DXRPlayground

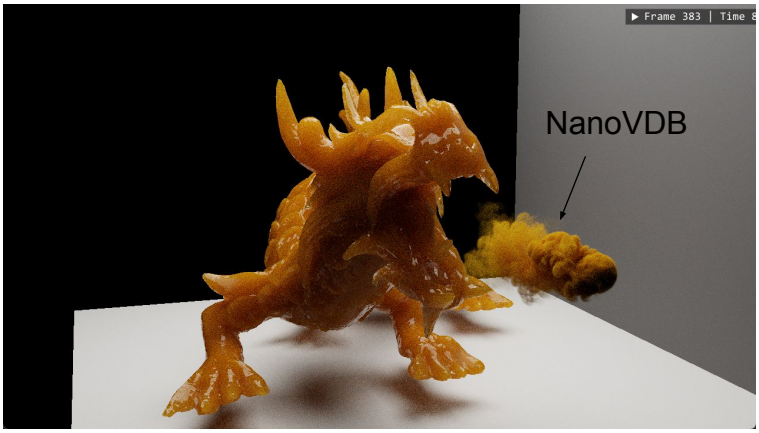
@kenhu07
レイトレ合宿11

DXR習い始めたのアプリが思ったより長生き（進化が遅いということ）

概要

- Volume Rendering (Delta Tracking)
- NanoVDB

- 制限時間 180s
  - 384spp
  - 24FPS x 8s



ほぼbrute force

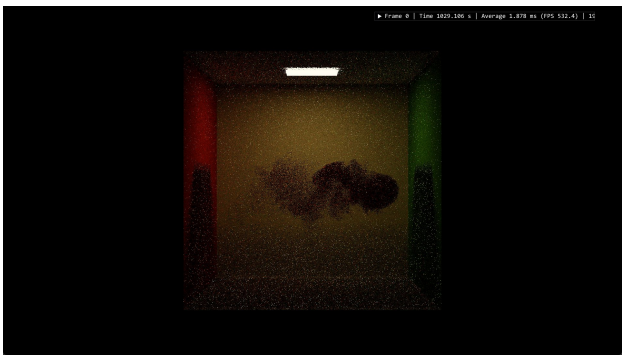
Library & Asset

- Library
  - Hardware Raytracing: **DirectX (DXR)**
  - Math: **glm**
  - Texture: **DirectXTex**
  - Scene: **tinyxml2**
  - Mesh: **tinyobjloader**
  - Camera Animation: **tinygltf** (Blenderで作成)
  - Noise Texture: **TileableVolumeNoise**
  - VDB: **NanoVDB** (unreal-vdbを参考)
- Asset
  - Stanford Asian Dragon
  - OpenVDB Smoke1

tiny系違法建築

NanoVDB

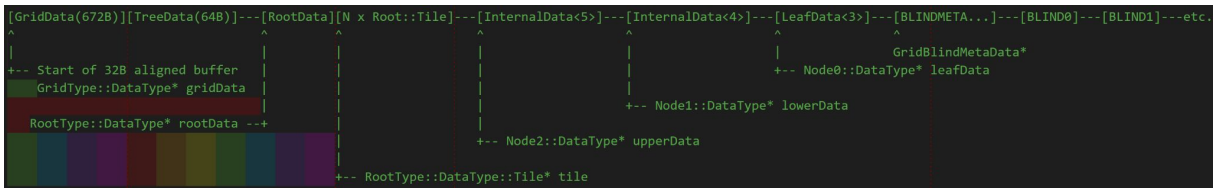
- Densityサンプリングしただけで、あまり活用できていない
- 111 x 222 x 112 in float
- Tree
  - 32^3->16^3->8^3
  - 1->2->3117
- NanoVDB vs. 3D Texture (DDS)
  - Memory: 6.69 MB vs. 10.5 MB
  - Time: 1.84ms vs. 1.21ms
  - Time (cache): 1.53ms vs. 1.21ms
    - `pnanovdb_readaccessor_t`



3D Textureは勝ててほしかった

NanoVDB

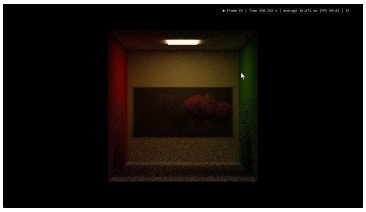
- 3DデータにおいてVDBより汎用フォーマットがないため便利な道具
- Tree 構造そのままのようで、もっと積極的に利用できそう
  - API に `pnanovdb_hdda_zero_crossing` が提供されてる
- ただ完全にリアルタイム向けならもっと色々コンパクトにもできそう



- ちなみに UE SparseVolumeTexture は Page Table 方式（2階層）
- データ圧縮という意味で将来的に全部 Neural になりそうだけど...

GPU向けにしは複雑すぎない？

気持ち程度の最適化

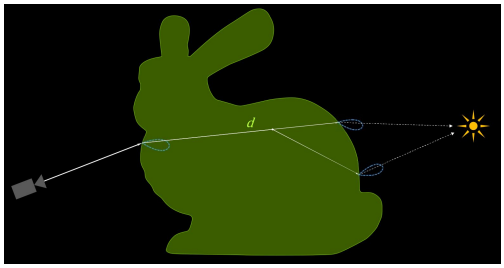


- **Delta Tracking**
  - Null scattering その場で loop する（最適化？）
    - 最初は path tracing の loop 流用してた、毎回 ray 飛ばすのも無駄が大きい
    - 特に NanoVDBの ほう空白多いため、loop 上限にひっかかりやすい
  - Ray がメッシュにヒットしても、scattering しないと確定するまでメッシュ情報読まない
    - フレーム時間 ⅛ に減るケースがある
- **NanoVDB**
  - Smoke1 Density Max 5.0+ だけどほとんど 2.0 未満でクランプしてしまった
- **コード削減**
  - 使われていない BSDF、デバッグ出力、Texture (UV) 無効化など
  - DXIL Instruction Count 8000+ => 2000+
- **メモリアクセス** (Long Scoreboard)
  - メッシュを 7万ポリゴン に reduction、Position を fp16 落としている
  - 無駄な groupshared 作って occupancy 制限するテクニックを使うと更に速くなる ...

bruteforceなアルゴリズムしかできていないので細かいところでパフォーマンスを稼ぐ

今後

- Volume Rendering 効率上げたいために調査（Houdiniプレビュー超早い...）
- リアルタイム向けSSS / Transmission 需要増えそうで、Samplingより自由に扱えるように



SIGGRAPH 2025
Real-Time Subsurface Scattering
0回、1回Scatteringまではパストレ
2回以降はDiffusion

- Radiance Cache まずシンプルな手法から入れてみる
- VDB を Volume Rendering以外での使い方も調べてみたい

リアルタイムでやりがちの 1bounceのみ、とpath tracerの無限bounceの中間地帯、今後も広がりそう
応用において、アルゴリズムだけでなく、ハードウェア特性やデータ作成方法など総合的な影響が入る