Intel ® oneAPI Rendering Toolkit

Intel® Open Volume Kernel Library

Johannes Meng
Graphics Software Engineer

intel.

What is Open VKL?

- Highly-optimized volume sampling and traversal kernel library
- “Embree for Volumes”
- Support for latest CPUs and ISAs (e.g. Intel® AVX-512)
- Windows*, macOS* 10.x, Linux* support
- API for easy integration into applications
- Open Source under Apache* 2.0 license www.openvkl.org

What is Open VKL good for?

- Open VKL provides data structures that store volumetric fields.

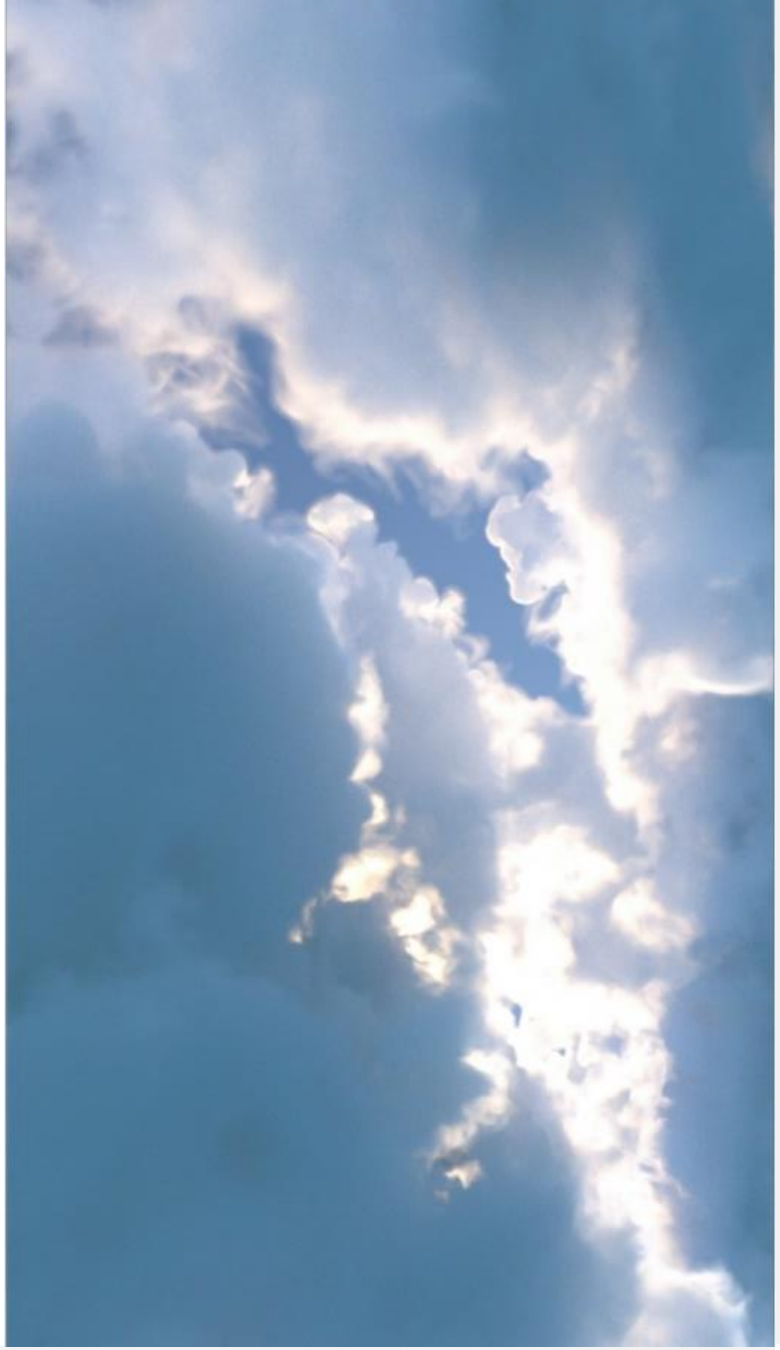
- Open VKL also provides algorithms:

  - point sampling

  - interval iterators

  - hit iterators

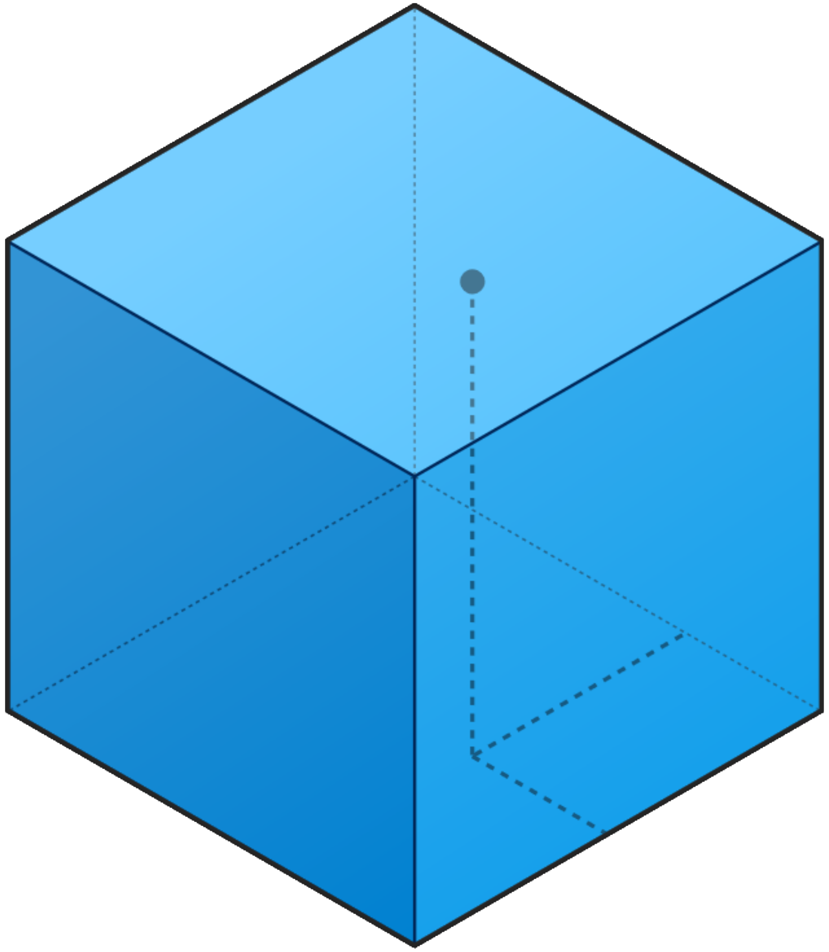
- Scientific visualization & production rendering!



Overview

1. Introduction to Open VKL
2. Installing the library
3. Demo
4. A simple example
5. Wrap up

3D Volumetric Fields



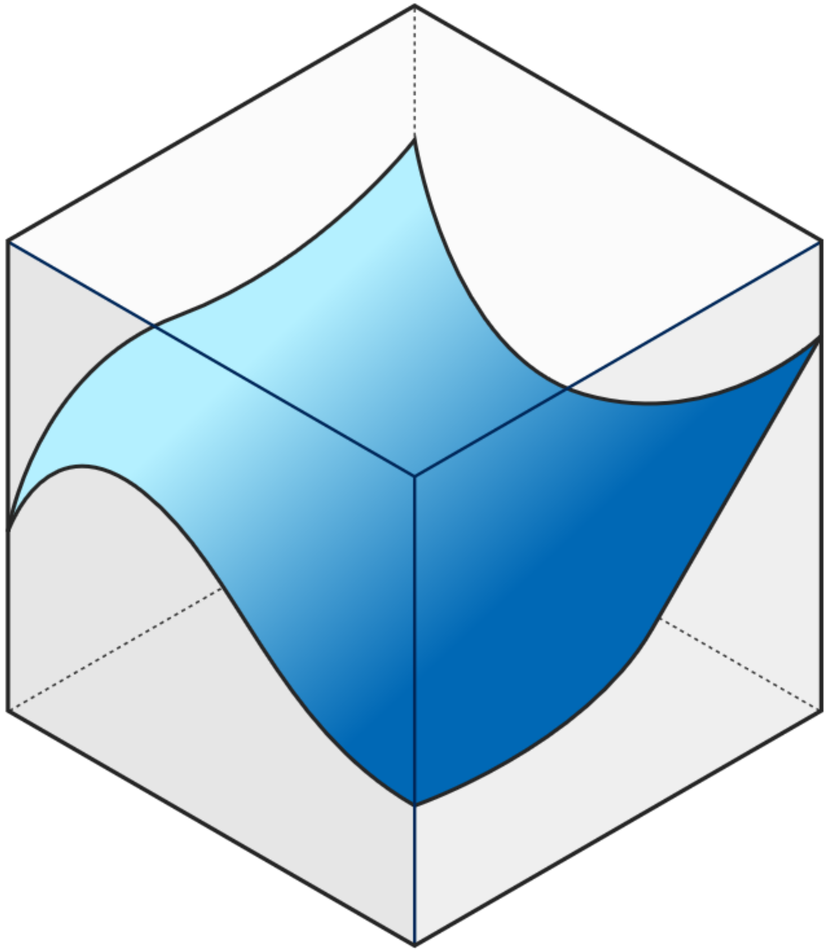
Some quantity that can be measured at a 3D position.

Scalar (density, temperature), vector (velocity)

$$(x_0, y_0, z_0) \mapsto \rho_0$$

Point sampling: **what is the field value** at a position?

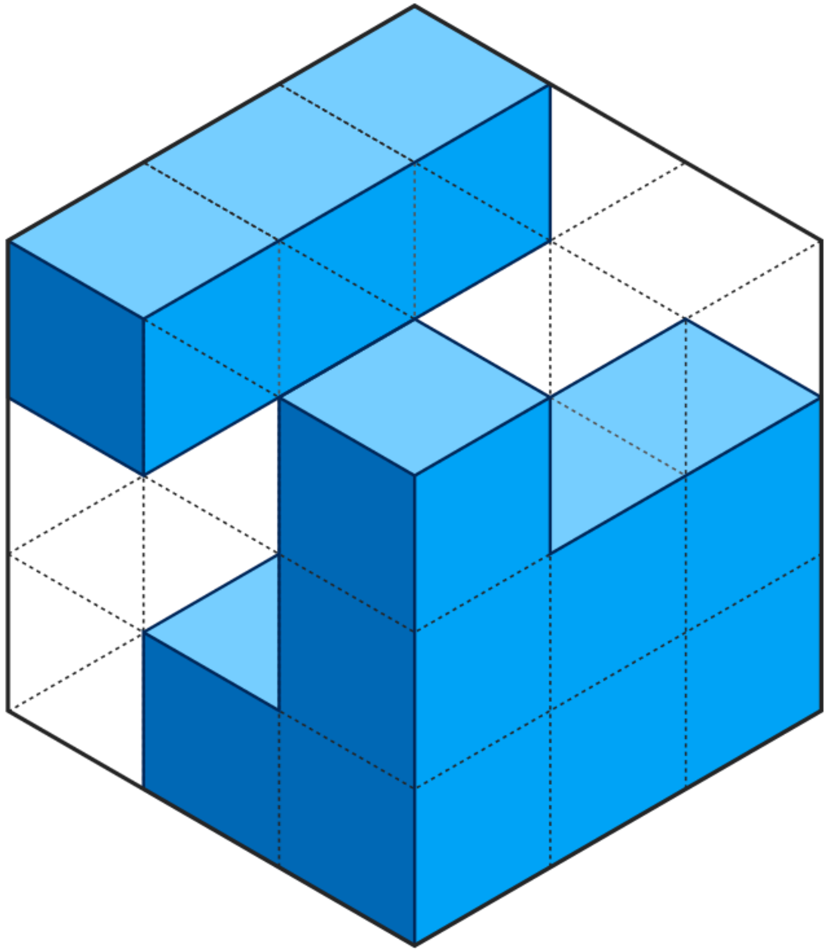
Isosurfaces



Where does the field have a given value?

Open VKL provides **hit iterators** to answer this question!

Discrete Volumetric Data



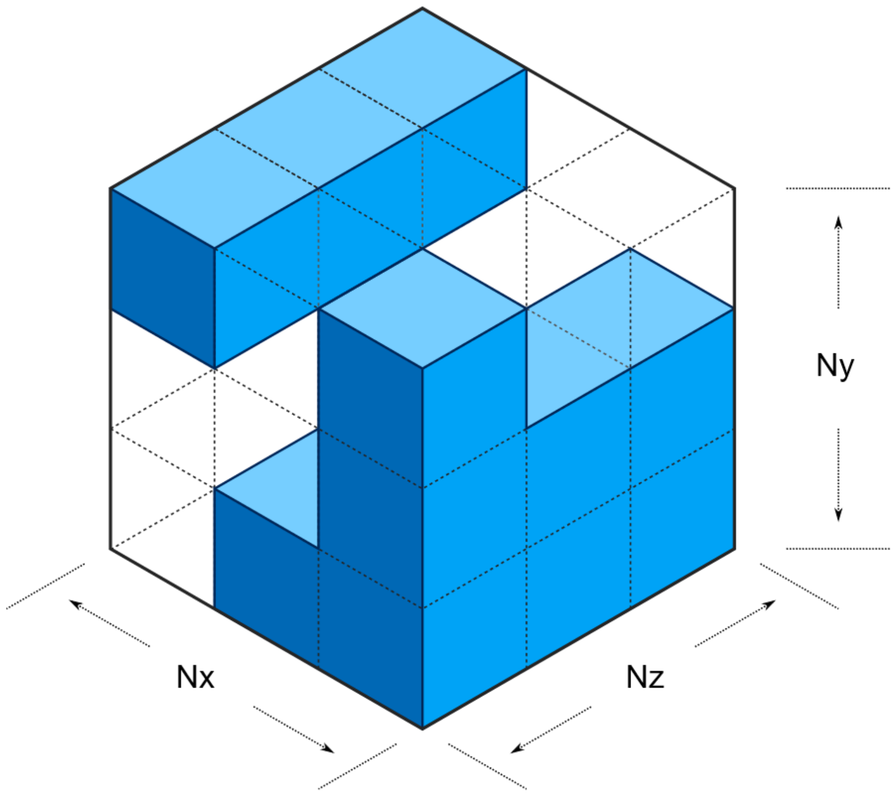
Simulation, content creation: discrete data

Voxels
Meshes (tetrahedron, etc.)
Particles

Use filters to reconstruct continuous field:
Nearest, trilinear, tricubic.

Data Structures in Open VKL

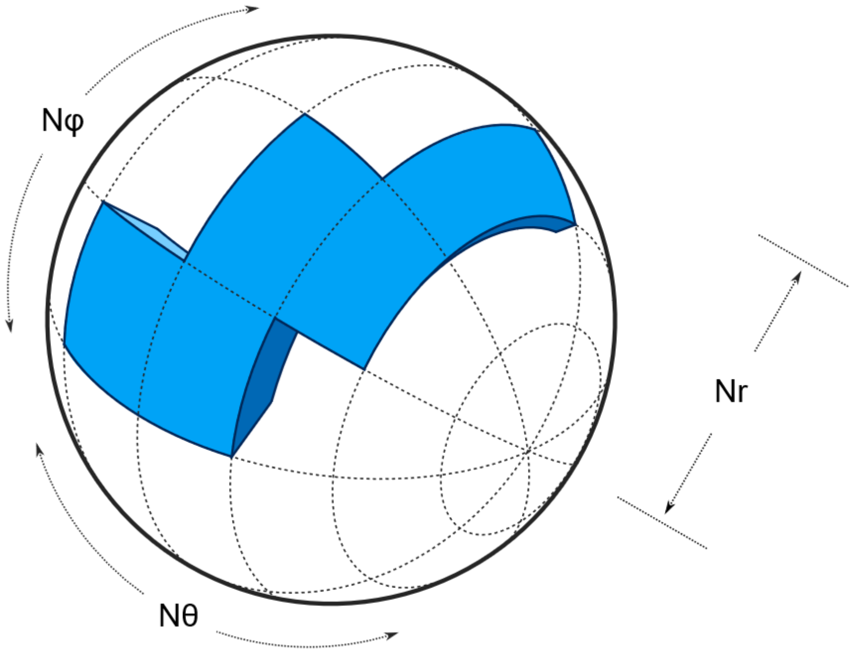
Structured Regular



- Dense 3D array.
- Very fast.
- Good fit for dense data.

Data Structures in Open VKL

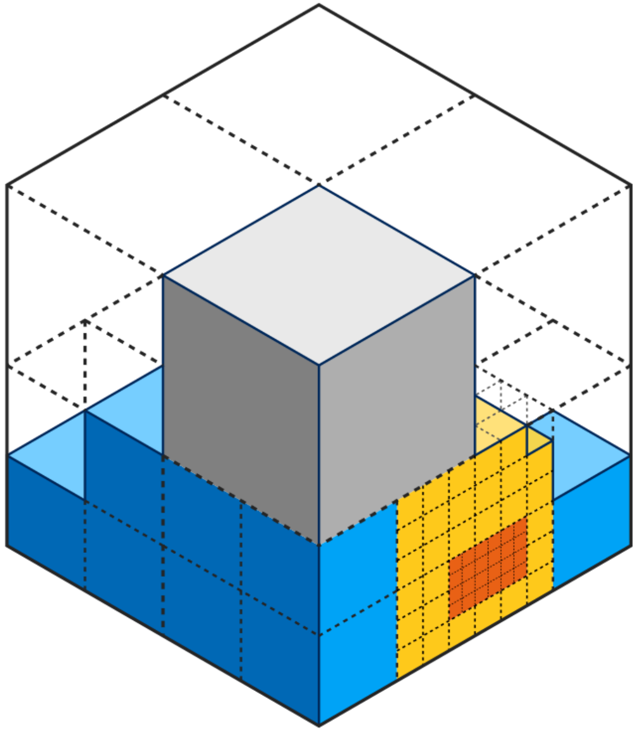
Structured Spherical



- Dense 3D array in spherical domain.
- Very fast.
- Good fit for spherical, dense data.

Data Structures in Open VKL

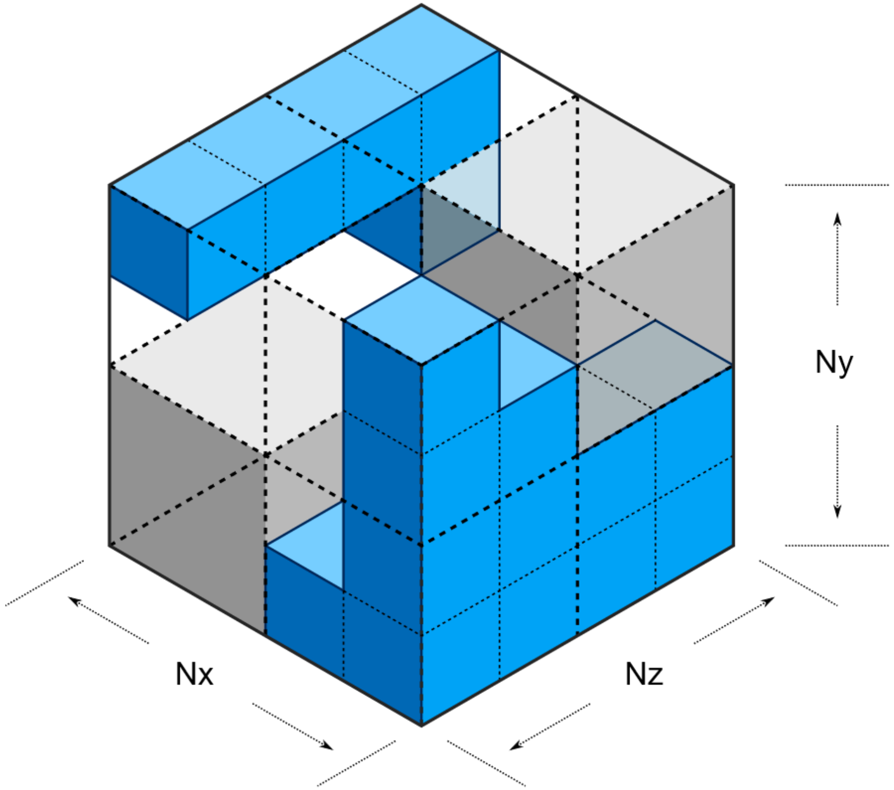
Adaptive Mesh Refinement



- Hierarchical data structure.
- Good for sparse data.
- “Unlimited detail”.
- Slower than regular grids.

Data Structures in Open VKL

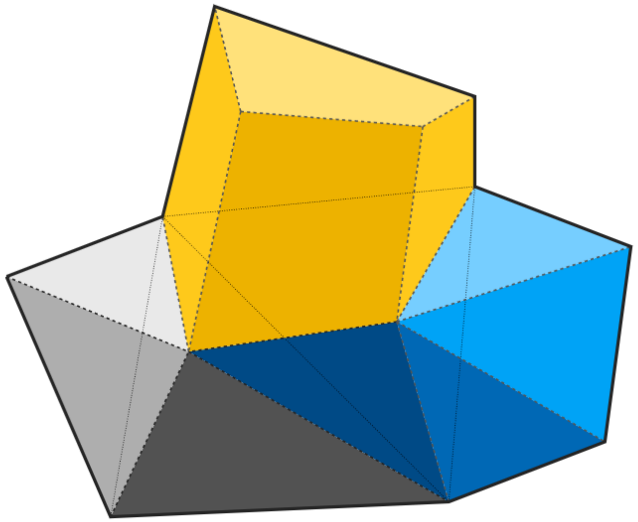
VDB



- Hierarchical data structure.
- Good for sparse data.
- Fixed depth, but huge domain.
- Faster than AMR, slower than regular grids.

Data Structures in Open VKL

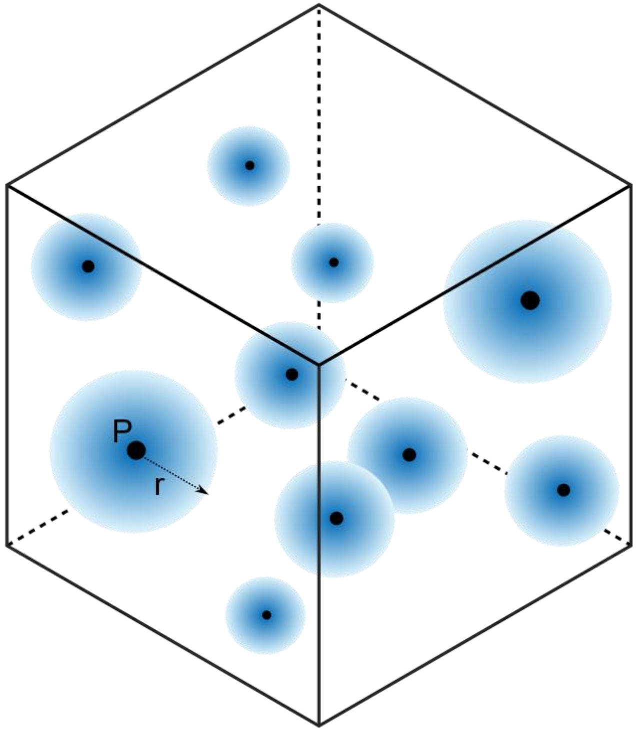
Unstructured



- Very flexible.
- Adapts to surfaces.
- Slower sampling than grid-based structures.

Data Structures in Open VKL

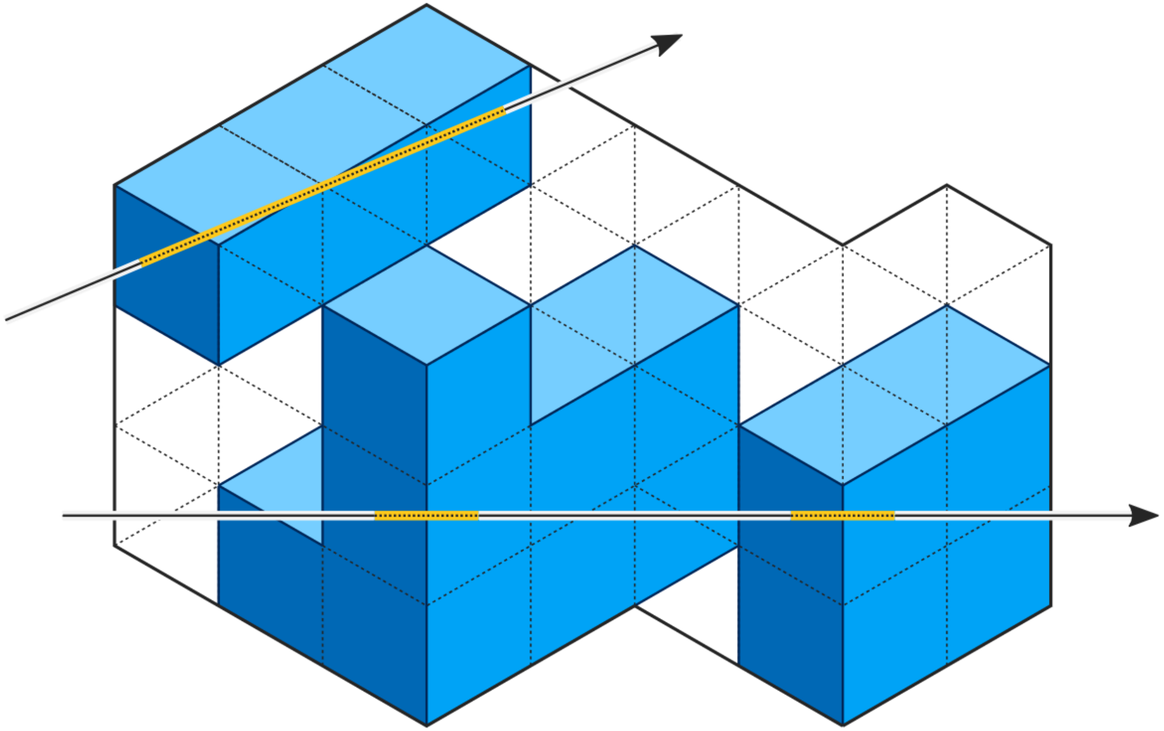
Particle



- Radial basis functions around input particles.
- Handles large data sets.
- Great if a simulation outputs particles.

Iterators

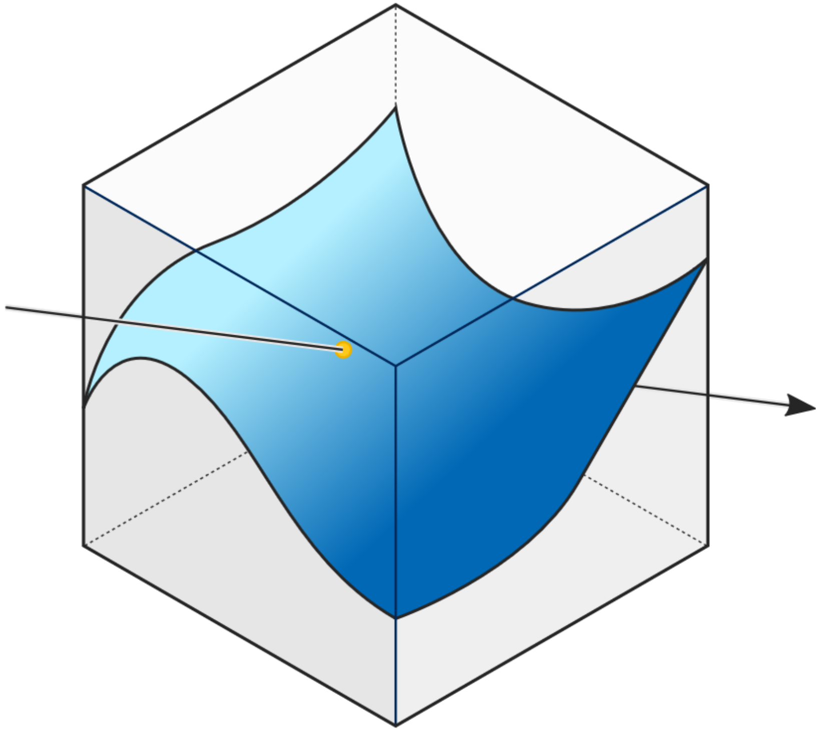
Interval Iterator



- Enumerate meaningful intervals along the ray
- Value range per interval
- Useful for ray marching

Iterators

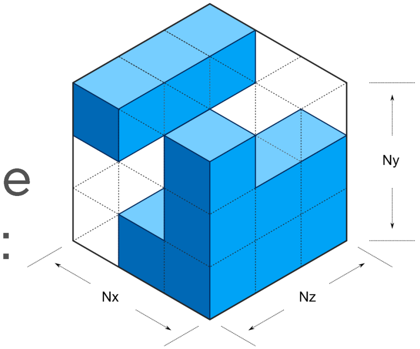
Hit Iterator



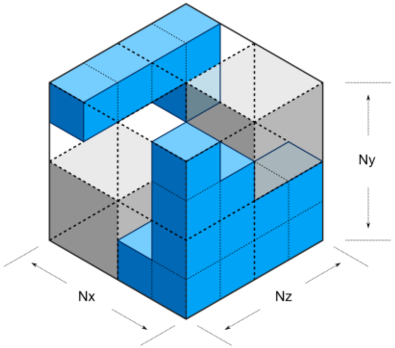
- Intersect isosurfaces.
- Multiple isosurface values can be specified.
- Sample gradient to obtain surface normal.

Recap: Open VKL Features

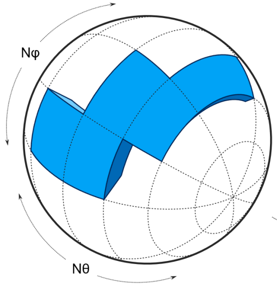
Volume types:



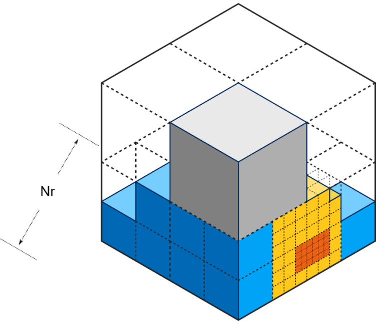
Structured



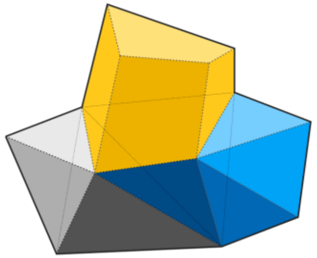
VDB



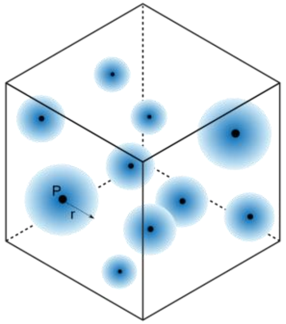
Spherical Structured



Adaptive Mesh Refinement



Unstructured



Particle

APIs:

- Sampling
- Gradient computation
- Ray-based interval iteration
- Implicit Isosurfacing
- Volume Observers

Recap: Open VKL Features

- Direct Support for Motion Blur
  - Optimal Performance with Structured Temporal Data
  - Flexibility and Memory Savings with Unstructured Temporal Data

APIs: Sampling | Gradient computation | Ray-based interval iteration | Implicit Isosurfacing | Volume Observers

Building Open VKL

- Get the latest source code from https://github.com/openvkl/openvkl/releases
- We recommend using our **cmake superbuild**.

```
$ tar -xf openvkl-0.13.0.tar.gz
$ mkdir build
$ cd build
$ cmake ../openvkl-0.13.0/superbuild
$ make -j
```

- The superbuild downloads all relevant dependencies and installs them to `build/install`.
- Use `-DCMAKE_INSTALL_PREFIX=/path/to/install/prefix` to install into a custom location.

Too Busy to Build?

- We provide binaries on our GitHub* release page: https://github.com/openvkl/openvkl/releases

- There are also oneAPI packages for many package managers.

- Develop in the Cloud: https://devcloud.intel.com

Hands On: Demo

A quick look at the vklExamples application.

Hands On: Minimal Example

Basic API concepts, initialization, structured regular volumes.

What to do next?

- Check out the Open VKL source: https://www.openvkl.org
- Try the superbuild.
- Play with our examples.

Please reach out if you have any questions!

intel®

Notices and Disclaimers

- No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
- Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.
- You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.
- The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.
- Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at [intel.com].
- Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit www.intel.com/benchmarks.
- Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.
- Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.
- Intel, Core and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.
- *Other names and brands may be claimed as the property of others
-