and Engineering (CCSE) at LBNL
- Members of the VisIt Development Team
- Members of IDAV Visualization Group (UC Davis)
- AG Graphische Datenverarbeitung und Computergeometrie
- Members of ZIB and AEI

- Department of Energy (LBNL)
- National Science Foundation
- Office of Naval Research
- Army Research Office
- NASA Ames Research Center
- North Atlantic Treaty Organization
- ALSTOM Schilling Robotics, Chevron, General Atomics, Silicon Graphics, and ST Microelectronics, Inc.
- Stiftung für Innovation des Landes Rheinland-Pfalz

Questions?