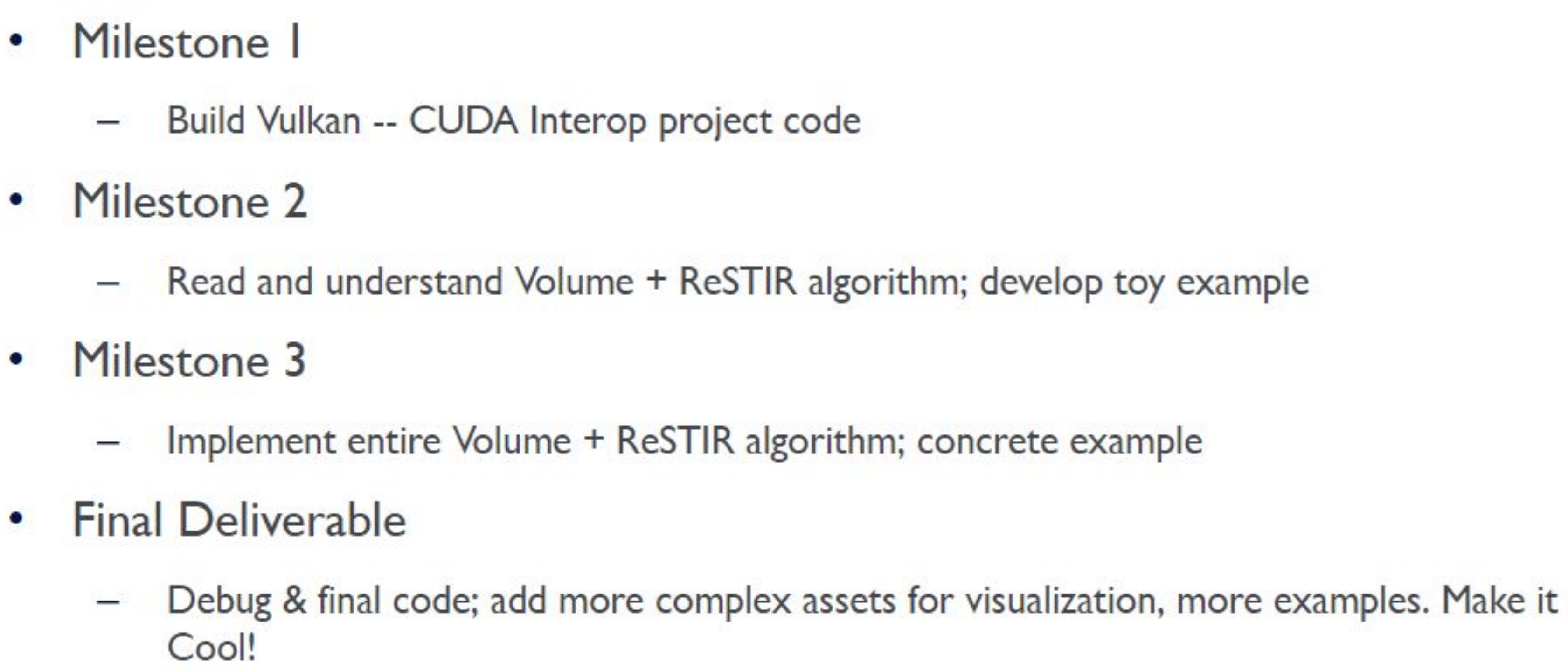
Fast Volume Rendering with Spatiotemporal Reservoir Resampling

Zhihao (ruanzh@seas.upenn.edu)
Shubham (sshubh@seas.upenn.edu)
Raymond (rayyang@seas.upenn.edu)

November 29, 2021

Penn Engineering

Milestones

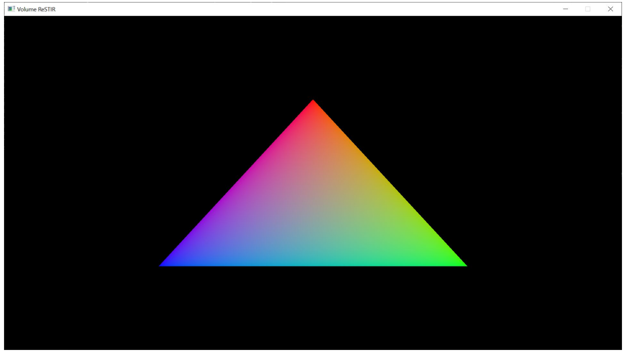


- Milestone 1
  - Build Vulkan -- CUDA Interop project code
- Milestone 2
  - Read and understand Volume + ReSTIR algorithm; develop toy example
- Milestone 3
  - Implement entire Volume + ReSTIR algorithm; concrete example
- Final Deliverable
  - Debug & final code; add more complex assets for visualization, more examples. Make it Cool!

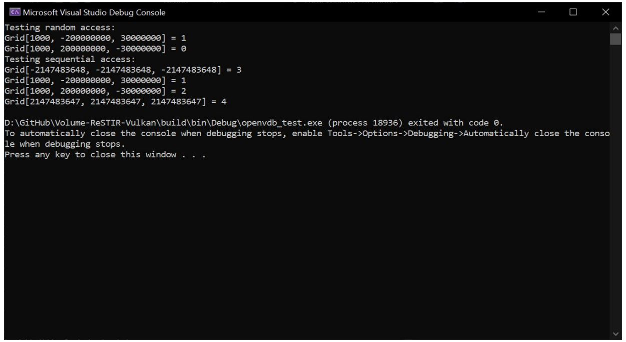
Penn Engineering

M1:Vulkan Pipeline with OpenVDB

Vulkan Pipeline with VK-Bootstrap:



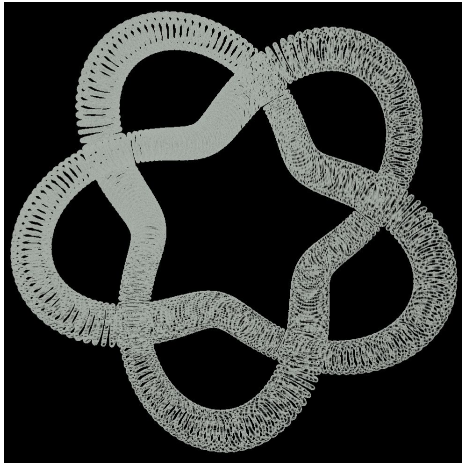
OpenVDB Integration:

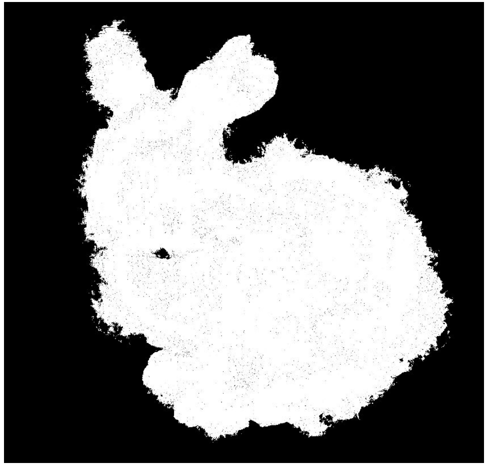


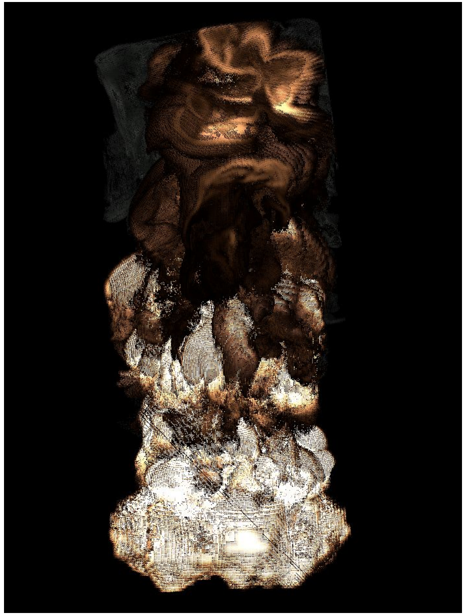
```
Testing random access:
Grid[1000, -200000000, 30000000] = 1
Grid[1000, 200000000, -30000000] = 0
Testing sequential access:
Grid[-2147483648, -2147483648, -2147483648] = 3
Grid[1000, -200000000, 30000000] = 1
Grid[1000, 200000000, -30000000] = 2
Grid[2147483647, 2147483647, 2147483647] = 4

D:\GitHub\Volume-ReSTIR-Vulkan\build\bin\Debug\openvdb_test.exe (process 18936) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

M2: Complete

Vulkan Volumetric Rendering of VDB Assets:

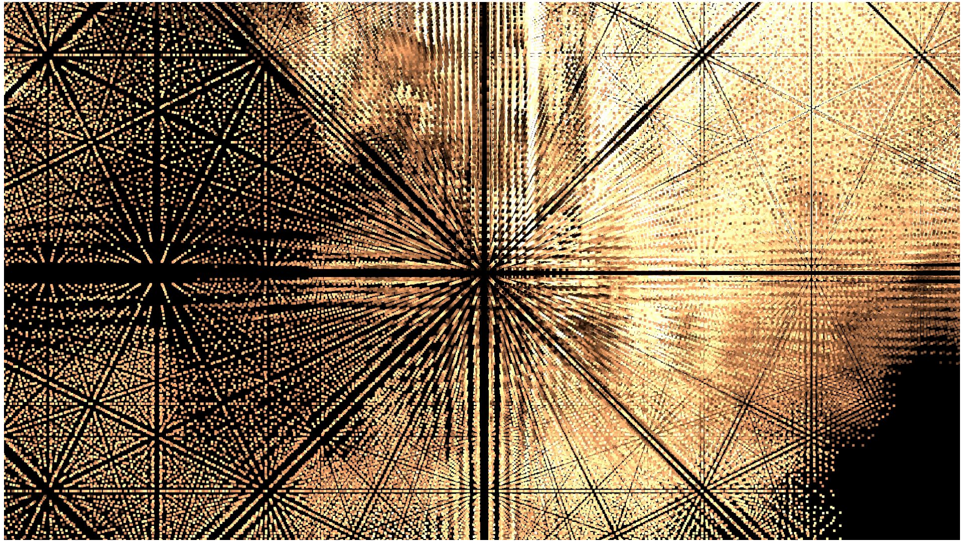


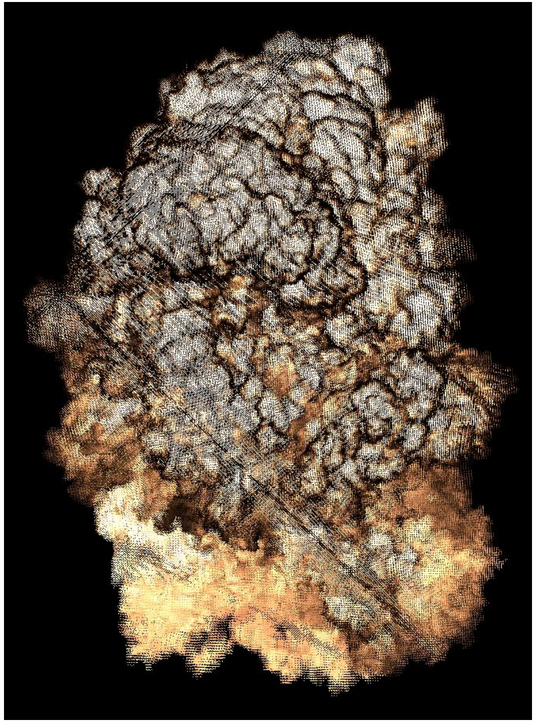




M2: Complete

Vulkan Volume Rendering of VDB Assets:



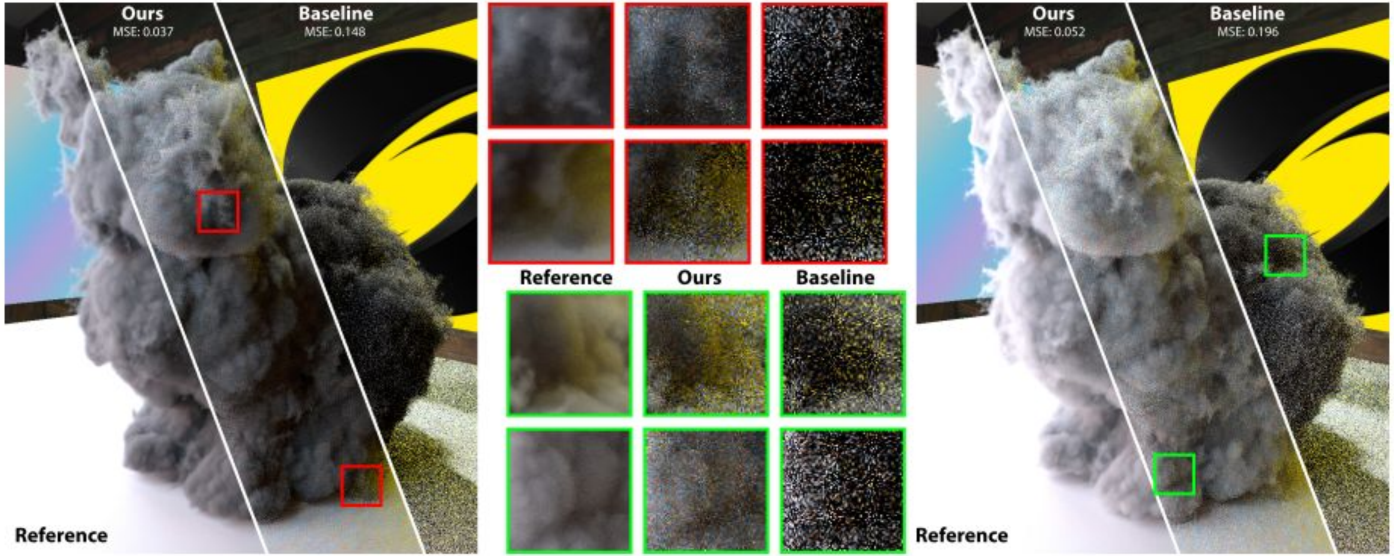


Penn Engineering

M2: Incomplete

- Basic Fast Volume Rendering with ReStir with working toy example

M3: Full Volume Rendering with RESTIR



References

[1] Volume Rendering

[2] Volume Rendering (Nvidia)

[3] Fast Volume Rendering with Spatiotemporal Reservoir Resampling (SIGGRAPH 2021)

[4] Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting (SIGGRAPH 2020)

[5] VK-Bootstrap

[6] OpenVDB