



SIGGRAPH 2023

LOS ANGELES+ 6-10 AUG

THE PREMIER CONFERENCE & EXHIBITION ON
COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

OPENVDB

KEN MUSETH, NVIDIA



Schedule

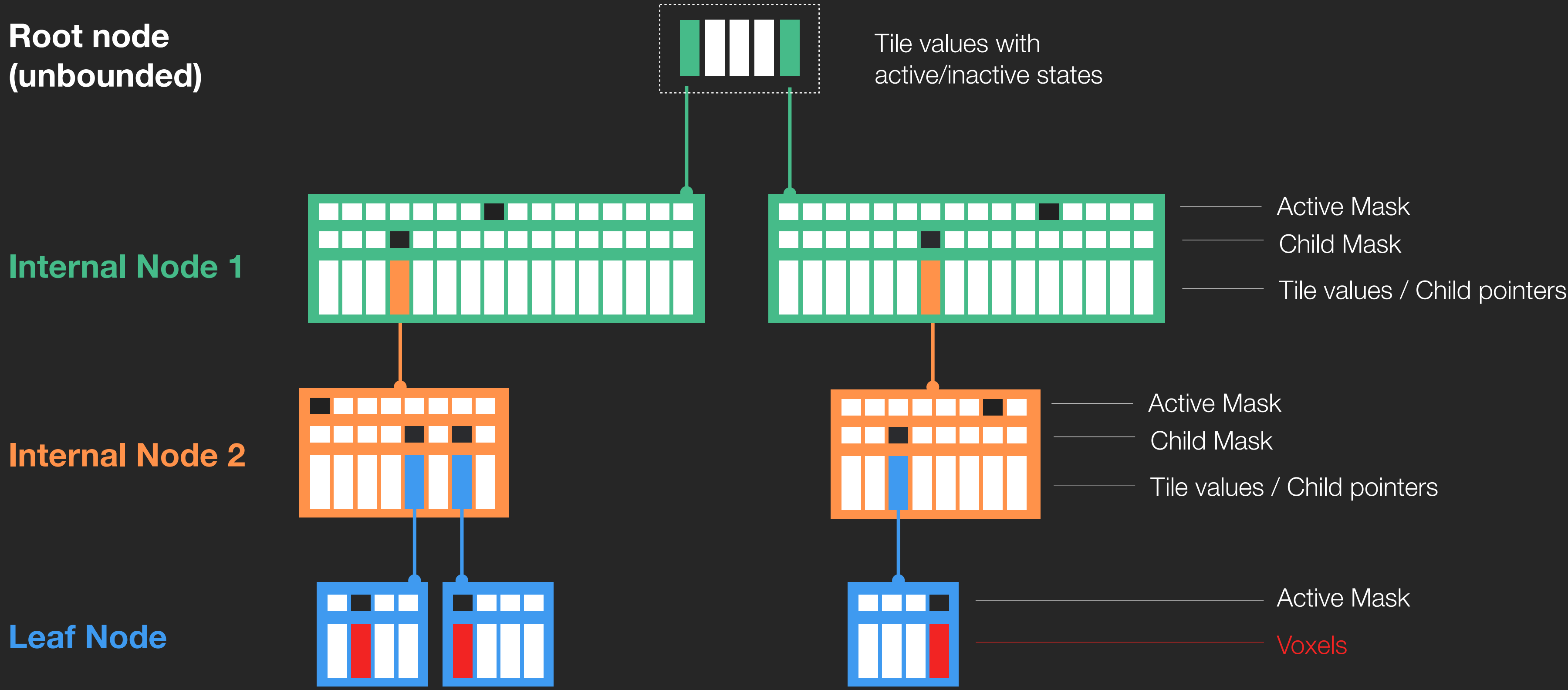


- **Introduction:** Ken Museth (NVIDIA, TSC Chair)
- **Multithreading in OpenVDB:** Dan Bailey (ILM, TSC)
- **Constructing NanoVDBs on the GPU:** Greg Klar (NVIDIA)
- **Developing a Fluid Solver in OpenVDB:** Andre Pradhana (NVIDIA, TSC)
- **Particle Surfacing:** Nick Avramoussis (WETA/UNITY, TSC)
- **Putting the Sign back into SDFs:** Jeff Lait (SideFX, TSC)
- **Mathematica Bindings:** Gregory Hurst (UT, TSC)
- **OpenVDB in Production:** Jeff Budsberg (DWA)



OpenVDB

Sparse Volume Tools build on VDB Data Structure



[K. Museth, SIGGRAPH / ACM TOG, 2013]

What's new ...



- **Version 10, October 2022**
 - NanoVDB supports of multiple channels with arbitrary values
 - Convex clipping in Houdini
 - Mathematica bindings
 - New rasterization tools
 - vdb_tool, command-line alternative to DCC

vdb_tool



- **Convert a mesh into SDF with fixed resolution**

```
vdb_tool -read mesh.obj -mesh2ls dim=512 -write sdf.vdb
```

- **Convert a mesh into SDF with resolution defined from another VDB**

```
vdb_tool -read bunny.vdb dragon.ply -mesh2ls voxel='{0:voxelSize:2:*}'
```

- **Meshing of fluid particles and pipe SDF to viewer**

```
vdb_tool -read points.abc -points2ls -dilate -gauss -erode -o stdout.vdb | vdb_view
```

- **Render thumbnails of all SDFs in a directory structure**

```
vdb_tool -each file=`find . -name '*.vdb'` -read '{$file}' -for  
grid=0, '{gridCount}' -if '{$grid:isLS}' -render vdb='{$grid}'  
thumbnail_ '{$grid:gridName}'.ppm image=256x256 keep=1 -end -end -clear -end
```




```
vdb_tool -read mesh_mask.obj -mesh2ls voxel=0.1 width=3 -for n=200,300,1 -read points_{$n:4:pad0}.vdb  
-vdb2points -points2ls voxel=0.035 radius=2.142 width=3 -dilate radius=2.5 space=5 time=1  
-gauss iter=2 space=5 time=1 size=1 -erode radius=2.5 space=5 time=1  
-ls2mesh vdb=0 mask=1 adapt=0.005 -write mesh_{$n:4:pad0}.abc -end
```




[https://github.com/AcademySoftwareFoundation/
openvdb/tree/master/openvdb_cmd/vdb_tool](https://github.com/AcademySoftwareFoundation/openvdb/tree/master/openvdb_cmd/vdb_tool)

Things to come ...



- **Version 10.1 (soon) & 11 (Oct 2023)**
 - Improved techniques for surfacing particles
 - Many improvements to NanoVDB
- **Version 11+ (still under evaluation)**
 - cuda tools in NanoVDB: pts -> LS -> mesh, re-norm, dilate/erode
 - Streaming OpenVDB grids (rendering at any resolution)
 - Multi-resolution grid from Autodesk
 - NeuralVDB



NanoVDB

Introducing new NanoGrid types



- **IndexGrid and OnIndexGrid!**
 - Dense or sparse indexing to side-cars
 - Compatible with ANY value/voxel type!
 - Supports multiple channels of any type!
 - Memory-light since they have no payload!
 - Can be-reused since they separate topology and values

Generating NanoGrids on the GPU



- **Build grids of any type**
 - Input: a list of voxels on the device
 - Output: a nanovdb grid of any type
- **Build BVH for points**
 - Input: list of points in world space on the device
 - Output: PointGrid that serves as a BVH with points attributes
 - Supports relative voxel-offsets in 8 or 16 bit representations

CUDA accelerates tools

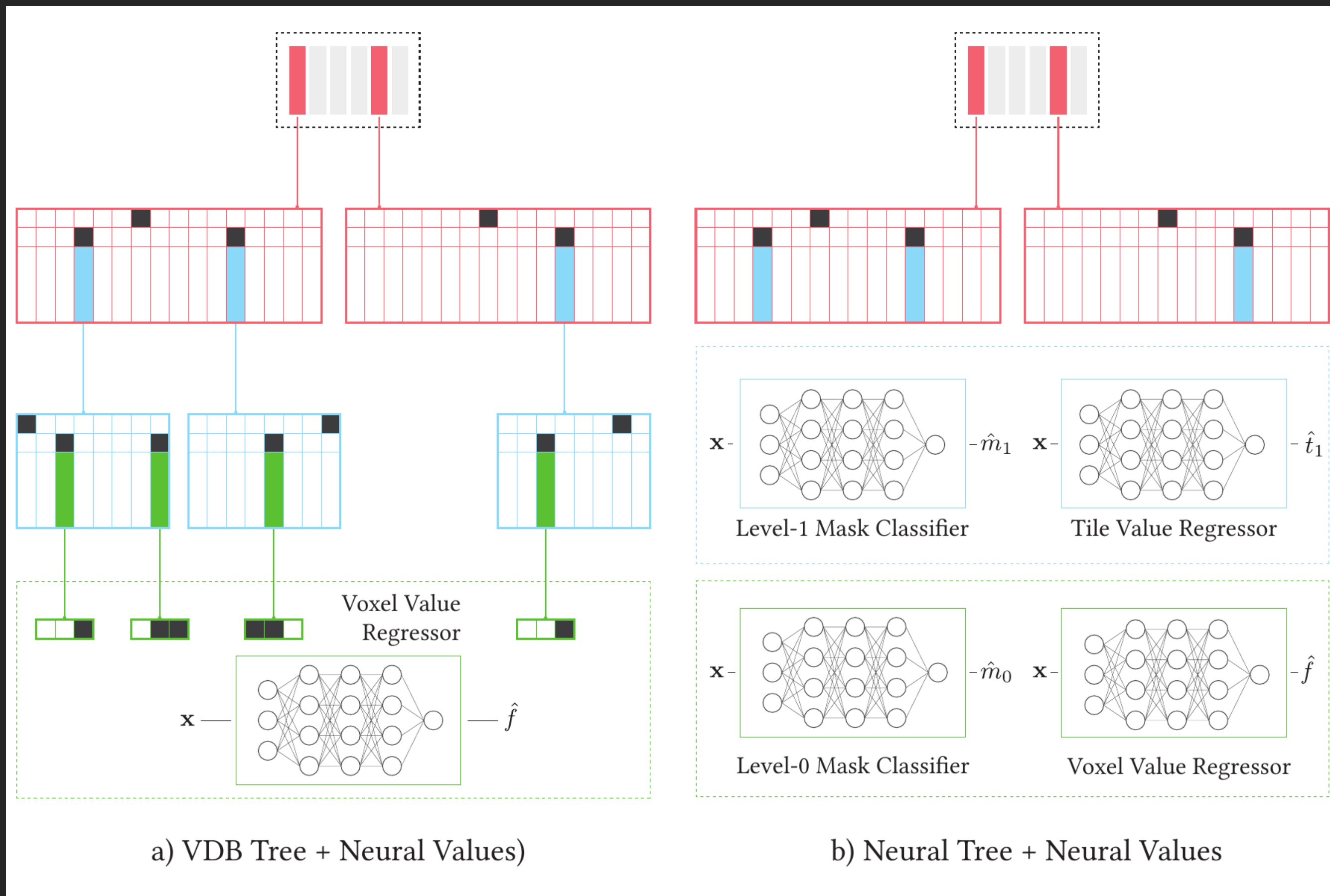


- `nanovdb::cudaSignedFloodFill`
- `nanovdb::cudaPointsToGrid`
- `nanovdb::cudaVoxelsToGrid`
- `nanovdb::cudaIndexToGrid`
- `nanovdb::cudaAddBlindData`
- `nanovdb::cudaDilateActiveVoxels`
-



PR#1651

NeuralVDB



OpenVDB: 3.8 GB
OpenVDB: 1.5 GB (half & Blosc)
NeuralVDB: 25 MB

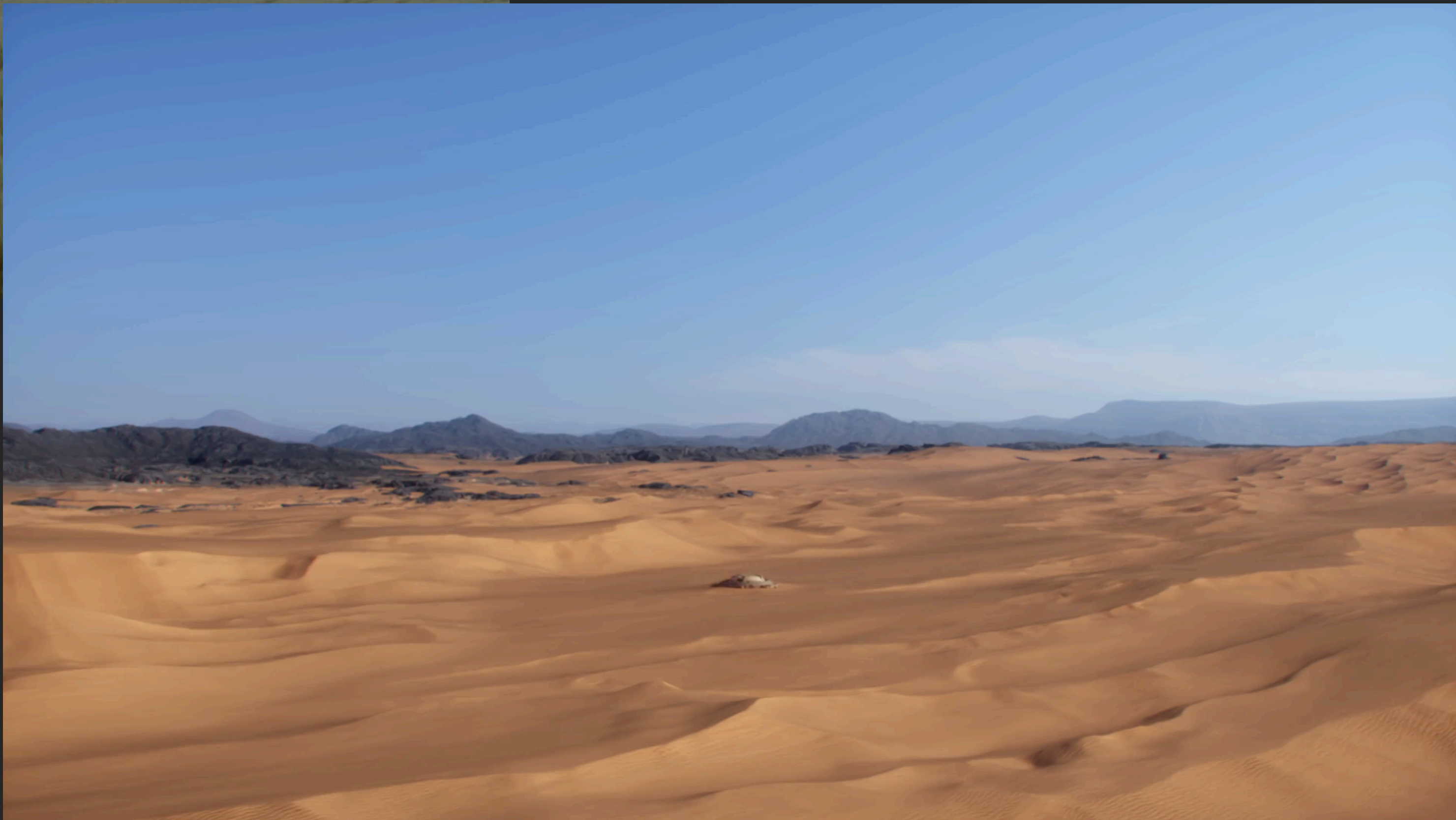


“NeuralVDB”, D. Kim, M. Lee, K. Museth, arXiv

News on NeuralVDB



- Early access program
- Auto-configuration of hyper-parameters
- 2X faster decoding with new backend (CUTLASS vs PyTorch)
- CPU fallback for encoding and decoding
- Other new features
 - More temporal smoothing options
 - Improved compression from pruning of near-zero voxels
 - New feature mapping based on InstantNGP (experimental)



NeuralVDB on Complex Geometry



15.3X compression



OpenVDB