EUROVIS Leipzig 2023

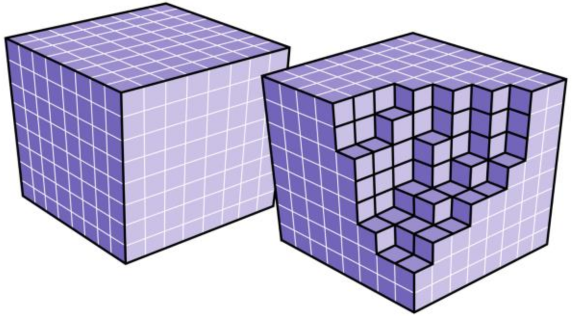
State-of-the-art in Large-Scale Volume Visualization Beyond Structured Data

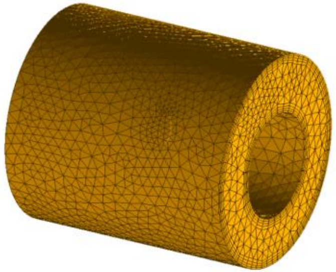
J. Sarton[1], S. Zellmann[2], S. Demirci[3], U. Güdükbay[3], W. Alexandre-Barff[4], L. Lucas[4], J.M. Dischler[1], S. Wesner[2], I. Wald[5]

[1] University of Strasbourg, France
[2] University of Cologne, Germany
[3] Bilkent University, Ankara, Turkey
[4] University of Reims Champagne-Ardenne, France
[5] NVIDIA

Context and motivation

Volume data produced by acquisition or simulation





Scientific visualization to: analyze datasets, extract information, guide phenomenon modeling, validate or invalidate models, evaluate experimental results, …

Volume rendering techniques: used to produce 2D images from this data

Context and motivation

EUROVIS Leipzig 2023

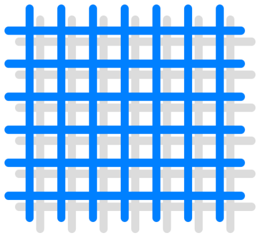
Volume data produced by acquisition or simulation

-> Usually massive: Large-scale volume data

Scientific visualization **Memory subsystem are the primary bottleneck**

Large-scale volume rendering techniques to cope with the input data and the auxiliary data structures needed to render them

Context and motivation



<u>Direct volume rendering</u>: historically associated with **regular grids** for medical imaging, microscopy, …

Simple representation (implicit topology)

Regular representation **adapted to GPU** architecture and texture units

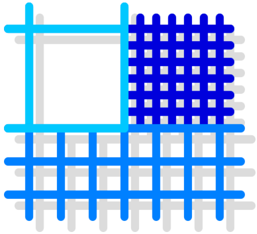
High resolution → **huge datasets** (up to 10^{12} voxels)

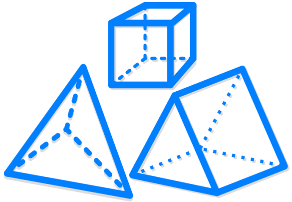
Context and motivation

EUROVIS Leipzig 2023

Better adapt to the domain of large-scale simulations:

- **hierarchical** representations
- **unstructured** volume data





→ reduce memory consumption

But

- these representations need **large and complex structures** for efficient interactive visualization
- large-scale simulations produced: time series, multi-variate volume data…

Survey scope

EUROVIS Leipzig 2023

Large-scale direct volume rendering techniques

- **ray tracing** techniques
- **sampling** and **traversal** methods of visualization algorithms
- **in-core** or **out-of-core** methods
- **single** or **multi CPU/GPU** approaches

And hardware-accelerated ray tracing

Not discuss: in-situ visualization solutions

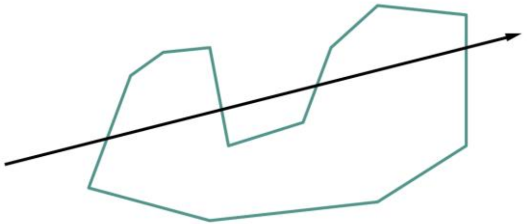
Background

EUROVIS Leipzig 2023

Volume rendering with ray-traversal

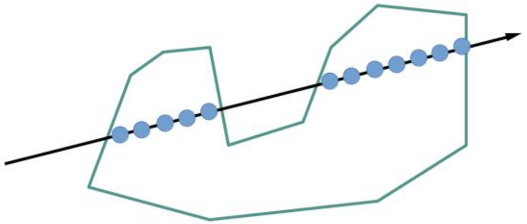
- **Traversal**

  obtaining the domains [tmin ,tmax]



- **Sampling**

  volume lookup and classification (with a Transfer Function)



Background

Hardware acceleration

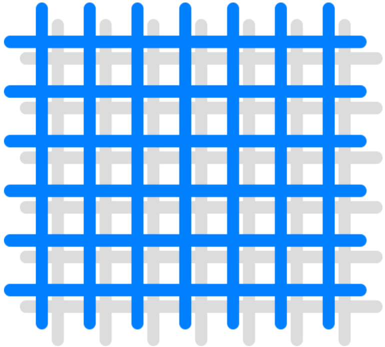
Specific hardware operations for ray-tracing

- BVH traversal
- Ray/triangle intersections

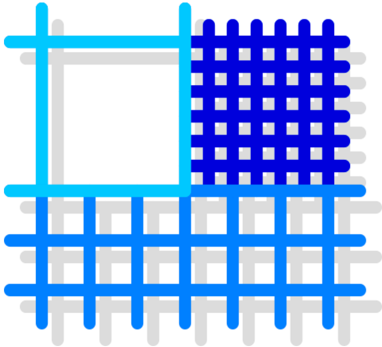
Emerging technology relevant to many of the papers discussed

Data driven structure

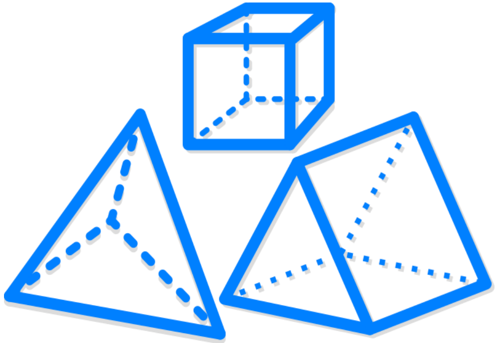
EUROVIS Leipzig 2023



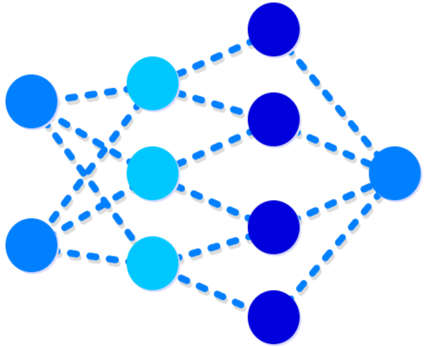
Structured volume data



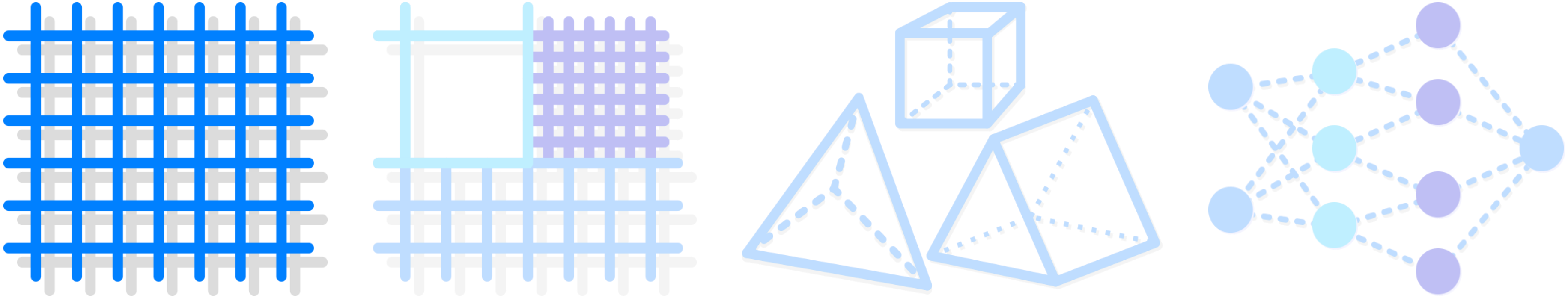
Adaptive Mesh Refinement volume data



Unstructured volume data



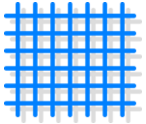
Compressed and neural representations



Structured Volume Data

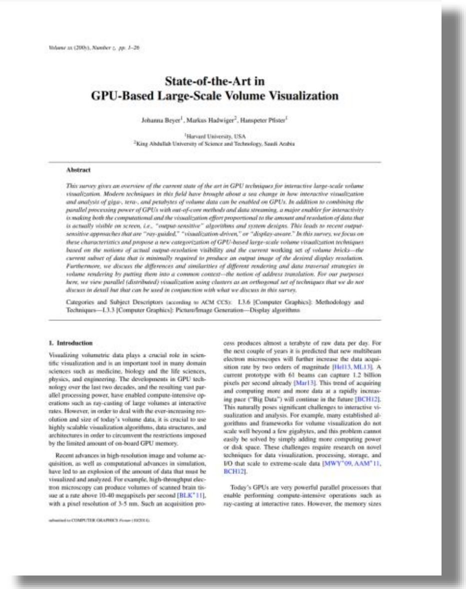
Background

Cartesian grids



GPU techniques for interactive large-scale (structured) volume visualization

Beyer et al. State-of-the-art in GPU-based large-scale volume visualization, Computer Graphics Forum (2015)



Volume xx (200y), Number z, pp. 1–26

State-of-the-Art in GPU-Based Large-Scale Volume Visualization

Johanna Beyer[1], Markus Hadwiger[2], Hanspeter Pfister[1]

[1]Harvard University, USA
[2]King Abdullah University of Science and Technology, Saudi Arabia

Abstract

This survey gives an overview of the current state of the art in GPU techniques for interactive large-scale volume visualization. Modern techniques in this field have brought about a sea change in how interactive visualization and analysis of giga-, tera-, and petabytes of volume data can be enabled on GPUs. In addition to combining the parallel processing power of GPUs with out-of-core methods and data streaming, a major enabler for interactivity is making both the computational and the visualization effort proportional to the amount and resolution of data that is actually visible on screen, i.e., "output-sensitive" algorithms and system designs. This leads to recent output-sensitive approaches that are "ray-guided," "visualization-driven," or "display-aware." In this survey, we focus on these characteristics and propose a new categorization of GPU-based large-scale volume visualization techniques based on the notions of actual output-resolution visibility and the current working set of volume bricks—the current subset of data that is minimally required to produce an output image of the desired display resolution. Furthermore, we discuss the differences and similarities of different rendering and data traversal strategies in volume rendering by putting them into a common context—the notion of address translation. For our purposes here, we view parallel (distributed) visualization using clusters as an orthogonal set of techniques that we do not discuss in detail but that can be used in conjunction with what we discuss in this survey.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques—I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms

1. Introduction

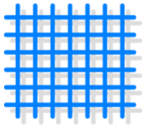
Visualizing volumetric data plays a crucial role in scientific visualization and is an important tool in many domain sciences such as medicine, biology and the life sciences, physics, and engineering. The developments in GPU technology over the last two decades, and the resulting vast parallel processing power, have enabled compute-intensive operations such as ray-casting of large volumes at interactive rates. However, in order to deal with the ever-increasing resolution and size of today's volume data, it is crucial to use highly scalable visualization algorithms, data structures, and architectures in order to circumvent the restrictions imposed by the limited amount of on-board GPU memory.

Recent advances in high-resolution image and volume acquisition, as well as computational advances in simulation, have led to an explosion of the amount of data that must be visualized and analyzed. For example, high-throughput electron microscopy can produce volumes of scanned brain tissue at a rate above 10-40 megapixels per second [BLK*11], with a pixel resolution of 3-5 nm. Such an acquisition process produces almost a terabyte of raw data per day. For the next couple of years it is predicted that new multibeam electron microscopes will further increase the data acquisition rate by two orders of magnitude [Hel13, ML13]. A current prototype with 61 beams can capture 1.2 billion pixels per second already [Mar13]. This trend of acquiring and computing more and more data at a rapidly increasing pace ("Big Data") will continue in the future [BCH12]. This naturally poses significant challenges to interactive visualization and analysis. For example, many established algorithms and frameworks for volume visualization do not scale well beyond a few gigabytes, and this problem cannot easily be solved by simply adding more computing power or disk space. These challenges require research on novel techniques for data visualization, processing, storage, and I/O that scale to extreme-scale data [MWY*09, AAM*11, BCH12].

Today's GPUs are very powerful parallel processors that enable performing compute-intensive operations such as ray-casting at interactive rates. However, the memory sizes

Volume rendering of regular grids and mostly review methods for scalar data and a single time step

Background



Structured Volume Data

Recent works focus on:

- Reduce pressure on the memory subsystem by reducing the number of samples: **Empty space skipping data structures**

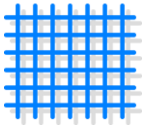
- Stream volume data that does not fully fit into the node’s main memory: **Paging and out-of-core approaches**

Empty space-skipping

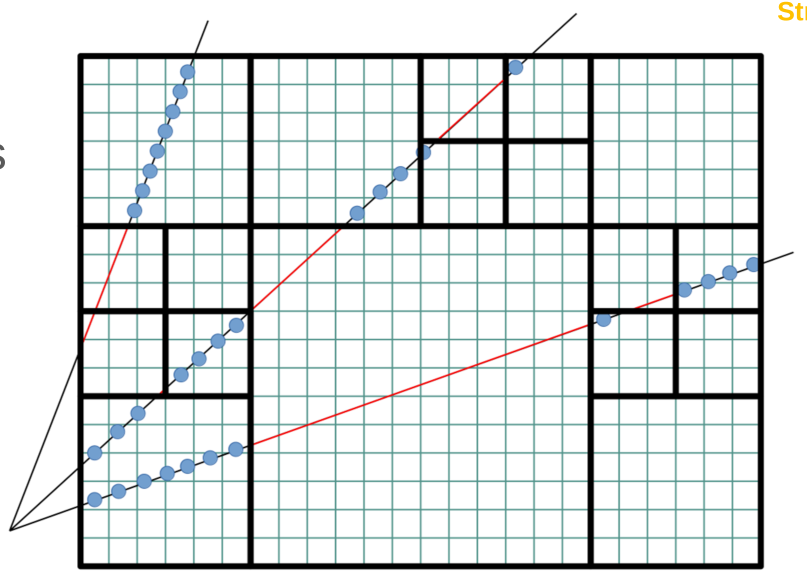
EUROVIS Leipzig 2023

Grid and **octree**-based methods

min/max trees from coarser regions of space : **octree leaves**



Structured Volume Data



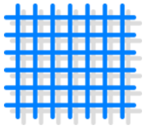
Hadwiger et al. ***SparseLeap: Efficient empty space skipping for large-scale volume rendering*****, IEEE TVCG (2018)**

Empty space skipping

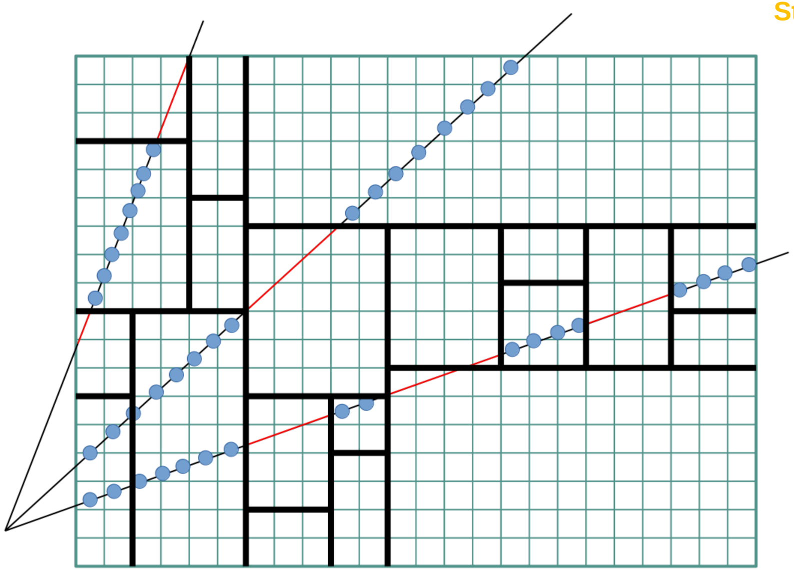
EUROVIS Leipzig 2023

High-quality k-d trees

build an exact data structure according to TF updates



Structured Volume Data

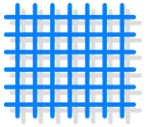


Vidal et al. *Simple empty-space removal for interactive volume rendering*, Journal of Graphics Tools (2008)

Zellmann et al. *Rapid k-d tree construction for sparse volume data*, EGPGV (2018)

Zellmann et al. *Binned k-d tree construction for sparse volume data on multi-core and GPU systems*, IEEE TVCG (2021)

Empty space skipping



Structured Volume Data

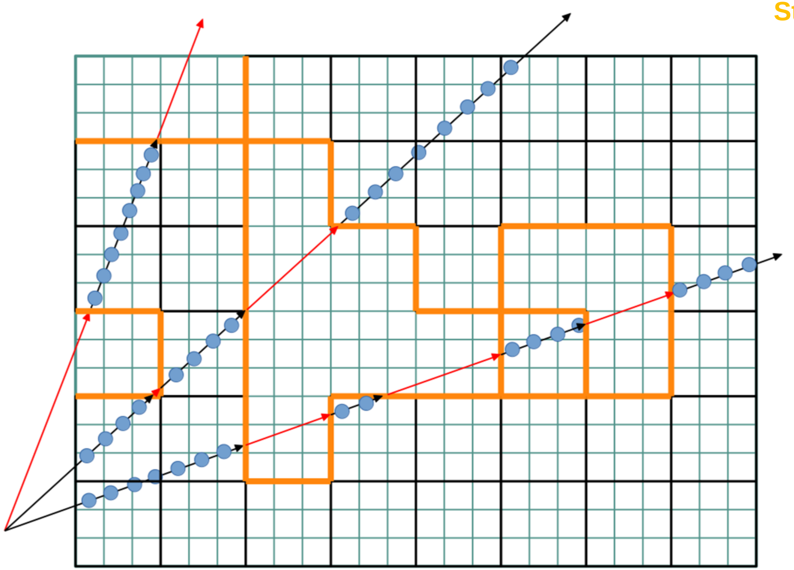
Hardware accelerated ray-tracing based space skipping

BVH traversal

Ganter and Manzke, *An analysis of region clustered BVH volume rendering on GPU*, Computer Graphics Forum (2019)

BVH traversal + ray/triangle intersection

Wald et al. *Faster RTX-accelerated empty space skipping using triangulated active region boundary geometry*, EGPGV (2021)



Paging and out-of-core

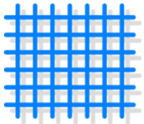
Structured Volume Data

When volume data does not fully fit into the node's main memory

Pageable memory to stream data to the GPU or asynchronously fetch the data from disk

- Out-of-core volume rendering on multi-core CPU architectures
- GPU out-of-core on-demand rendering and processing

Paging and out-of-core



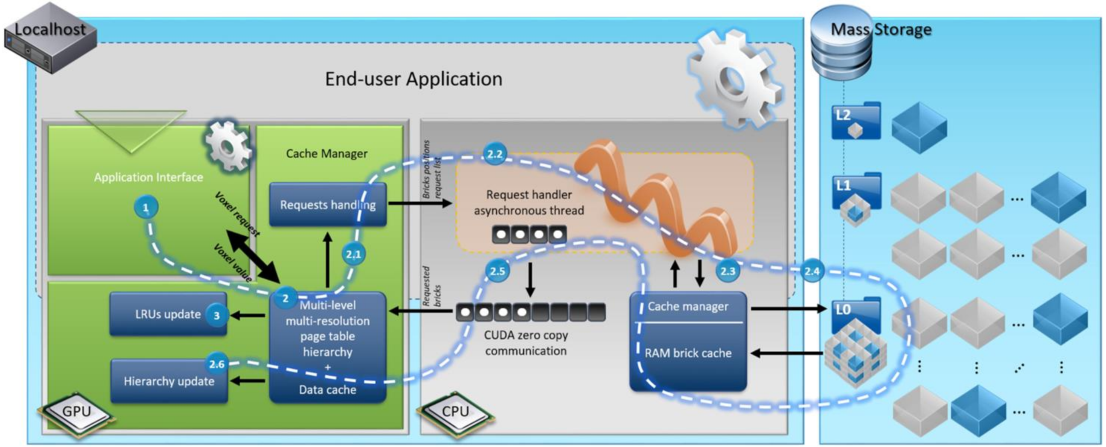
Structured Volume Data

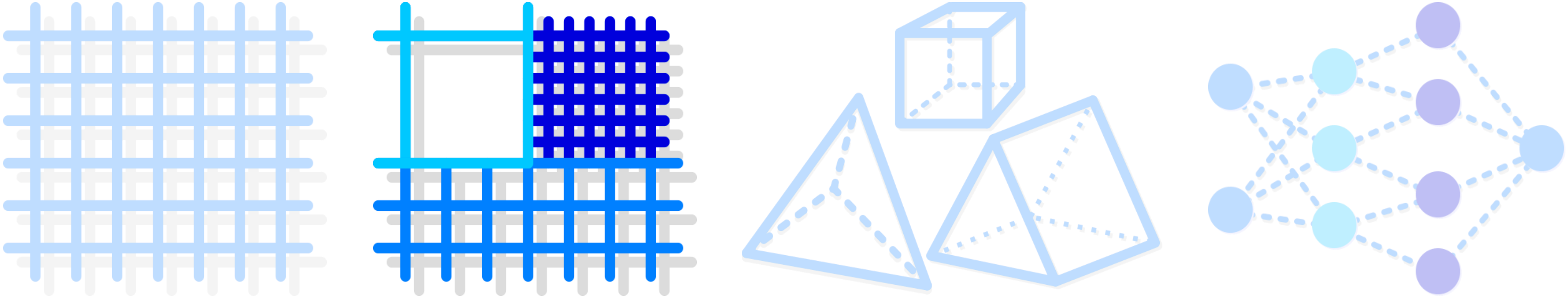
GPU out-of-core on-demand rendering and processing

Multi-resolution virtual addressing structure fully managed on GPU

Asynchronous and demand-driven system can scale well for both visualization and image processing applications

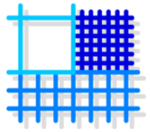
Sarton et al. *Interactive visualization and on-demand processing of large volume data: A fully GPU-based out-of-core approach*, IEEE TVCG (2020)





Adaptive Mesh Refinement Volume Data

Background



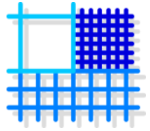
Adaptive Mesh Refinement Volume Data

AMR data: terminology & definitions

- Different flavors of AMR. State-art focuses on type where subgrids are Cartesian
- Each cell stores its refinement level $l=\{0,1,...\}$
- Assume that cells are voxels (unit length)
- Refinement level determines voxel size: $s=2^l$
- Blockstructured AMR, Octree AMR, wide Octrees, etc.

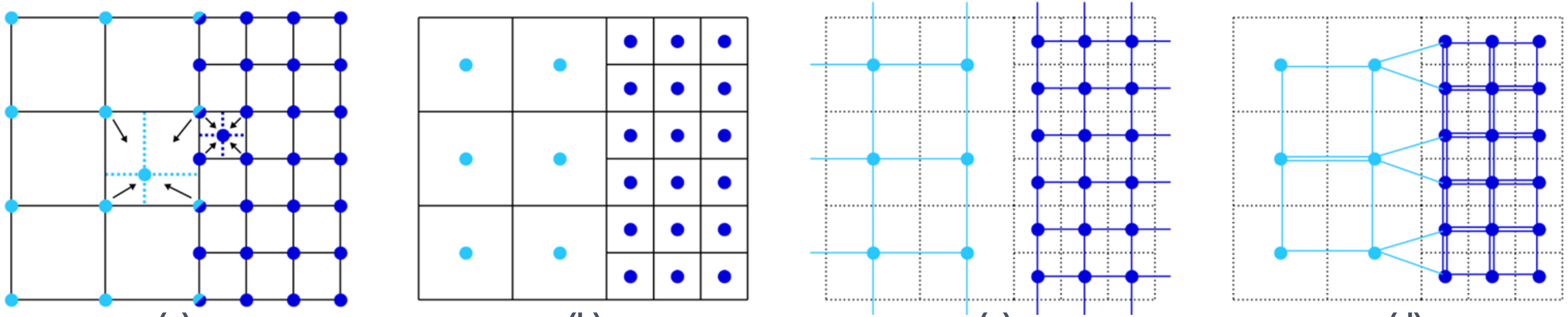
Background

EUROVIS
Leipzig 2023



Adaptive Mesh Refinement Volume Data

Motivating problem



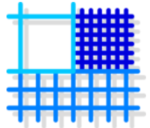
(a) Vertex-centered AMR: Simple reconstruction with *trilerp*

(b) Cell-centered AMR

(c) T-Junction Boundaries

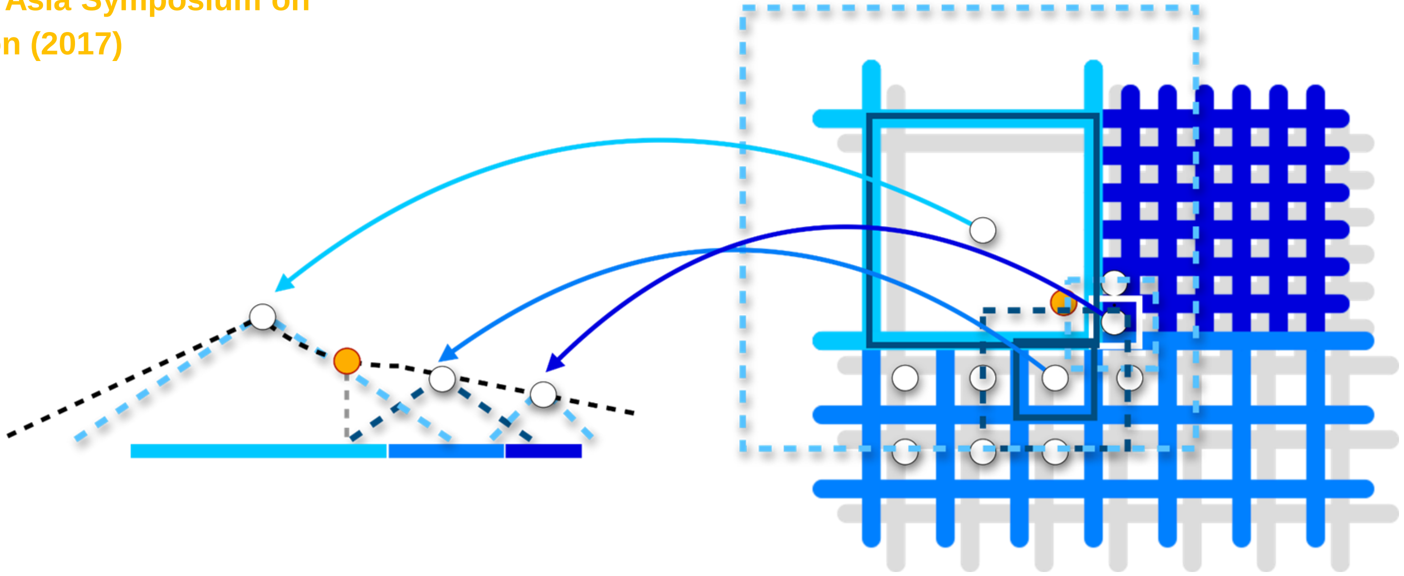
(d) Stitching

Hat-Shaped Basis Functions

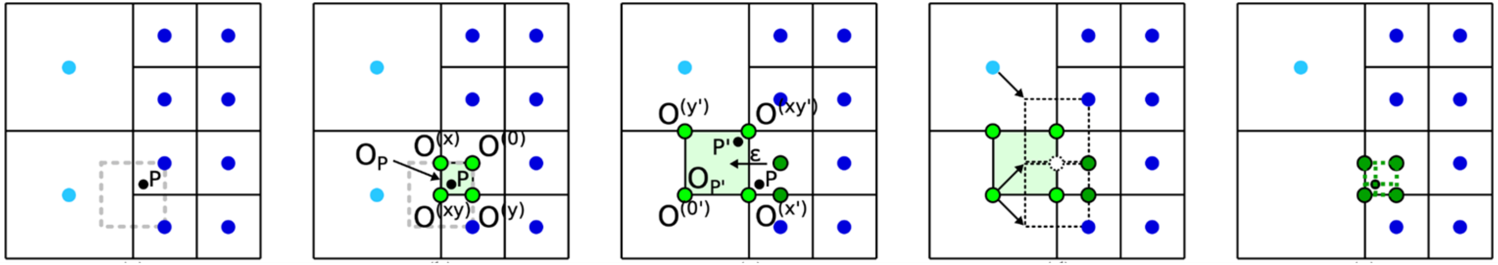


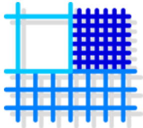
Adaptive Mesh Refinement Volume Data

Wald et al. *CPU volume rendering of adaptive mesh refinement data*, **SIGGRAPH Asia Symposium on Visualization (2017)**



Octant Method





Adaptive Mesh Refinement Volume Data

Octant method not an interpolator on its own

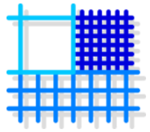
Provides tessellation of base domain into dual cells—w/o explicitly storing dual cells

Assignment of dual corner values requires interpolant (e.g., hat basis functions)

Wang et al. *CPU isosurface ray tracing of adaptive mesh refinement data*, **IEEE TVCG (2019)**

ExaBricks Data Structure

EUROVIS Leipzig 2023



Adaptive Mesh Refinement Volume Data

<u>Problem</u>: high-quality interpolators require to locate multiple cells per single sample

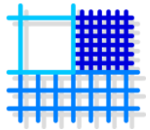
No adjacency information nor refinement level of neighboring cells available

Each located cell requires traversal of hierarchy (kd-tree over cells, etc.)

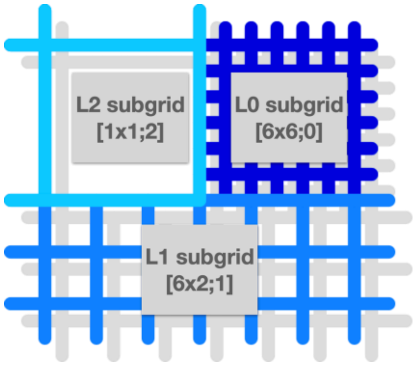
Prohibitive, especially on GPUs

Wald et al. *Ray Tracing Structrured AMR Data Using ExaBricks*, **IEEE TVCG (2021)**

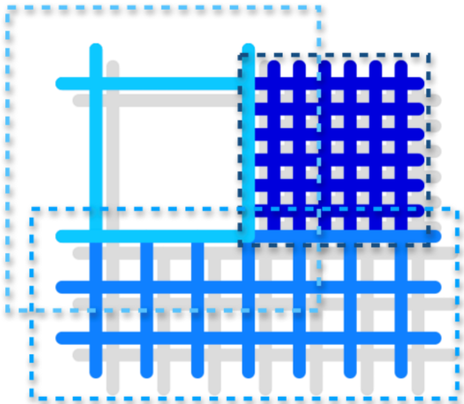
ExaBricks Data Structure



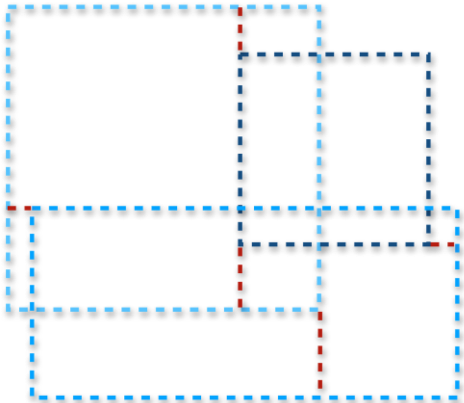
Adaptive Mesh Refinement Volume Data



<u>Step 1:</u> Build coarse bricks



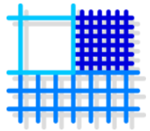
<u>Step 2:</u> Brick overlap regions



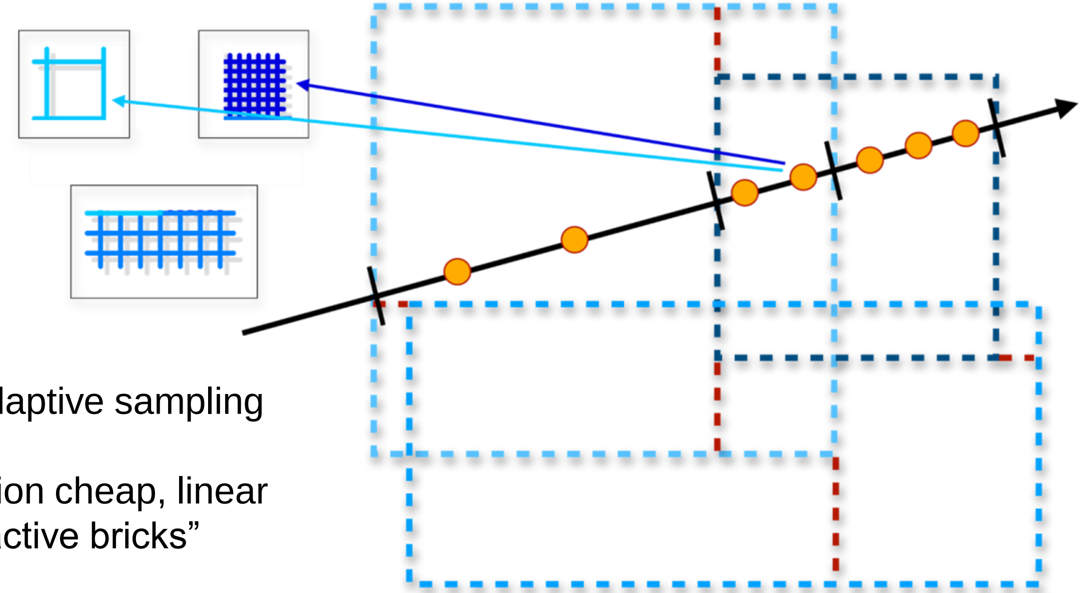
<u>Step 3:</u> OptiX overlap BVH

Wald et al. *Ray Tracing Structrured AMR Data Using ExaBricks*, **IEEE TVCG (2021)**

ExaBricks Data Structure



Adaptive Mesh Refinement Volume Data

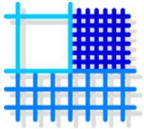


Overlap regions for adaptive sampling

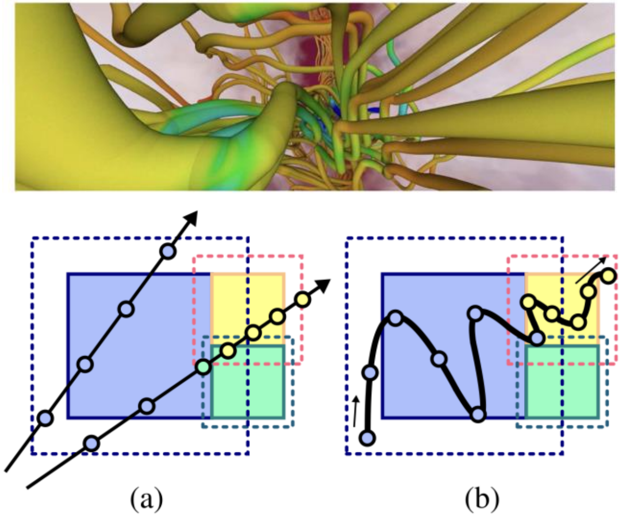
Sampling inside a region cheap, linear traversal over list of “active bricks”

Wald et al. *Ray Tracing Structrured AMR Data Using ExaBricks*, **IEEE TVCG (2021)**

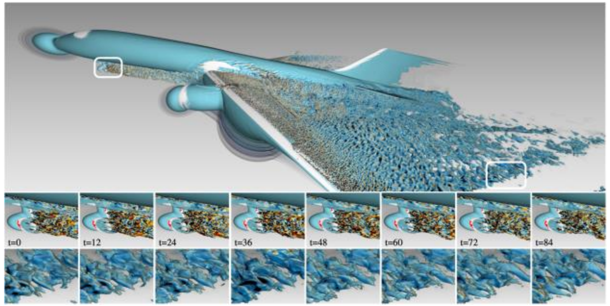
Future Research Challenges



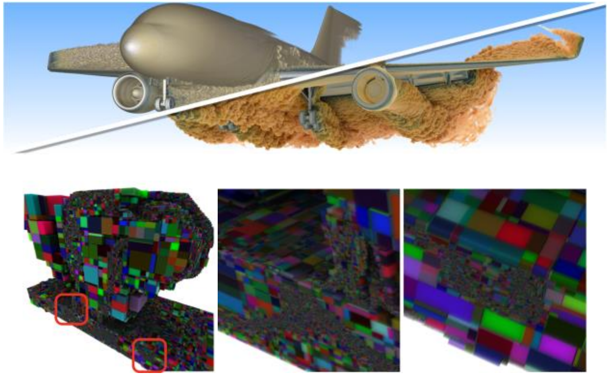
Adaptive Mesh Refinement Volume Data



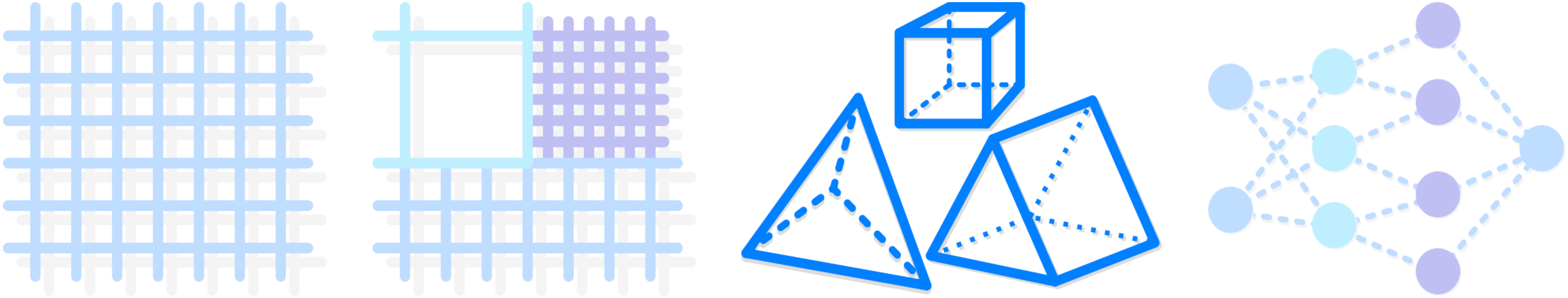
Zellmann et al. *Point Containment Queries on Ray Tracing Cores for AMR Flow Visualization*, **CiSE (2022)**



Zellmann et al. *Design and Evaluation of a GPU Streaming Framework for Visualizing Time-Varying AMR Data*, **EGPGV (2022)**



Zellmann et al. *Beyond ExaBricks: GPU Volume Path Tracing of AMR Data*, **ArXiv preprint**



Unstructured Volume Data

Background

Unstructured Volume Data

Ultimate way of refining the data to adapt to regions of interest

Tetrahedra only / different types of polyhedra (tetrahedra, hexahedra, pyramids, wedges) / high order, twisted and bent elements

Requires managing the topology, geometry, and scalar field to locate and render these polyhedral cells

Classification

Unstructured Volume Data

Recent trends in traversing and sampling unstructured volumes

- element marching approaches

Muigg et al. *Scalable hybrid unstructured and structured grid raycasting*, IEEE TVCG (2007)
Muigg et al. *Interactive volume visualization of general polyhedral grids*, IEEE TVCG (2011)
Sahistan et al. *Ray-traced shell traversal of tetrahedral meshes for direct volume visualization*, IEEE Vis (2021)

- fixed sampling pattern

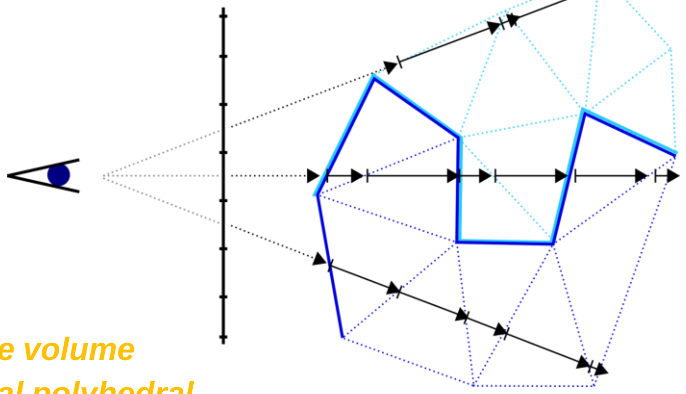
Binyahib et al. *A scalable hybrid scheme for ray-casting of unstructured volume data*, IEEE TVCG (2019)
Sahistan et al. *GPU-based data-parallel rendering of large, unstructured, and non-convexly partitioned data*, arXiv (2022)

- point ray sampling with hardware-accelerated ray-tracing

Morrical et al. *Accelerating unstructured mesh point location with RT cores*, IEEE TVCG (2022)
Wald et al. *A memory efficient encoding for ray tracing large unstructured data*, IEEE TVCG (2021)

Element marching

Traversal: marching from cell to cell via connectivity information

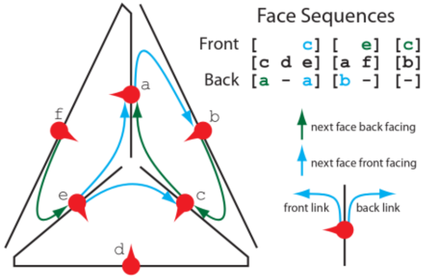


Unstructured Volume Data

Muigg et al. *Scalable hybrid unstructured and structured grid raycasting*, IEEE TVCG (2007)

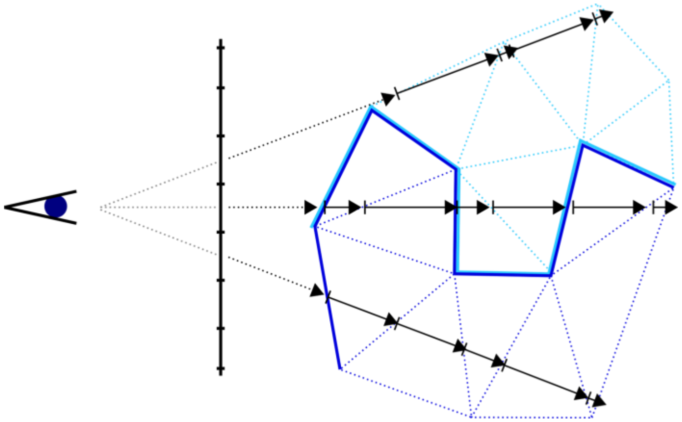
Muigg et al. *Interactive volume visualization of general polyhedral grids*, IEEE TVCG (2011)

- Decomposes the unstructured grid into bricks using a k-d tree
- Find entry/exit points: using rasterization and depth peeling
- Compact data structure to store the connectivity information: list of faces



Element marching

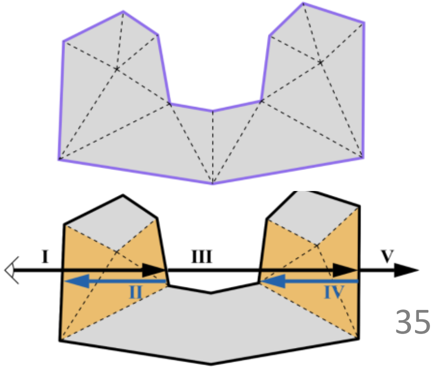
Traversal: marching from cell to cell via connectivity information



Unstructured Volume Data

Sahistan et al. *Ray-traced shell traversal of tetrahedral meshes for direct volume visualization*, IEEE Vis (2021)

- Hardware-accelerated ray tracing for element marching
- OptiX triangle BVH to find entry and exit points (over tetrahedral meshes)
- Data compression with XOR compaction



Parallel and distributed rendering

EUROVIS Leipzig 2023

Unstructured Volume Data

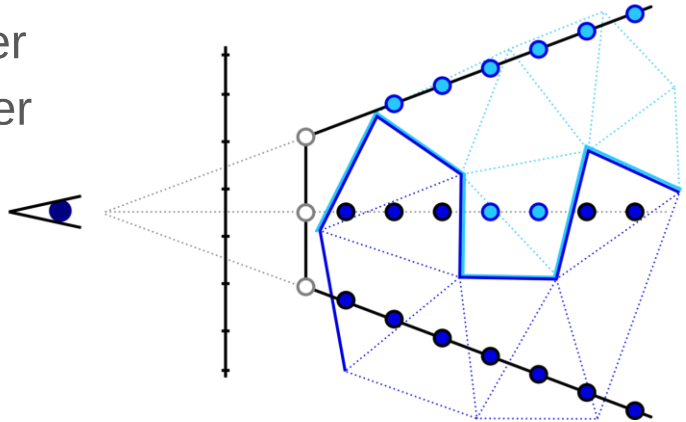
Binyahib et al. *A scalable hybrid scheme for ray-casting of unstructured volume data*, IEEE TVCG (2019)

A scalable unstructured volume rendering algorithm

- Small cells -> object-order
- Large cells -> image-order

Fixed sampling pattern

Compositing



Cell location

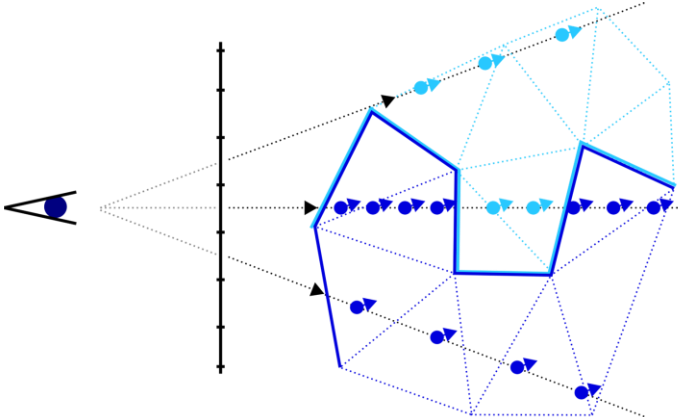
Unstructured Volume Data

Obtain the samples via **point queries**

Rathke et al. *SIMD parallel ray tracing of homogeneous polyhedral grids*, EGPGV (2015)

Use a hierarchical acceleration structure

Decouples traversal and sampling



Hardware accelerated cell location

Wald et al. *A memory efficient encoding for ray tracing large unstructured data*, IEEE TVCG (2021)

Morrical et al. *Accelerating unstructured mesh point location with RT cores*, IEEE TVCG (2022)

Cell location

Unstructured Volume Data

Hardware accelerated cell location

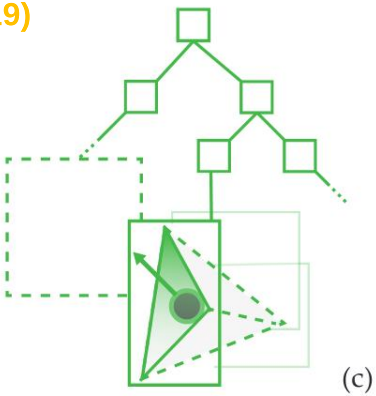
Using ray-tracing cores -> cast **zero length rays** (tmin = tmax = 0) for each sample

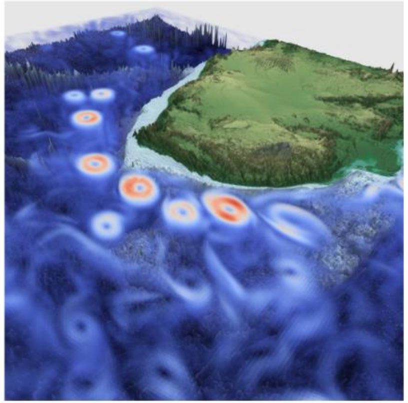
Wald et al. *RTX beyond ray tracing: Exploring the use of hardware ray tracing cores for tet-mesh point location*, HPG (2019)

Hardware BVH traversal + ray/triangle intersections

Good performances but memory intensive

Morrical et al. *Accelerating unstructured mesh point location with RT cores*, IEEE TVCG (2022)





Cell location

Unstructured Volume Data

Empty space skipping

Unstructured meshes:

- often non-convex boundaries

  -> empty space = boundary mesh and RGBα TF

- individual elements can vary significantly in size

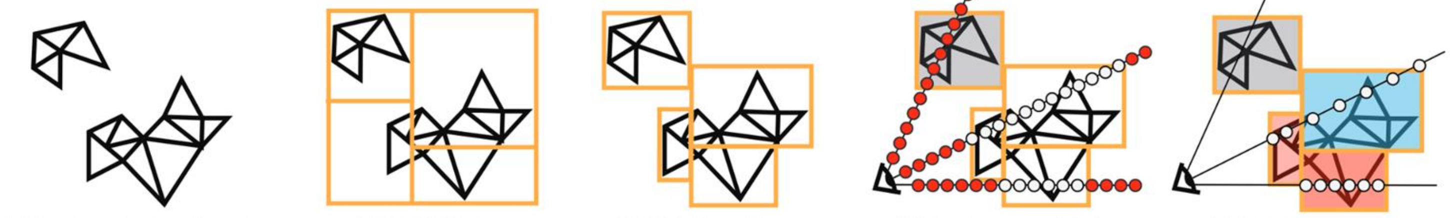
Cell location

Unstructured Volume Data

Hardware accelerated cell location with

- **Space skipping**: data structure (k-d tree partitioning) traversed using OptiX and RTX + min-max tree-based empty space skipping
- **Adaptive sampling**: adapt sampling rate according to TF variance in partitions

Morrical et al. ***Efficient space skipping and adaptive sampling of unstructured volumes using hardware accelerated ray tracing*****, IEEE Vis (2019)**



(a) Input unstructured mesh (b) Build KD-tree (c) Shrink partitions (d) Empty space skipping (e) Adaptive sampling

Memory usage

Element marcher: only requiring face connectivity

-> can be compressed significantly

Vs.

Hardware-accelerated point queries: require BVH and complete vertex index lists

-> very high memory costs

Cell location

Unstructured Volume Data

Mesh data compression

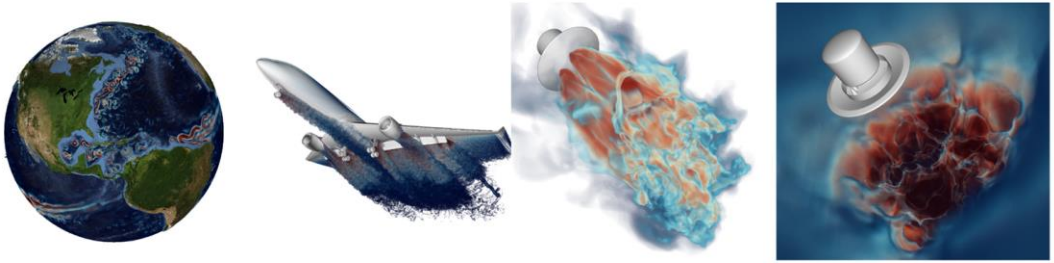
Meshlet decomposition - compact indices representation with element regrouping

Acceleration structure compression

Eight-wide BVH + quantization - rearranging nodes and primitives

Collapse leaves

Wald et al. *A memory efficient encoding for ray tracing large unstructured data*, IEEE TVCG (2021)



DKRZ full ocean	NASA Exa-Jet	NASA Mars Lander (small)	NASA Mars Lander (large)
485 M verts, 749 M elements	656 M verts, 652 M elements	143 M verts, 789 M elements	576 M verts, 2.9 B elements
compression rate 5.9 : 1	compression rate 4.9 : 1	compression rate 14.0 : 1	compression rate 12.3 : 1

Cell location

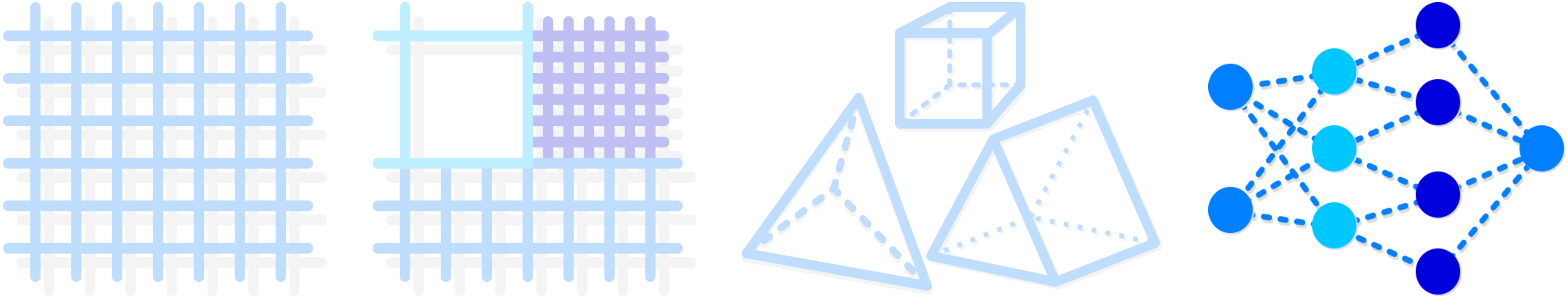
Unstructured Volume Data

Extreme compression but bottleneck for **acceleration structure building time**

Morrical et al. *Quick clusters: A GPU-parallel partitioning for efficient path tracing of unstructured volumetric grids*, IEEE TVCG (2022)

Lazy sorting and clustering scheme -> much lower pre-processing times

-> trades sampling performance for pre-processing times



Compressed and Neural Representations

Hierarchical Compression

EUROVIS Leipzig 2023

Compressed & Neural Representations

Unified **resolution–precision** compression

- Reducing grid **resolution**
- Reducing scalar **precision**

Fine control over resolution vs. precision

Hoang et al. ***Efficient and flexible hierarchical data layouts for a unified encoding of scalar field precision and resolution*****, IEEE TVCG (2021)**

Bhatia et al. ***AMM: Adaptive Multilinear Meshes*****, IEEE TVCG (2022)**

Neural Representations

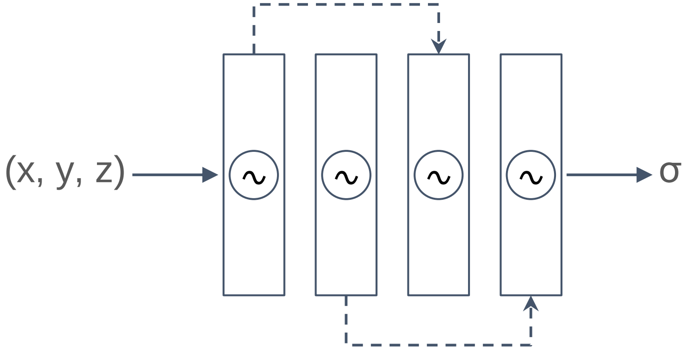
Compressed & Neural Representations

Coordinate-based scene representation networks

- $F(x, y, z) \rightarrow \tau, \sigma, \delta, rgb$

Lu et al. → Compressive neural representations

Weiss et al. → Faster inference



Lu et al., *Compressive neural representations of volumetric scalar fields*, Computer Graphics Forum 40, 3 (2021)

Weiss et al., *Fast neural representations for direct volume rendering.* Computer Graphics Forum 41, 6 (2022)

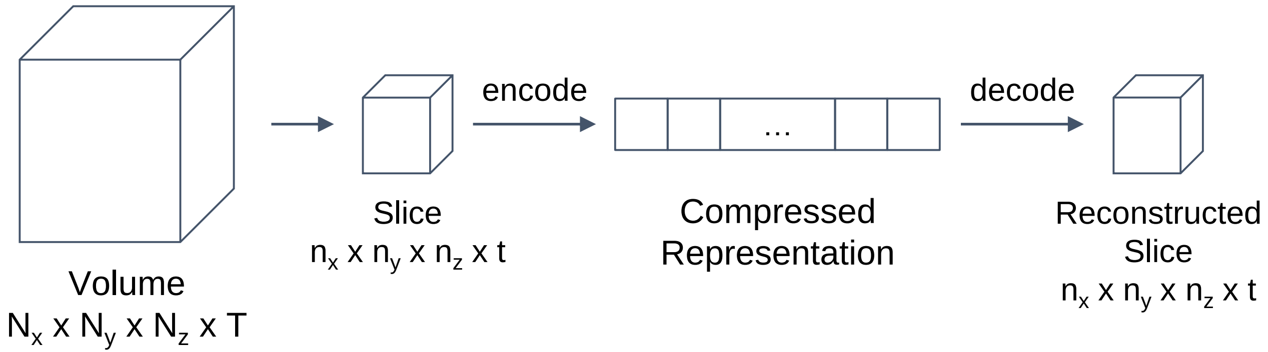
Neural Representations

Compressed & Neural Representations

Autoencoder based methods for time-varying data,

Pan et al., *Adaptive deep learning based time-varying volume compression.* IEEE International Conference on Big Data (2019)

Jain et al., *Compressed volume rendering using deep learning.* In Proceedings of LDAV (2017)



Findings

Classification

EUROVIS Leipzig 2023

	Platform		Architecture			Traversal		Sampling	
	CPU	GPU	in-core	out-of-core	distributed	software	hardware	hierarchical	direct/linear
🟢 [HAAB*18] [BMA*19] [SCRL20]		✓		✓		✓		✓	
🟢 [WWJ19]	✓			✓		✓		✓	
🟢 [GM19] [WZM21]		✓	✓				✓		✓
🟢 [ZSL21]	✓	✓	✓			✓			✓
🟡 [WBUK17] [WWW*19] [WMU*20]	✓		✓			✓		✓	
🟡 [WZU*21] [ZWS*22a] [ZSM*22]		✓	✓				✓		✓
🟡 [ZWS*22b]		✓	✓			✓	✓	✓	✓
🟠 [MHDH07] [MHDG11]		✓	✓			✓			✓
🟠 [SDM*21]		✓	✓				✓		✓
🟠 [BPL*19]		✓			✓				✓
🟠 [SDW*22]		✓			✓		✓		✓
🟠 [WUM*19] [MUWP19] [WMZ21] [MWUP22] [MSG*22]		✓	✓				✓	✓	
🟣 [HSB*21] [BHM*22]				✓				✓	
🟣 [LJLB21]		✓	✓						

🟢 Structured 🟡 AMR 🟠 Unstructured 🟣 Compressed and neural

Classification

EUROVIS Leipzig 2023

	Platform		Architecture			Traversal		Sampling	
	CPU	GPU	in-core	out-of-core	distributed	software	hardware	hierarchical	direct/linear
[HAAB*18] [BMA*19] [SCRL20]		✓		✓		✓		✓	
[WWJ19]	✓			✓		✓		✓	
[GM19] [WZM21]		✓	✓				✓		✓
[ZSL21]	✓	✓	✓			✓			✓
[WBUK17] [WWW*19] [WMU*20]	✓		✓			✓		✓	
[WZU*21] [ZWS*22a] [ZSM*22]		✓	✓				✓		✓
[ZWS*22b]		✓	✓			✓	✓	✓	✓
[MHDH07] [MHDG11]		✓	✓			✓			✓
[SDM*21]		✓	✓				✓		✓
[BPL*19]		✓			✓				✓
[SDW*22]		✓			✓		✓		✓
[WUM*19] [MUWP19] [WMZ21] [MWUP22] [MSG*22]		✓	✓				✓	✓	
[HSB*21] [BHM*22]				✓				✓	
[LJLB21]		✓	✓						

Structured AMR Unstructured Compressed and neural

Classification

EUROVIS Leipzig 2023

	Platform		Architecture			Traversal		Sampling	
	CPU	GPU	in-core	out-of-core	distributed	software	hardware	hierarchical	direct/linear
● [HAAB*18] [BMA*19] [SCRL20]		✓		✓		✓		✓	
● [WWJ19]	✓			✓		✓		✓	
● [GM19] [WZM21]		✓	✓				✓		✓
● [ZSL21]	✓	✓	✓			✓			✓
● [WBUK17] [WWW*19] [WMU*20]	✓		✓			✓		✓	
● [WZU*21] [ZWS*22a] [ZSM*22]		✓	✓				✓		✓
● [ZWS*22b]		✓	✓			✓	✓	✓	✓
● [MHDH07] [MHDG11]		✓	✓			✓			✓
● [SDM*21]		✓	✓				✓		✓
● [BPL*19]		✓			✓				✓
● [SDW*22]		✓			✓		✓		✓
● [WUM*19] [MUWP19] [WMZ21] [MWUP22] [MSG*22]		✓	✓				✓	✓	
● [HSB*21] [BHM*22]				✓				✓	
● [LJLB21]		✓	✓						

● Structured ● AMR ● Unstructured ● Compressed and neural

Classification

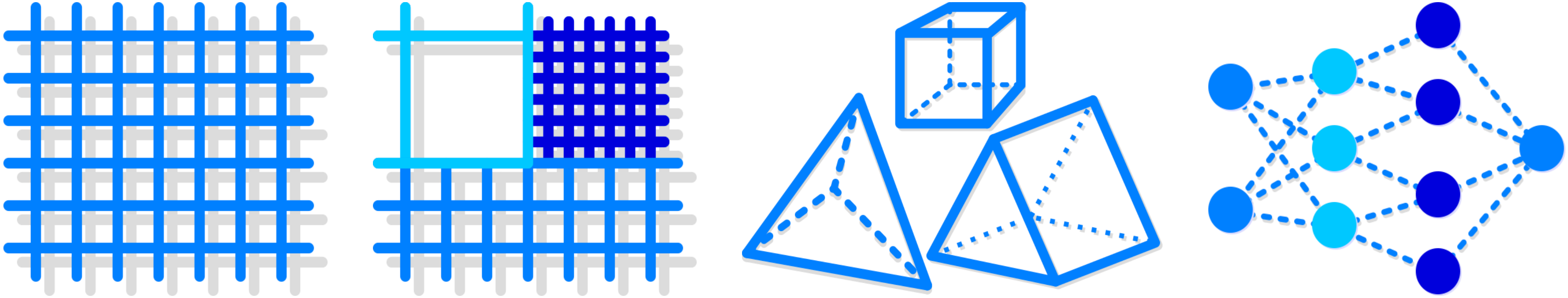
	Platform		Architecture			Traversal		Sampling	
	CPU	GPU	in-core	out-of-core	distributed	software	hardware	hierarchical	direct/linear
🟢 [HAAB*18] [BMA*19] [SCRL20]		✓		✓		✓		✓	
🟢 [WWJ19]	✓			✓		✓		✓	
🟢 [GM19] [WZM21]		✓	✓				✓		✓
🟢 [ZSL21]	✓	✓	✓			✓			✓
🟡 [WBUK17] [WWW*19] [WMU*20]	✓		✓			✓		✓	
🟡 [WZU*21] [ZWS*22a] [ZSM*22]		✓	✓				✓		✓
🟡 [ZWS*22b]		✓	✓			✓	✓	✓	✓
🟠 [MHDH07] [MHDG11]		✓	✓			✓			✓
🟠 [SDM*21]		✓	✓				✓		✓
🟠 [BPL*19]		✓			✓				✓
🟠 [SDW*22]		✓			✓		✓		✓
🟠 [WUM*19] [MUWP19] [WMZ21] [MWUP22] [MSG*22]		✓	✓				✓	✓	
🟣 [HSB*21] [BHM*22]				✓				✓	
🟣 [LJLB21]		✓	✓						

🟢 Structured 🟡 AMR 🟠 Unstructured 🟣 Compressed and neural

Trends observation

- Going more toward unstructured format than regular format
- Sampling workloads become increasingly incoherent:
  - ❖ secondary shadow or scattering rays into the volume
  - ❖ arbitrary sampling e.g., to compute flow visualizations
  - ❖ path tracing is adopted by the sci-vis community
- Compressing given data representations; extremely lossy compression by encoding the volume as a neural network (NeRF)

Conclusion



Our contribution:

Review of recent research in interactive volume visualization for large-scale data, divided into 4 main data representations.

Mostly focus on general-purpose ray tracing techniques.

Conclusion

EUROVIS Leipzig 2023

Sampling is critical for reconstructing values, but becomes costly the less regular the data.

Introduction of hierarchical and less structured volume data to reduce memory overhead, but at the cost of complex visualization algorithms.

Popularization of GPUs with accelerated ray tracing capability, which helps with visualization algorithm flexibility but is still heavy in memory consumption and interactive changes.

EUROVIS
Leipzig 2023

Thank you for your attention!