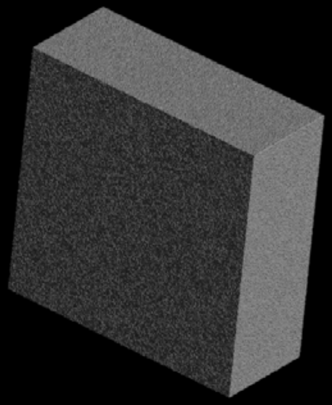
Screen-Space Normal Distribution Function Caching for Consistent Multi-Resolution Rendering of Large Particle Data ... A CPU Prospect?

Mohamed Ibrahim[1], Patrick Wickenhäuser[2], Peter Rautek[1], Guido Reina[2], Markus Hadwiger[1]

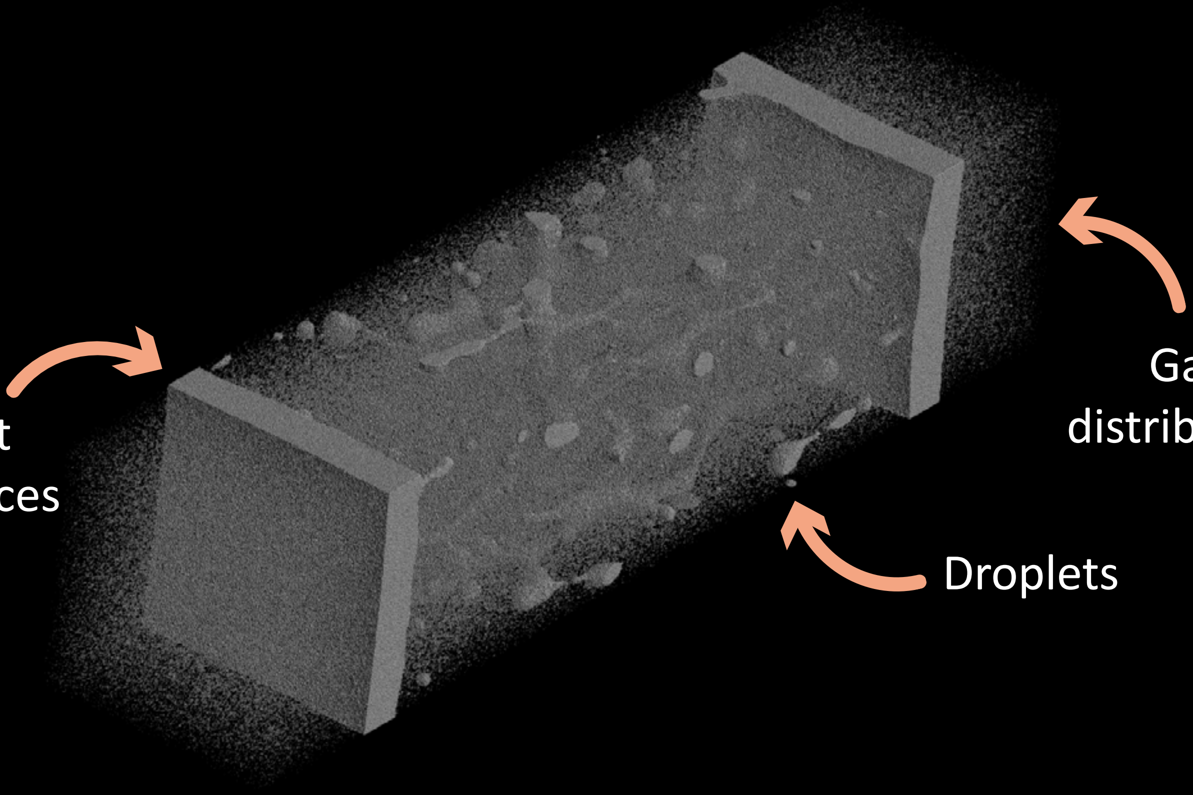
[1]King Abdullah University of Science and Technology (KAUST),
[2]University of Stuttgart

IEEE VIS 2017

Expanding Fluid Layer (30 million particles)



Expanding Fluid Layer (30 million particles)



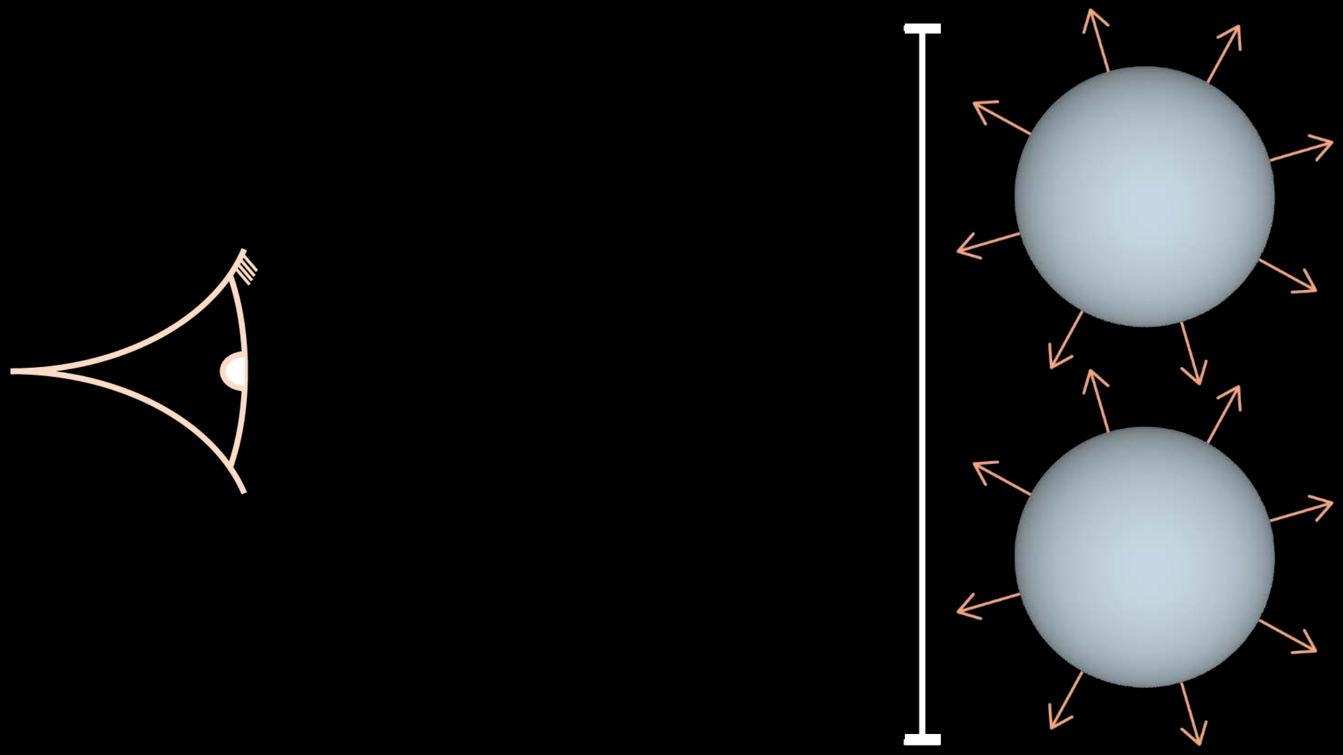
Flat Surfaces

Gas distribution

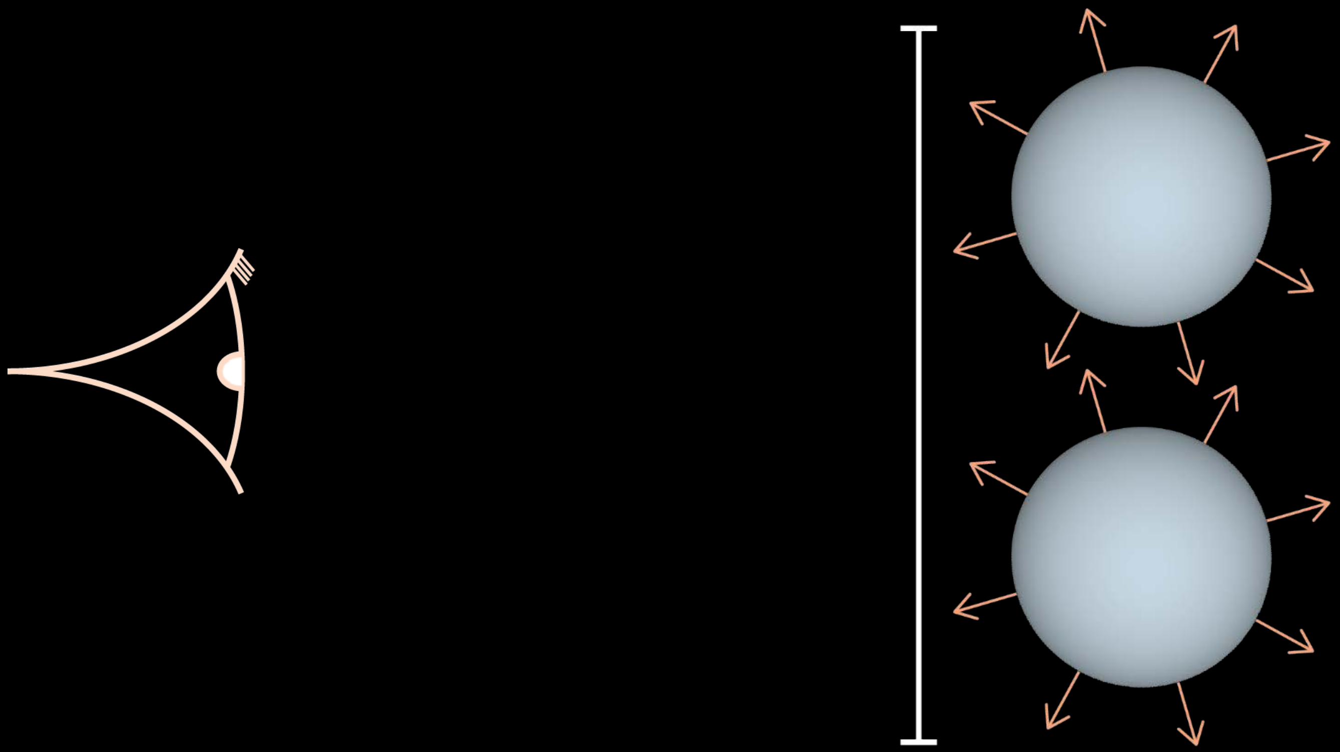
Droplets

Standard Raycasting

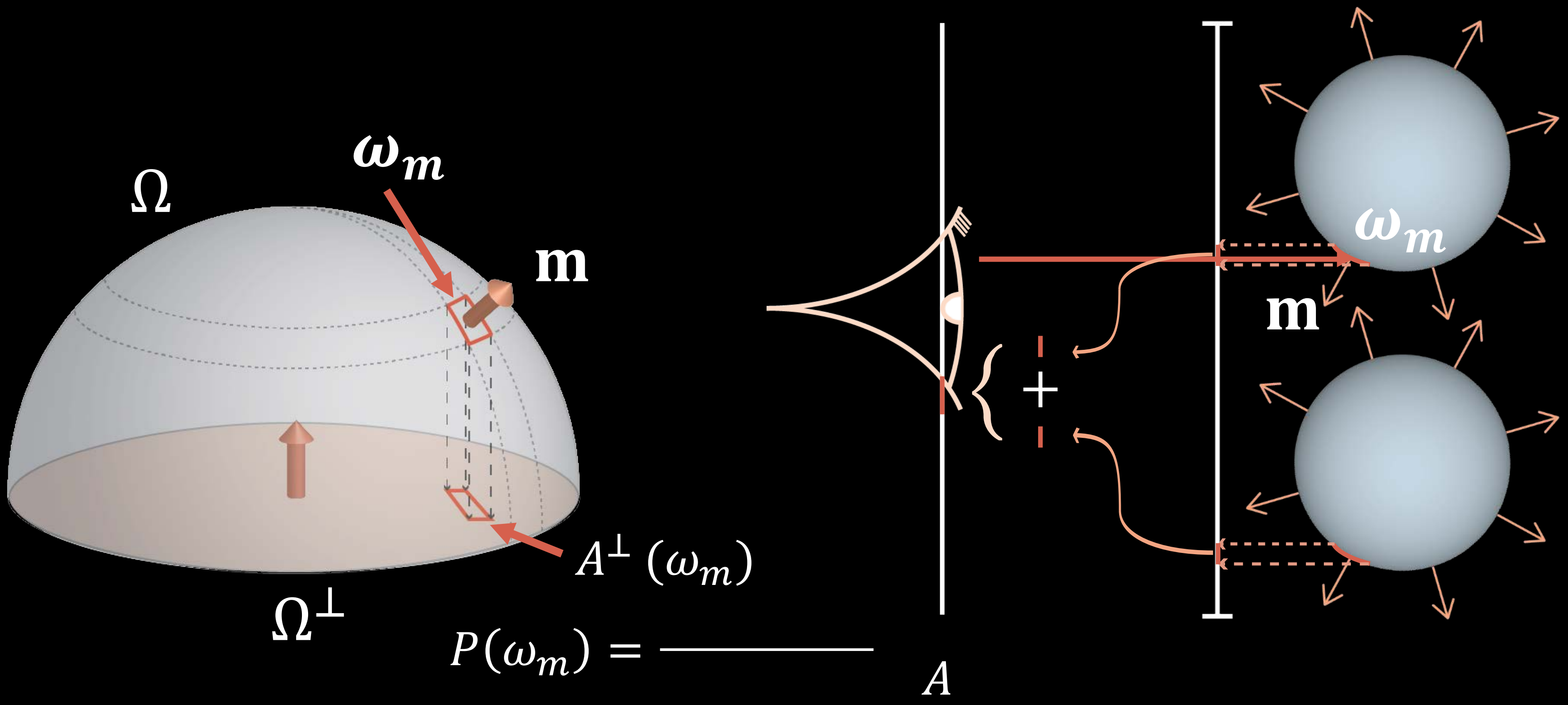
Screen-Space Normal Distribution Functions (S-NDFs)



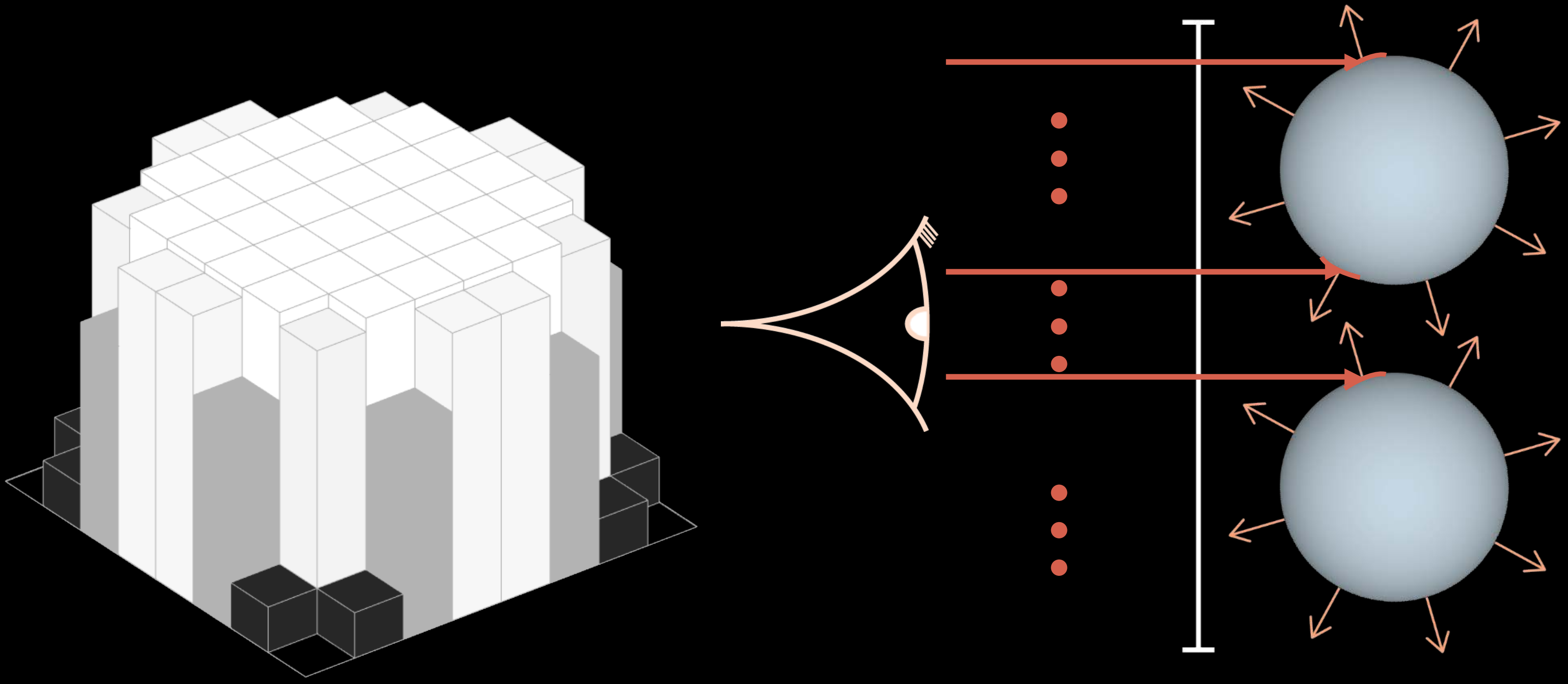
Screen-Space Normal Distribution Functions (S-NDFs)



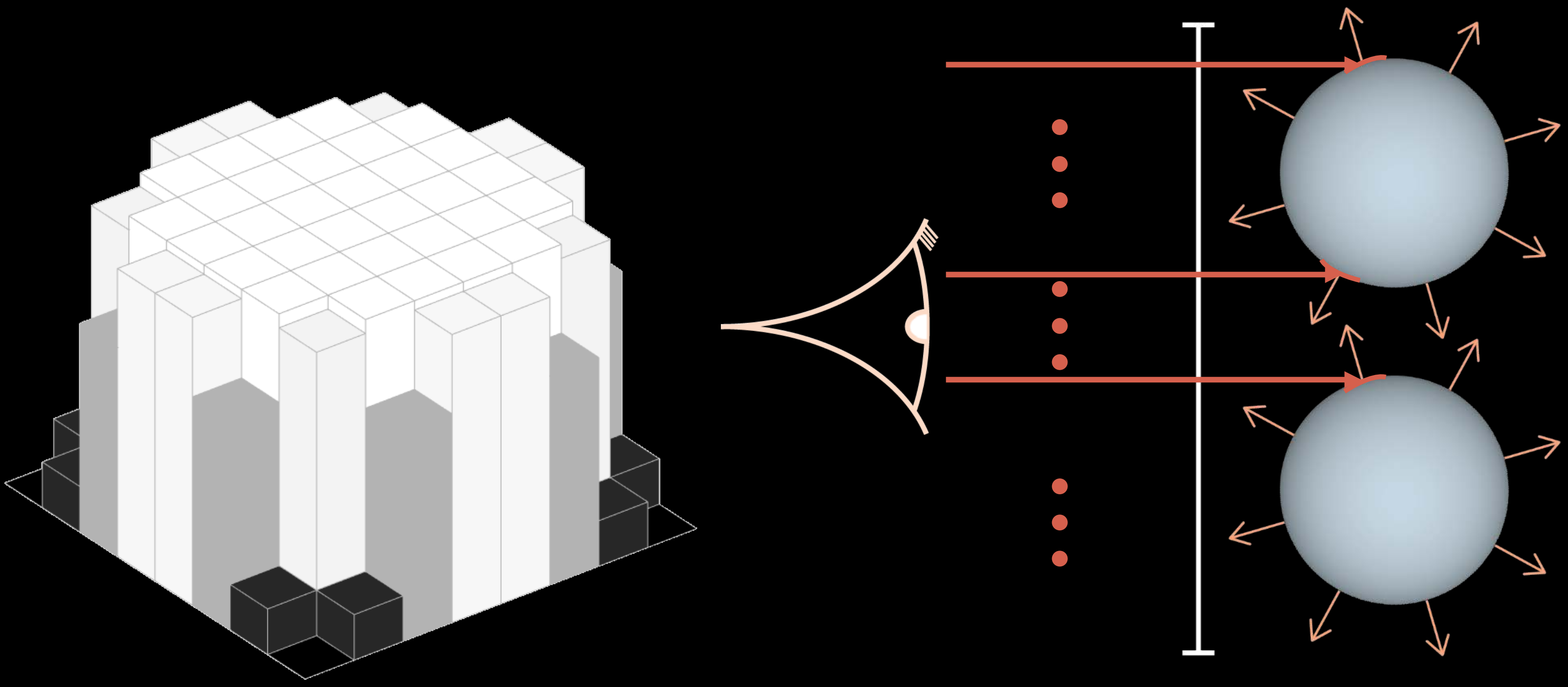
Screen-Space Normal Distribution Functions (S-NDFs)



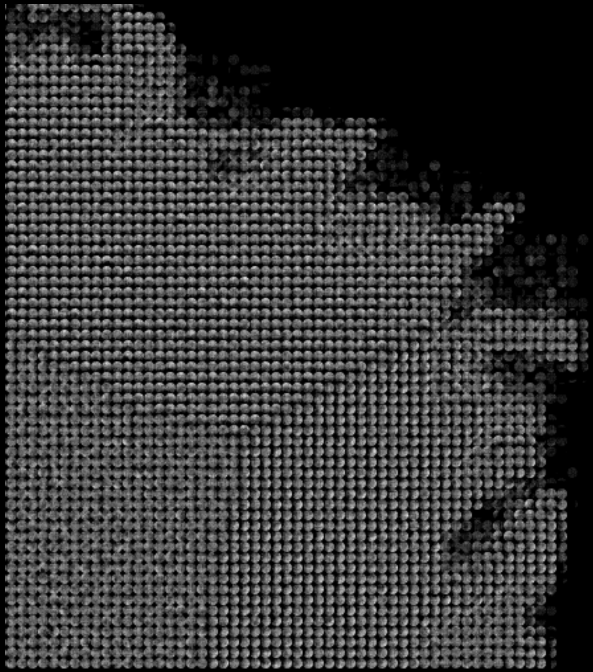
Screen-Space Normal Distribution Functions (S-NDFs)



Screen-Space Normal Distribution Functions (S-NDFs)

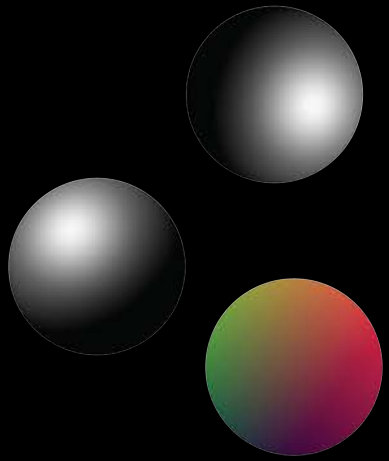


Screen-Space Normal Distribution Functions (S-NDFs)



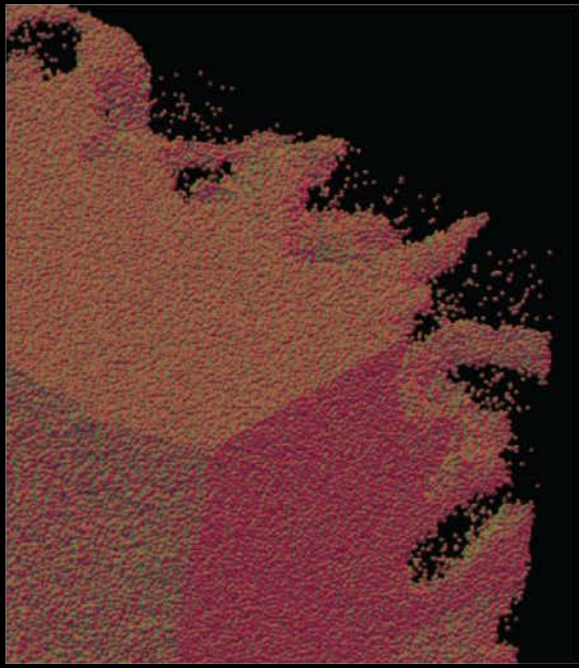
S-NDFs

*



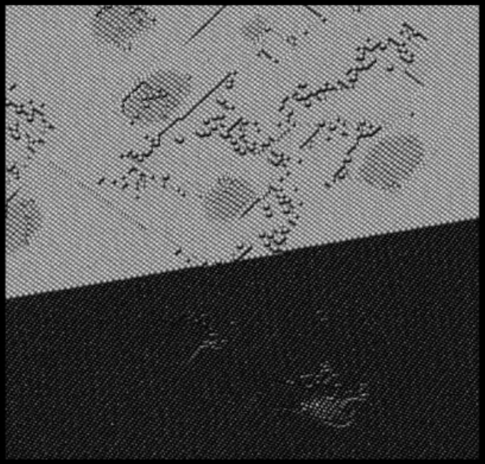
Transfer function

⇒

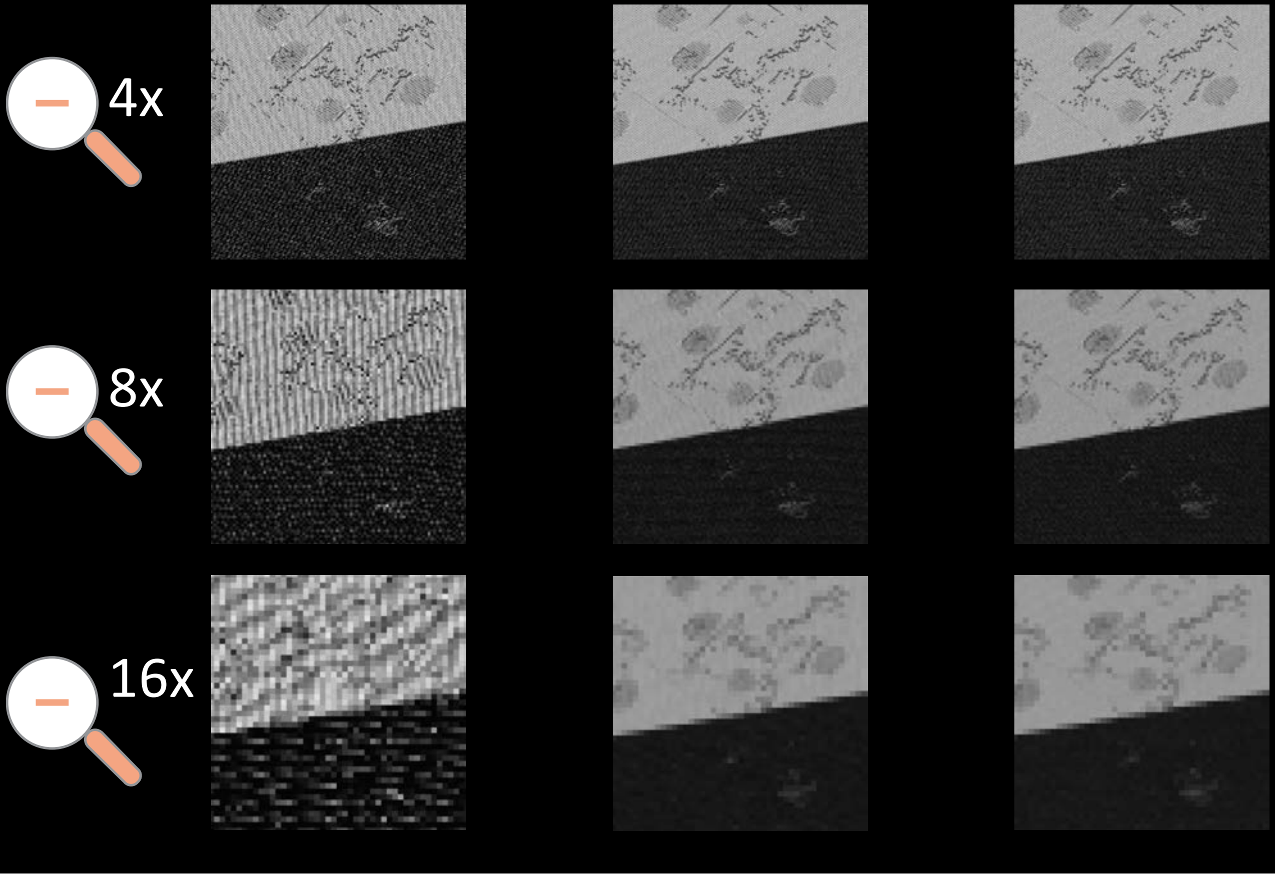


Rendered output

Scale Consistency



Lod 0

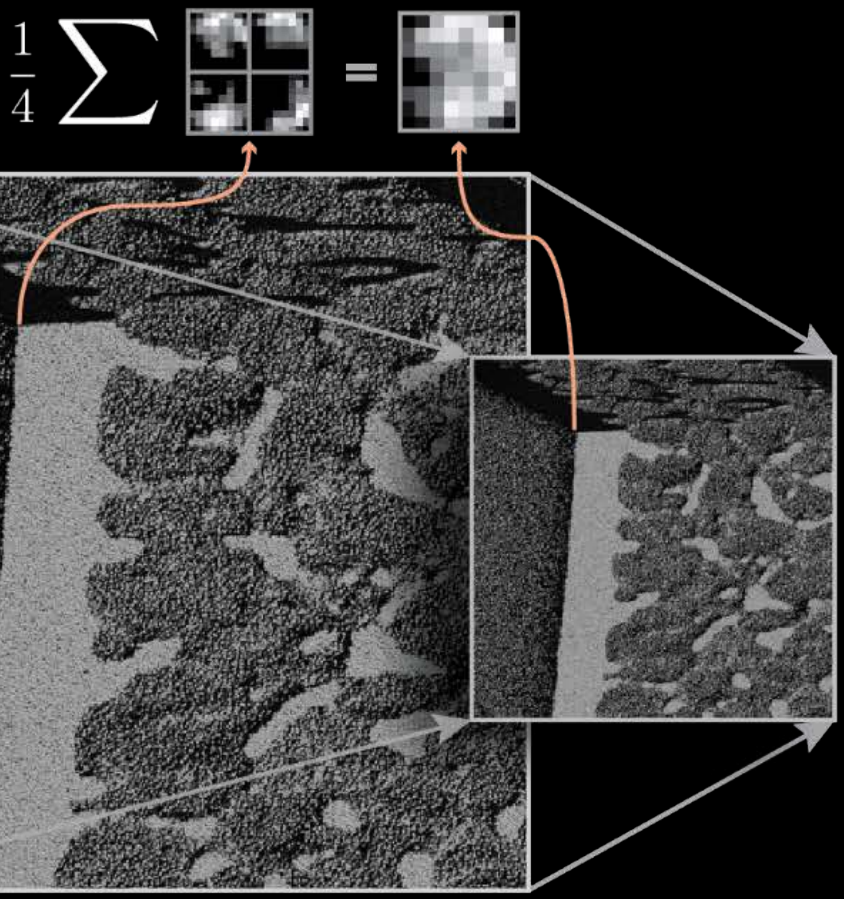


Raycasting (1 ray)

Raycasting (4^{lod} rays)

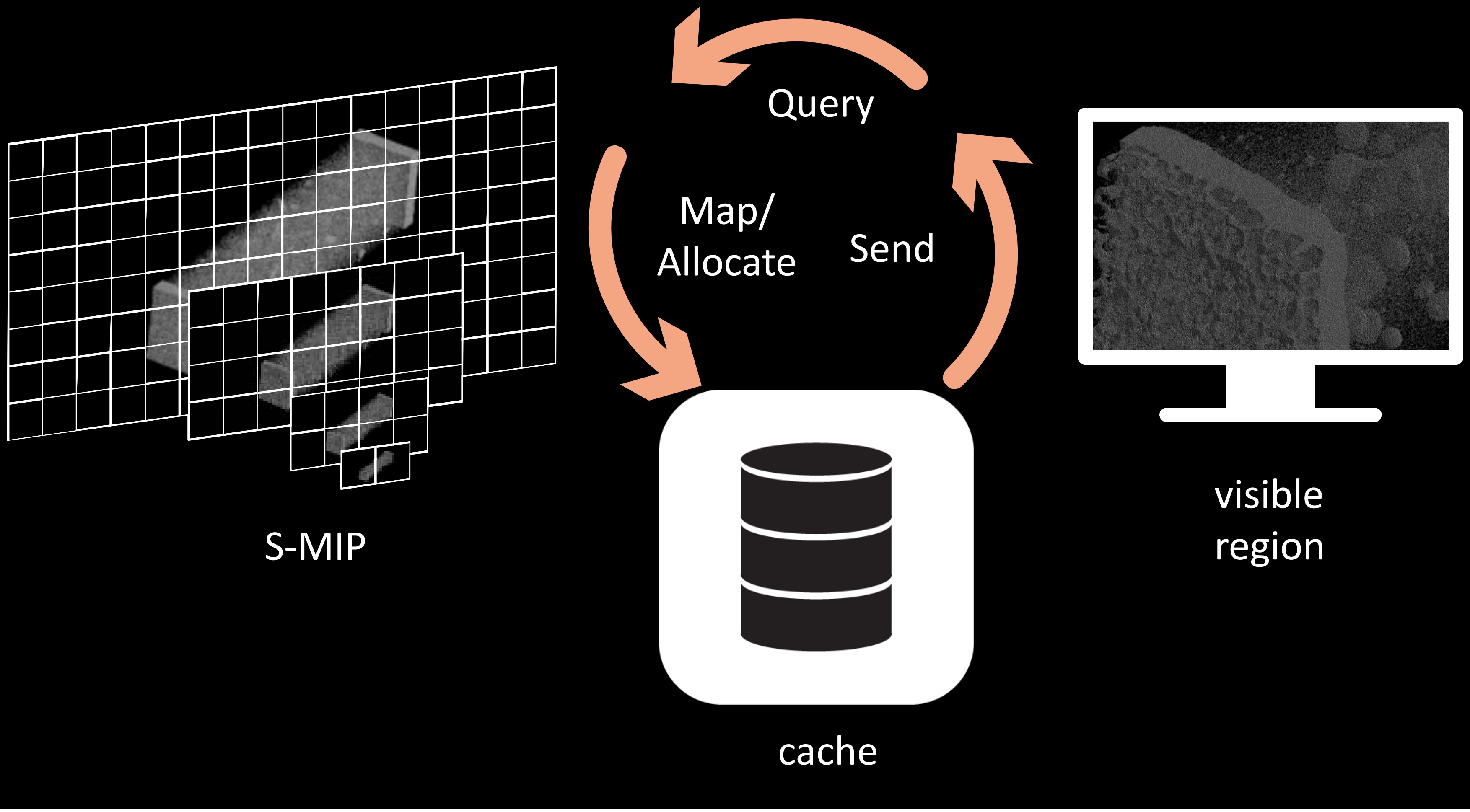
Our Approach

Linearly Filtering S-NDFs



✓

Caching S-NDFs



Query

Map/ Allocate

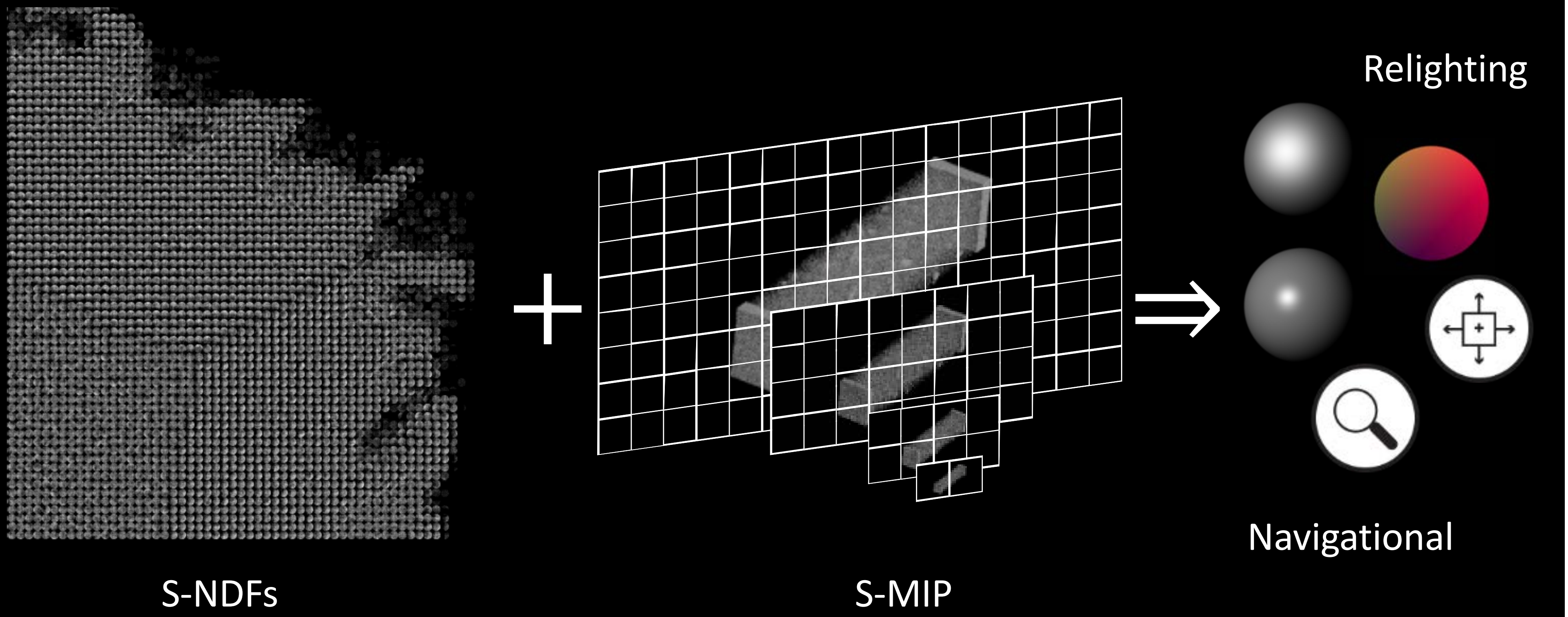
Send

S-MIP

cache

visible region

Supported Operations



S-NDFs

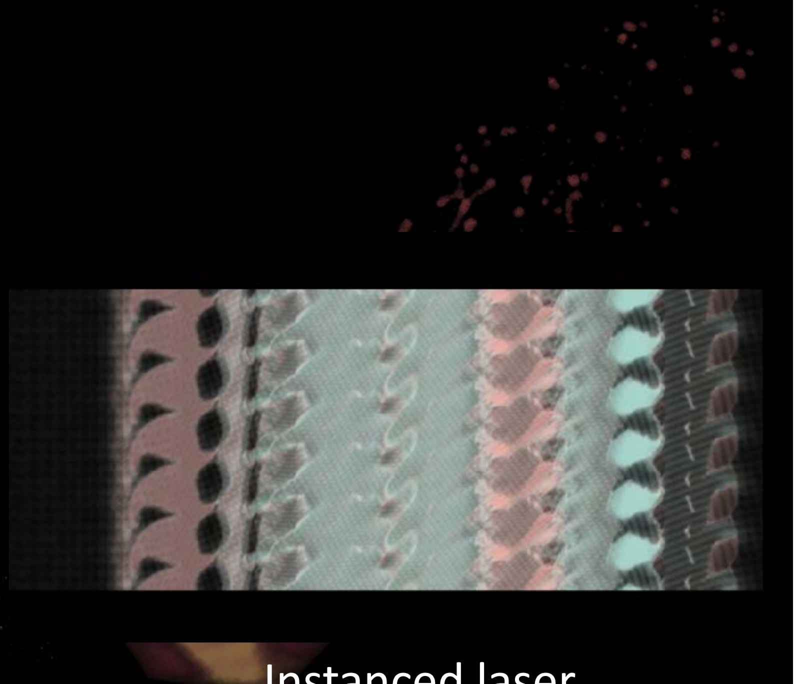
S-MIP

Relighting

Navigational

Performance Measures (GPU)

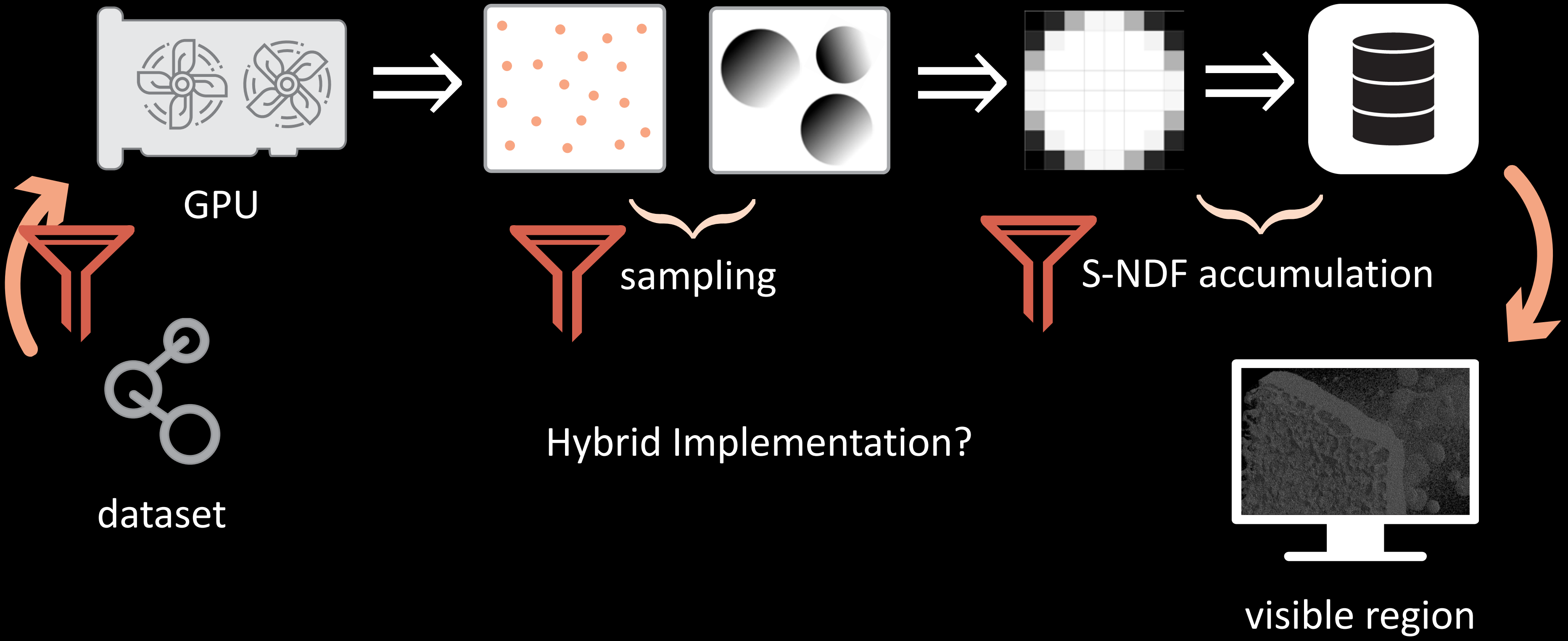
data set	particle count	re-light [ms]	
		ray casting + S-MIP	S-NDF + S-MIP
Laser ablation crown	518,000	1129.74	26.53
Copper/silver mixture	14,500,00	1309.35	19.37
Expanding fluid layer	30,000,000	1523.61	22.19
Laser ablation	48,000,000	1253.84	22.17
Instanced laser (5×)	5× 12,500,000	1524.25	18.95
Large laser ablation	199,940.704	2203.66	16.87



Instanced laser

Expanding fluid layer

Implementation Details



Hybrid Implementation?

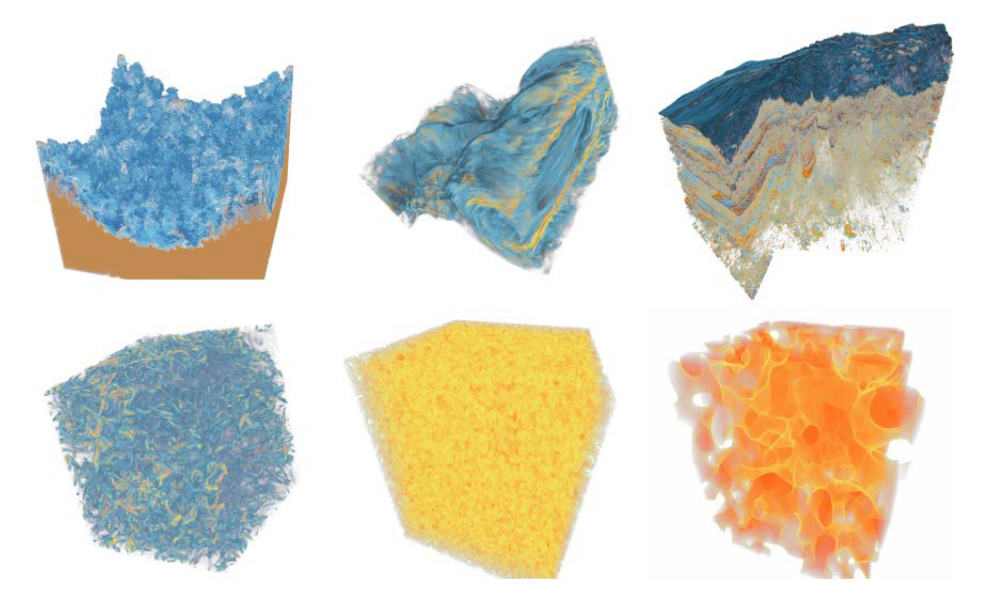
OSPRay

Rendering API

Platform Independent

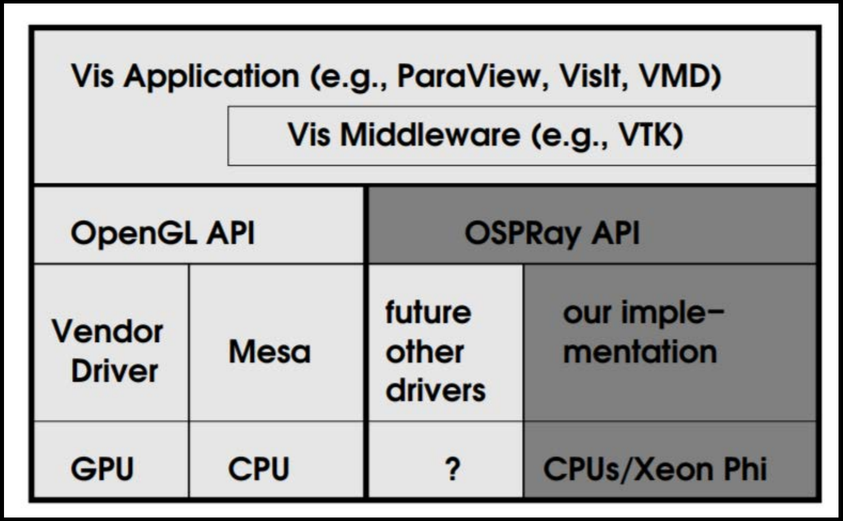
Visualization-oriented

CPU implementation

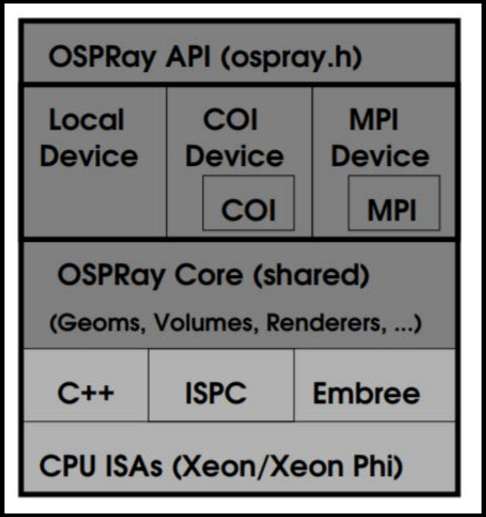


OSPRay – A CPU Ray Tracing Framework for Scientific Visualization.
Wald, I. et al., IEEE Scientific Visualization 2016

OSPRay … Architecture



Rendering API

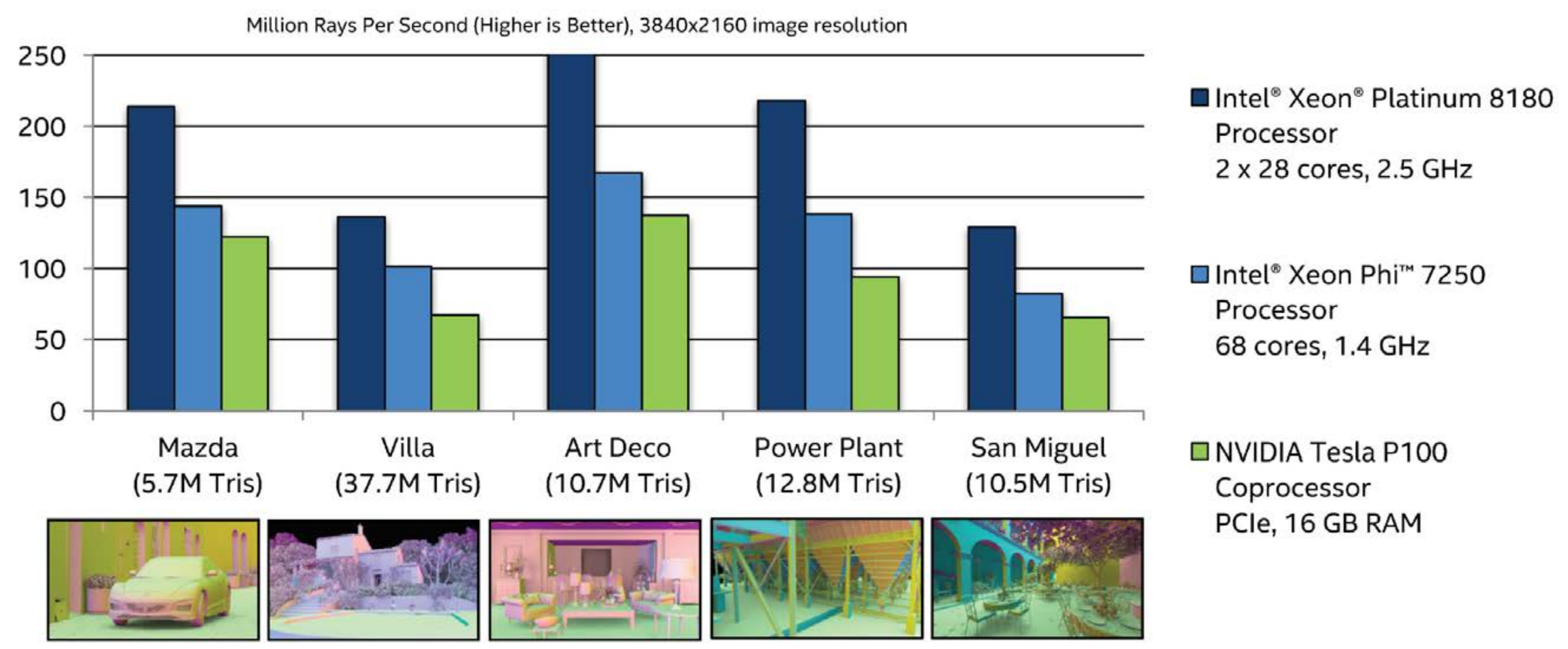


Device Abstraction

OSPRay – A CPU Ray Tracing Framework for Scientific Visualization.
Wald, I et al., IEEE Scientific Visualization 2016

OSPRay … Performance

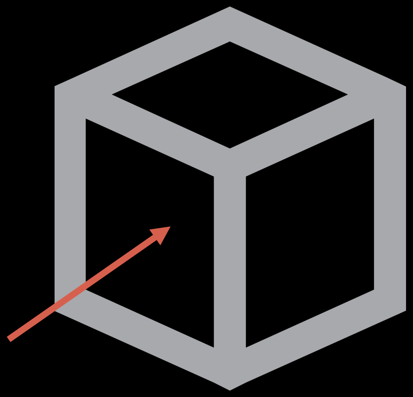
Diffuse Path Tracing Performance of Embree vs. NVIDIA OptiX Prime.



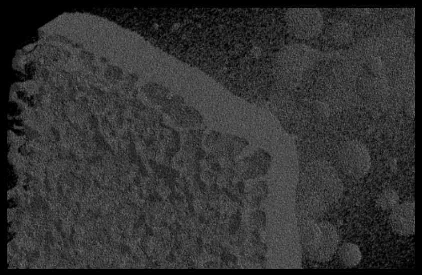
Embree: A Kernel Framework for Efficient CPU Ray Tracing.
Wald, I. et al., ACM SIGGRAPH 2014

OSPRay … Integration

RTCBoundsFunc

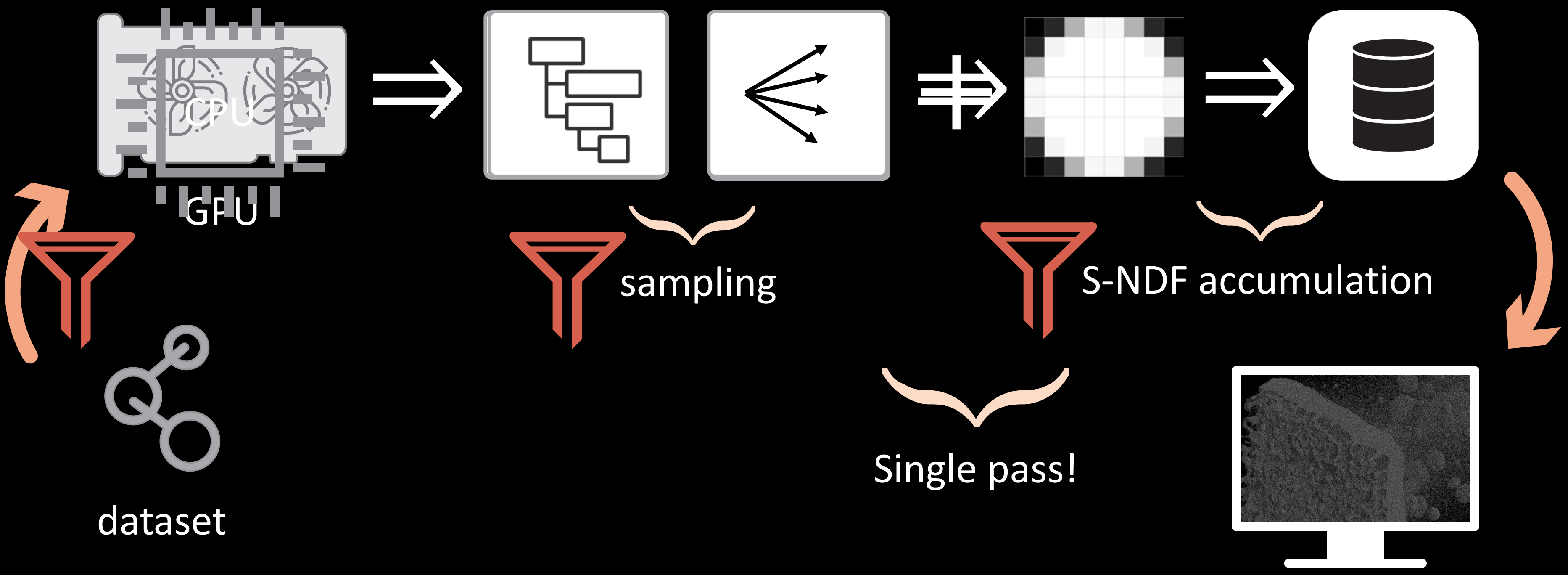


RTCIntersectFunc



RTCOccludedFunc

OSPRay + S-NDFs + S-MIP?



CPU

GPU

sampling

S-NDF accumulation

Single pass!

dataset

visible region

OSPRay + S-NDFs + S-MIP?

Large data

Large cache

Finer S-NDFS

4D S-NDFS

Fast traversal

Multiple samples

Guided sampling

Paraview

MegaMol

VMD

VisIt

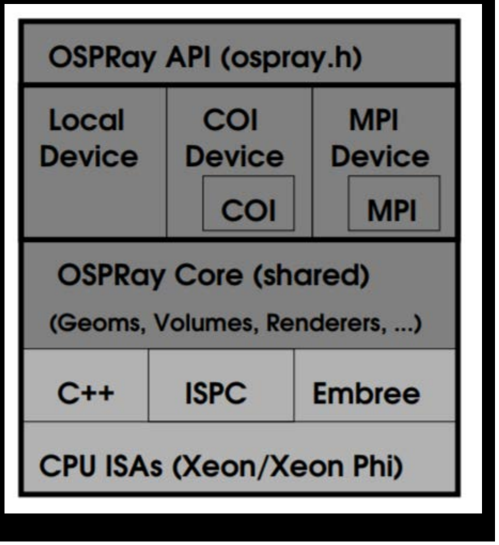
Thank you

http://www.vccvisualization.org

Application Scenarios

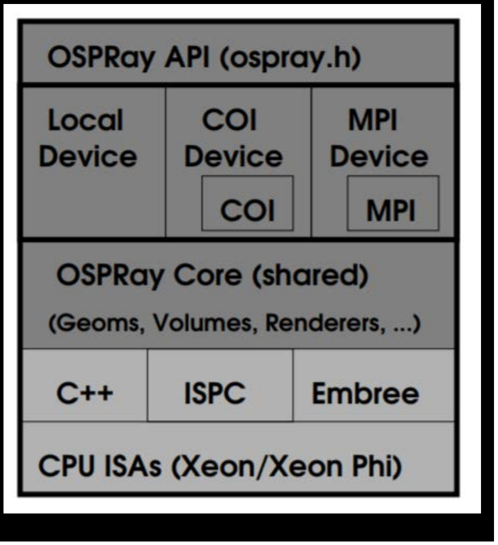
Implementation … Device Abstraction

- Ideal implementation differs according to target:
  - PCI-card based on Xeon Phi coprocessor
  - MPI-parallelism
- OSPRay supports device abstraction
  - API calls are seen as a stream of commands.
    - Each, internally, routed to one of multiple possible backend devices.



Implementation … Device Abstraction

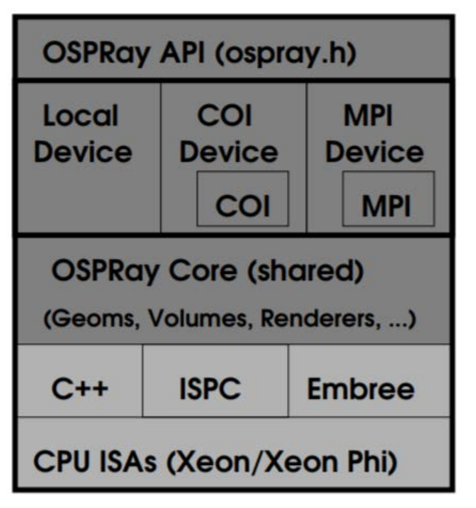
- Local Device
  - Executes all rendering right in the same process as a the visualization application.
- COI Device
  - Offload to first-generation Xeon Phi Knights Corner coprocessor cards.
- MPIDevice
  - Support to MPI-parallel rendering
- All devices build on top of a shared rendering infrastructure that implements the actors.



Software Infrastructure

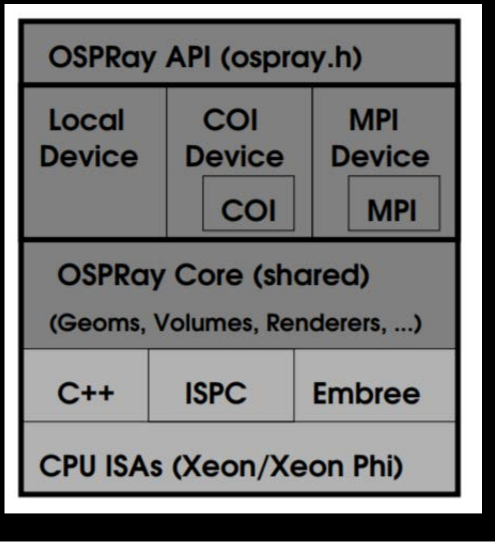
- Build on top of
  - Intel SPMD Program Compiler (ISPC)
  - Embree ray tracing kernels

- Embree
  - Building and traversing acceleration structures.
  - Automatically selects acceleration structure and traversal kernels best suited for a given CPU.



Software Infrastructure

- ISPC
  - All throughput sensitive operations that require vectorization (rendering, shading, primitive intersection)
  - Allows transparent targeting of multiple vector instruction set architectures (ISA).
    - Intel SSE4, AVX, AVX2, etc…



OSPRay API (ospray.h)

Local Device

COI Device

COI

MPI Device

MPI

OSPRay Core (shared)

(Geoms, Volumes, Renderers, ...)

C++

ISPC

Embree

CPU ISAs (Xeon/Xeon Phi)