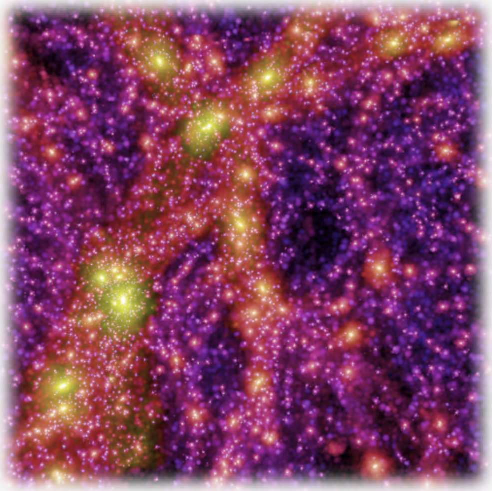
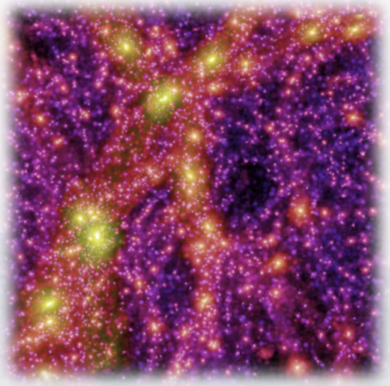


Exploring the Millennium Run

Scalable Rendering of Large-Scale Cosmological Datasets

Roland Fraedrich, Jens Schneider, Rüdiger Westermann
Technische Universität München

Millennium Run



- Goal
  - Simulation of the evolution of the Universe (ΛCDM-Model)
- Setup
  - N-body/SPH simulation with $2160^3 > 10^{10}$ particles
  - Cubic region of 2.23 billion light-years per side
  - Effective resolution of $100{,}000^3$
  - 28 machine days at 0.2 Tflops using 1 TB RAM
- Output
  - Per snapshots: Position + attributes of all particles (225 GB)

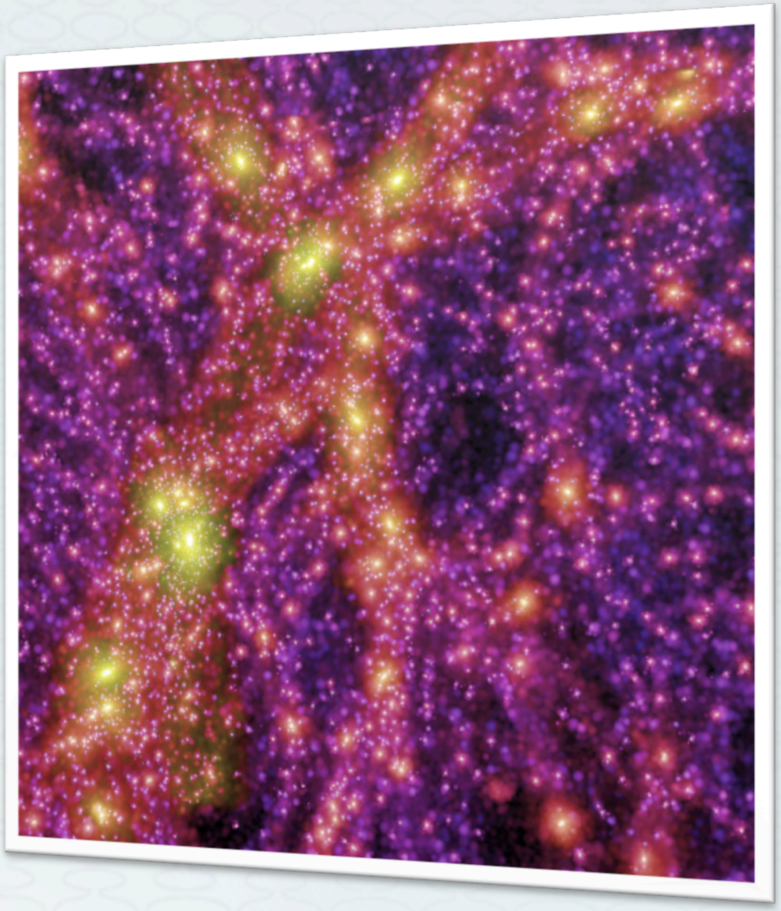
Vis09

http://www.mpa-garching.mpg.de/galform/virgo/millennium/

tum.3D computer graphics & visualization

Contribution

- Scalable rendering approach
  - Datasets > 10 billion particles
  - Data volume > 200 GB
  - Interactive frame rates
  - Common PC hardware
  - Screen space error < 1 pixel

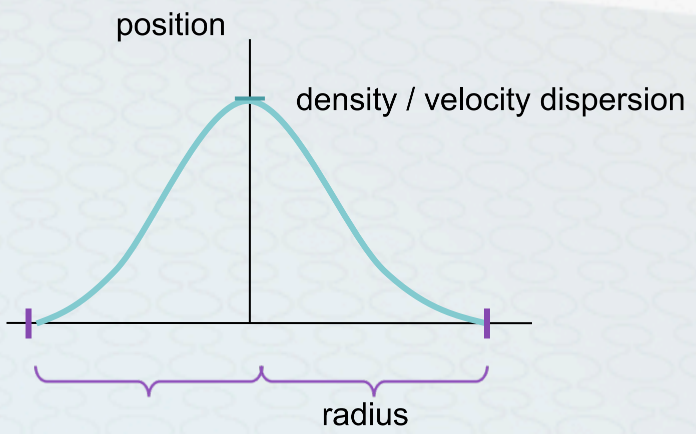


Major Constraints for Rendering

- Limited Geometry Throughput
- Memory Limitations (GPU)
- RAM → GPU Transfer
- Memory Limitations (RAM)
- HD → RAM Transfer
- Disc Access Latency

Visualization Approach

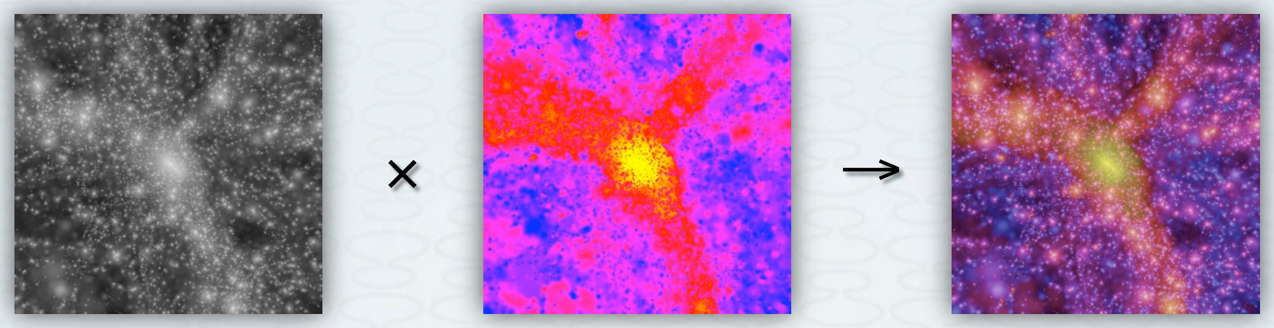
- SPH Particle Properties
  - Position (float3), Radius (float)
  - Density (float), Velocity Dispersion (float)
- Common Visualization Technique
  - Order-independent integration along the line of sight
    - Squared density ρ^2
    - Velocity dispersion weighted with ρ^2



computer graphics & visualization

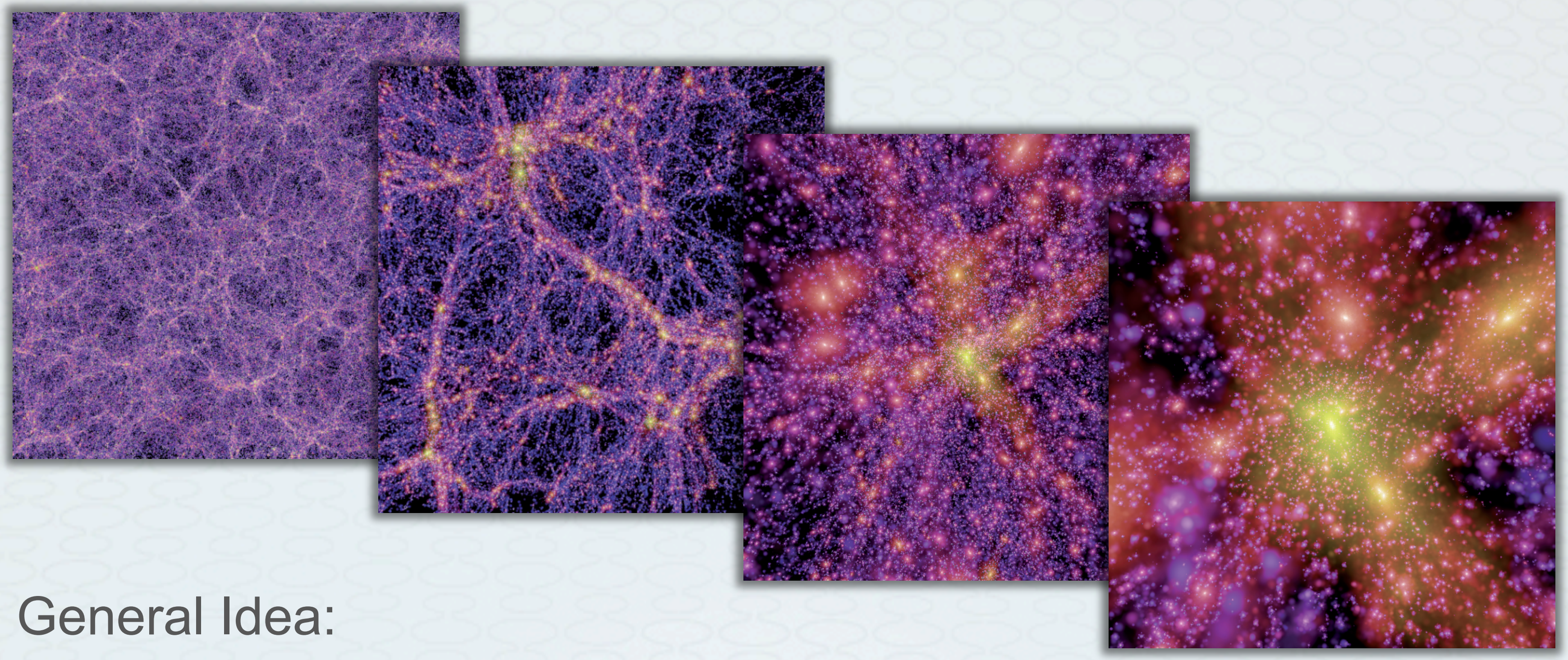
Visualization Approach

- Common Visualization Technique
  - Order-independent integration along the line of sight
    - Squared density ρ^2
    - Velocity dispersion weighted with ρ^2
  - Color Coding
    - Brightness: Logarithm of ρ^2
    - Hue: Logarithm of velocity dispersion weighted with ρ^2



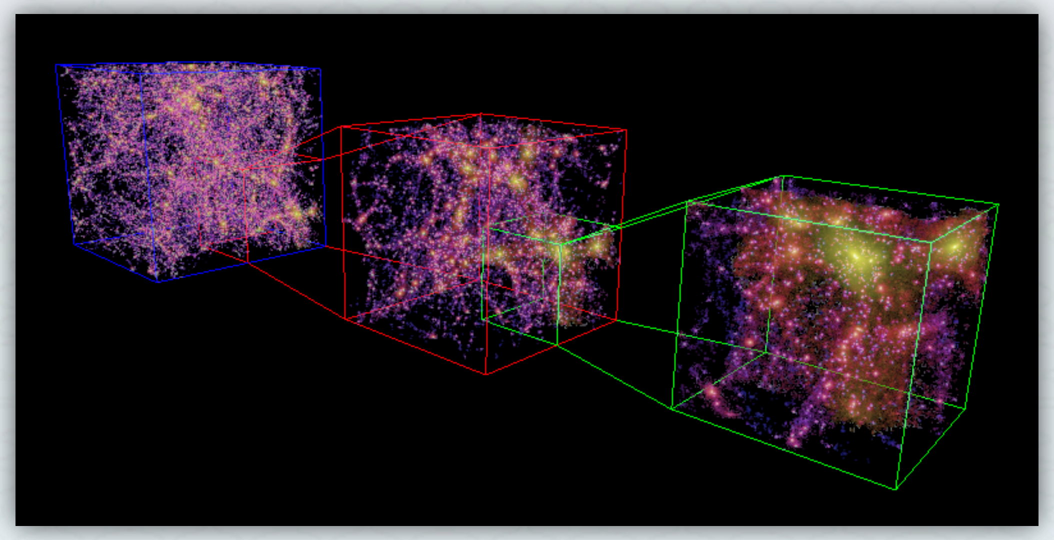
tum.3D computer graphics & visualization

Data Representation



- General Idea:
  - Hierarchical data representation
  - Minimize number of particles for a given subpixel error for any view

Data Representation

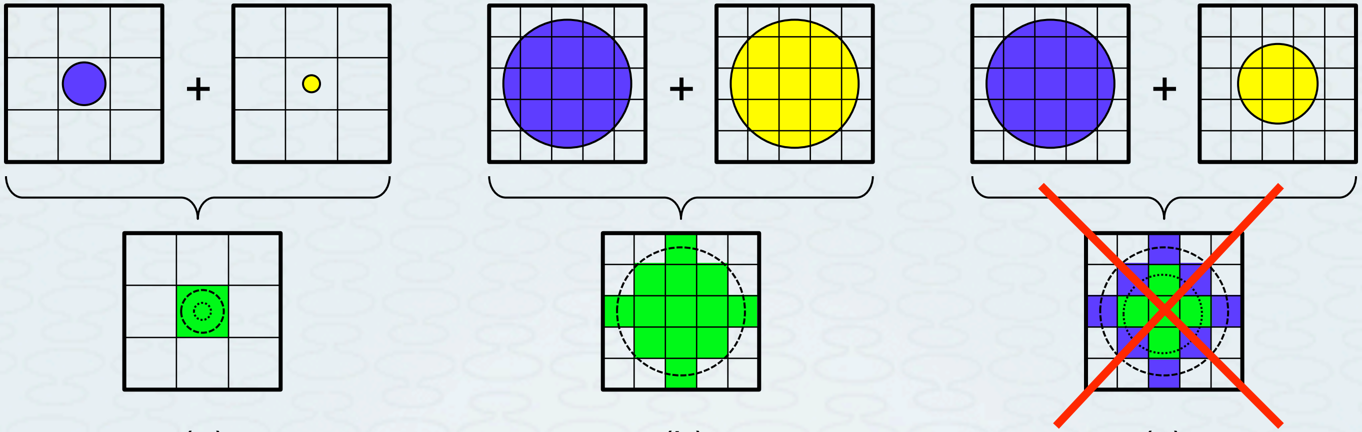


- Multi-Resolution Hierarchy
  - Octree-based data structure
  - Subdomains are discretized into regular grids of size 8^3
  - Refinement down to $128K^3$

tum.3D computer graphics & visualization

Particle Thinning

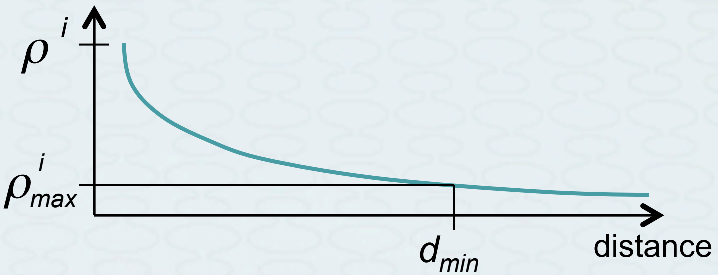
- Merging of particles within a quantization bin
  - (a) Particles with a diameter < 1 pixel (→ single point primitive)
  - (b) Particles with the same radius (→ single point sprite)
  - (c) All other particles remain separated



(a) (b) (c)

Particle Thinning

- General Idea:
  - Discard particles having no influence on final image
- How to measure influence on screen?
  - Density of particle ρ^i
  - Inverse-square law
  - d_{min} of octree node

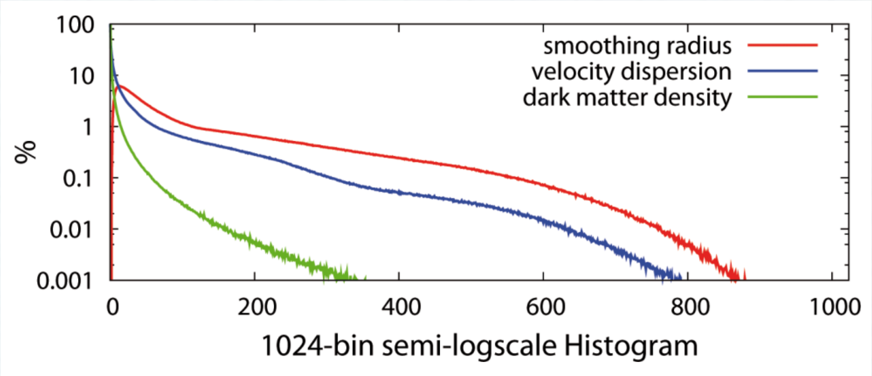


- If $\rho^i_{max} < P_{\min}$ of color table → Particle not visible
  - <u>However:</u> Integration of *n* of such particles can be visible

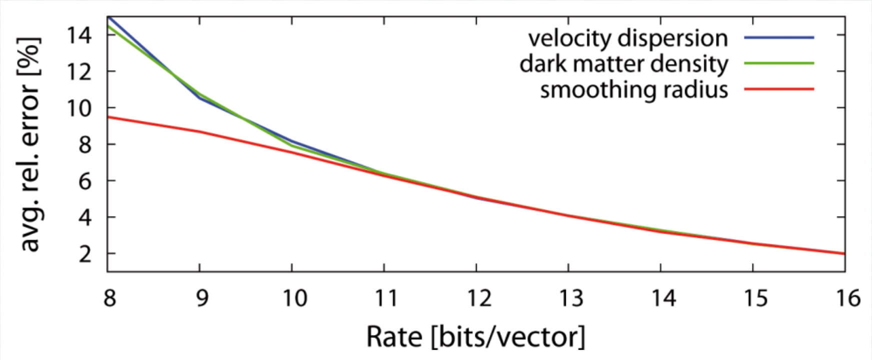
    → adopt rule to $\rho^i_{max} < P_{min} / n$ to discard particles.

Attribute Compression

- Additional particle quantities (radius, temperature, density)
  - High range of values
  - Stochastically dependent



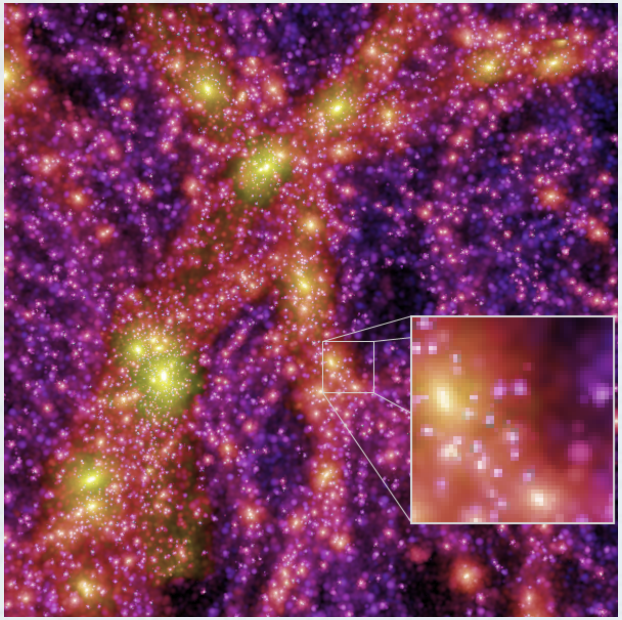
- Vector quantization (Logarithmically scaled to minimize relative error)
  - 8 bit (12 : 1): error ≈ 13 %
  - 12 bit (8 : 1): error ≈ 5 %
  - 16 bit (6 : 1): error ≈ 2 %



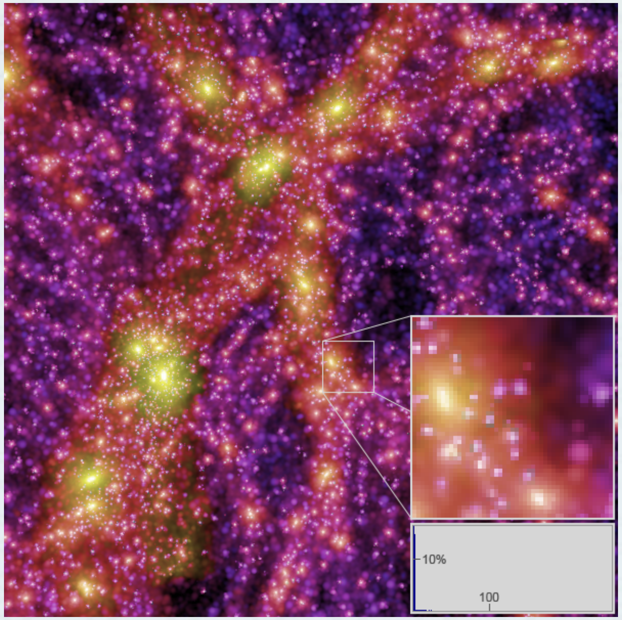
VIS09

tum.3D computer graphics & visualization

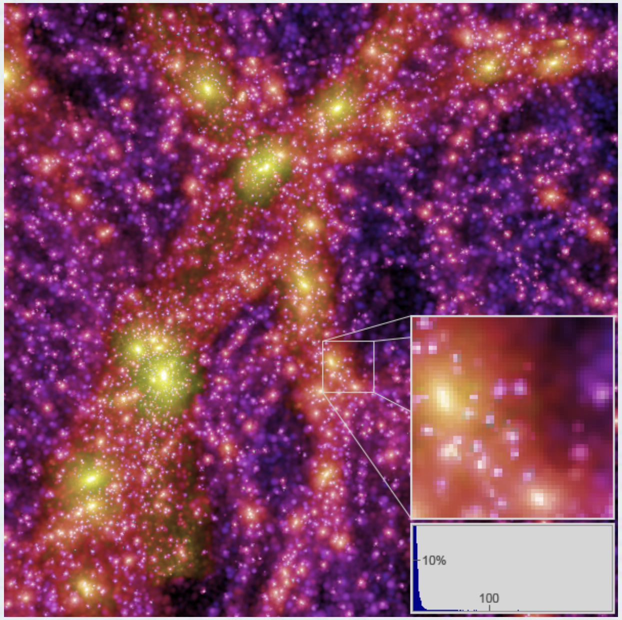
Attribute Compression



Original



16 bit VQ (6:1)
Avg. Error: 0.2*



8 bit VQ (12:1)
Avg. Error: 2.0*

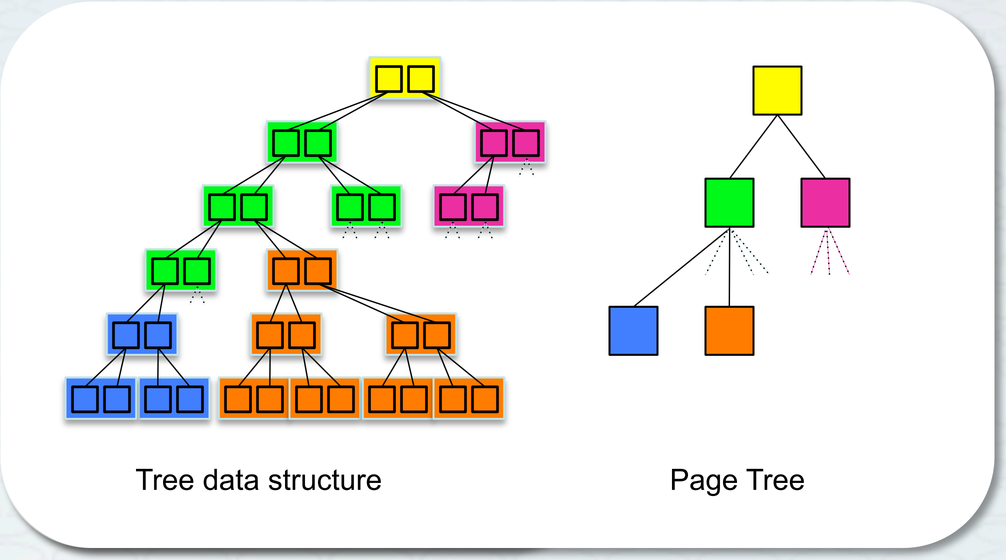
**Error measured as Euclidean distance in RGB color space r,g,b ∈ [0,255] for multiple views.*

computer graphics & visualization

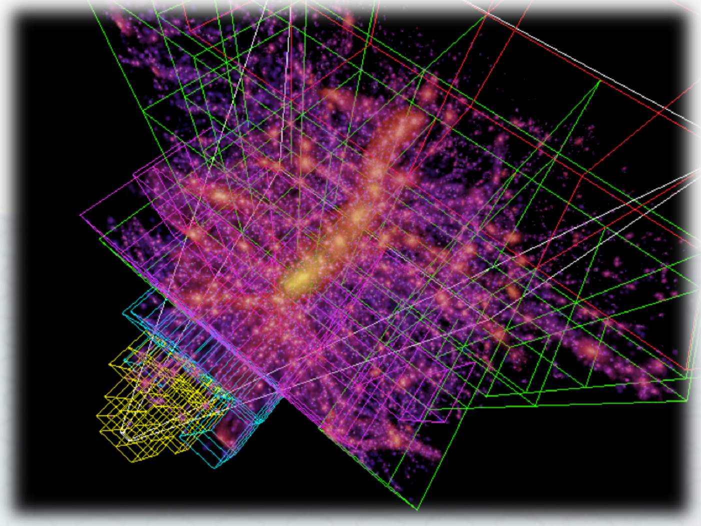
Data Management

- General Idea:
  - Increase overall data throughput by reducing number of disk seek operations
- Collect data of nodes into pages:
  - Siblings are stored together
  - If pageSize < 3 MB

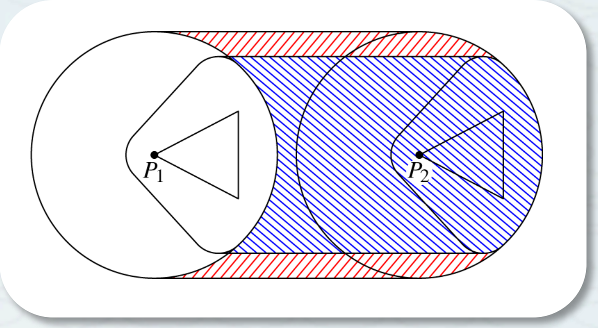
    → Fill pages breadth-first



Rendering



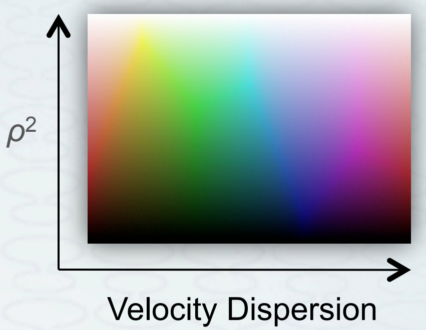
- Node selection (CPU)
  - Tree traversal (top down)
  - View-frustum culling
  - LOD-selection based on user-defined subpixel screen space error
- Loading of pages if necessary (asynchronous I/O)
- Sphere-shaped pre-fetching region
- Buddy-system on GPU
- LRU-Cache on GPU and CPU



computer graphics & visualization

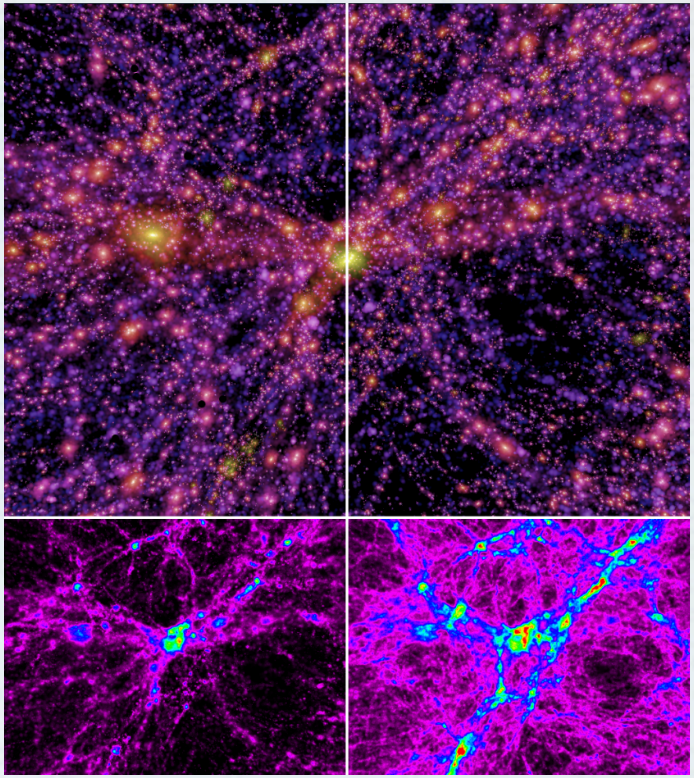
Rendering

- 1^{st} Pass: Integration along viewing rays
  - Order-independent Rendering (BlendAdd)
    - Point Primitives
    - Point Sprites
  - Result
    - Integrals of ρ^2 and weighted velocity dispersion
    - Stored in a 2D render target
- 2^{nd} Pass: Color Coding
  - Single texture fetch into color table



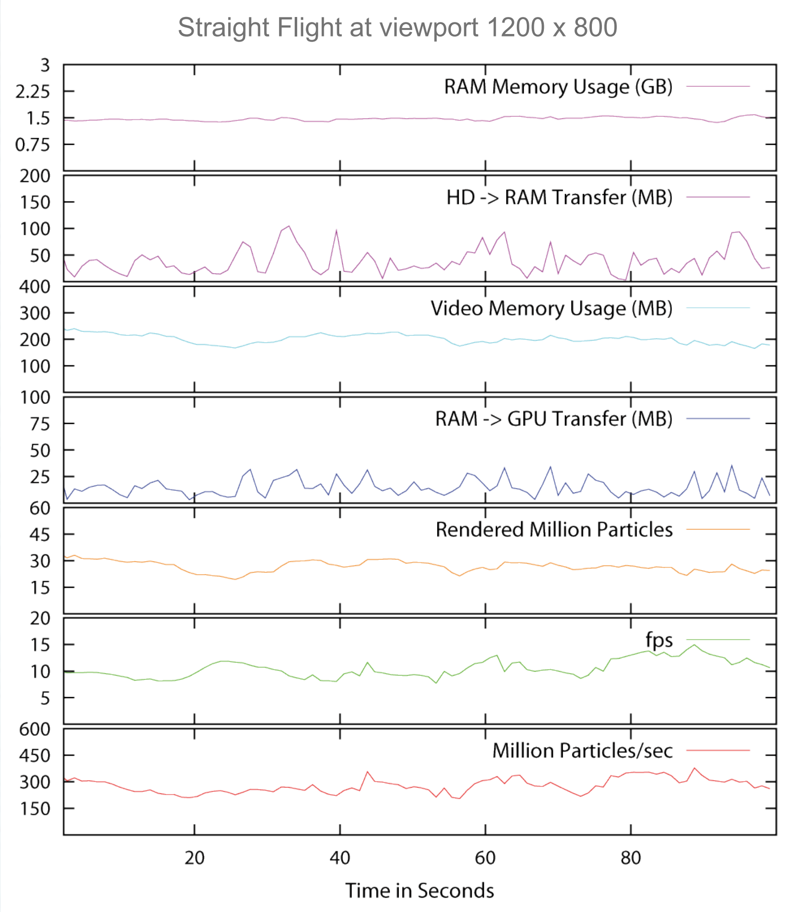
Rendering

- Bottleneck:
  - Heavy geometry and rasterization load
- Solution:
  - Discard particles from the rendering pipeline as early as possible
    - Fine-grain view-frustum culling
    - Density-based culling



10.4 fps 4.4 fps

Results



→ RAM Usage ≈ 1.5 GB

→ Disk Transfer < 110 MB/s

→ Video Memory < 250 MB

→ GPU Upload < 35 MB/s

→ Rendered Particles ≈ 30 M

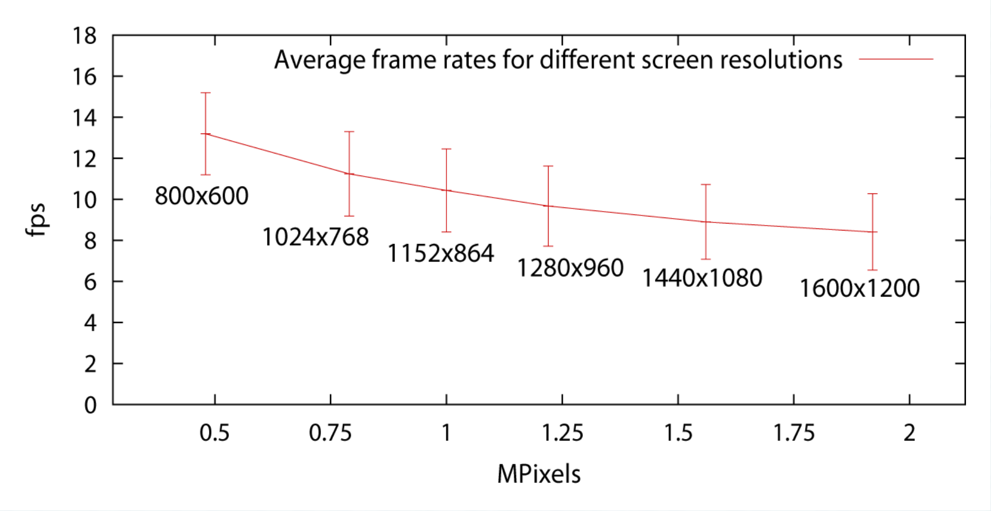
→ Rendering Speed ≈ 11 fps

→ Throughput ≈ 280 MP/s

PC:
- Intel Core 2 Quad 2.66 GHz
- RAM: 4 GB
- NVIDIA GeForce GTX 280 (1024 MB)
- PCI Express 2.0 x16

Results

- Scalability with regard to the screen resolution



Summary

- Interactive visualization of cosmological particle data
  - Achieving 13 fps for datasets exceeding 10 billion particles
  - Scalable to much larger datasets
- Components
  - Hierarchical data representation
    - Significant reduction of particle data at no visual loss
  - Efficient out-of-core and in-core memory management
    - Effectively hiding unavoidable data streaming
  - High performance rendering
    - To deal with the remaining load of particles

Future Work

- Bottleneck
  - High geometry and rasterization load
- Correct Volume Rendering
  - Ray Tracing using volume rendering integral
  - „Sorting“ of particles necessary
- Extension to time-dimensional data
  - Millennium Run: 20 TB
  - Requires new data representation