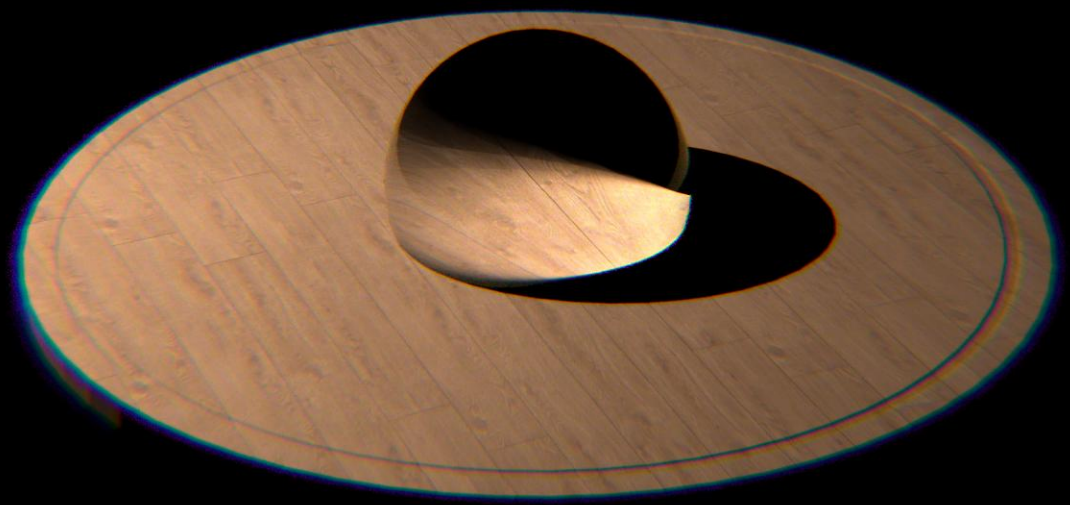
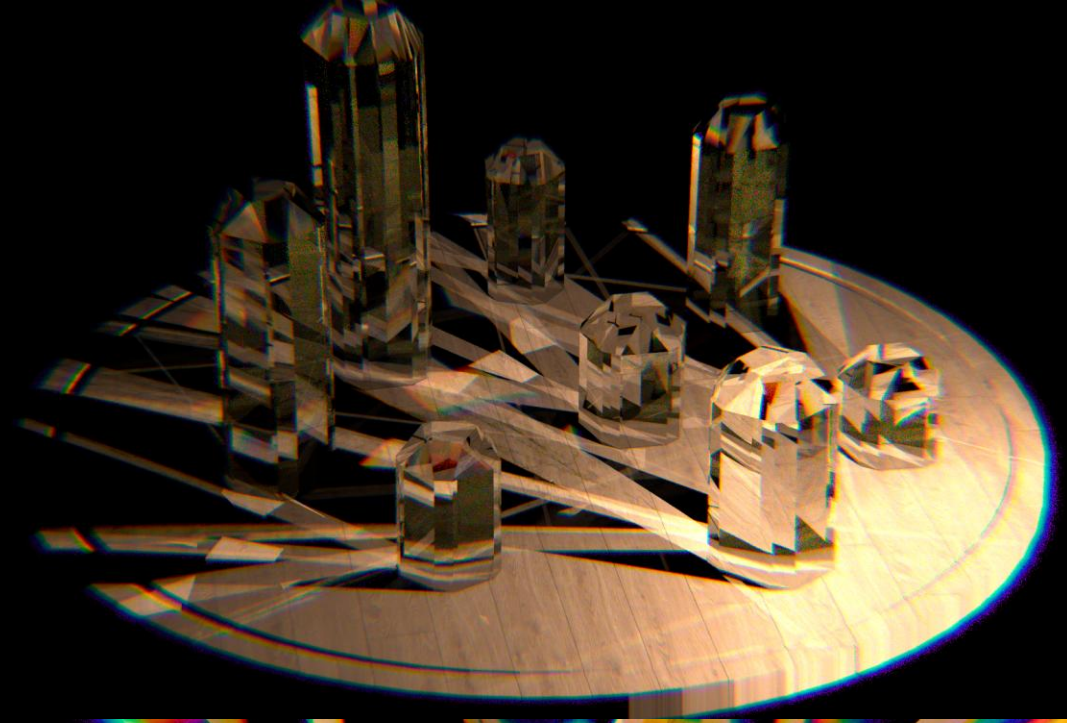
PALZILLA

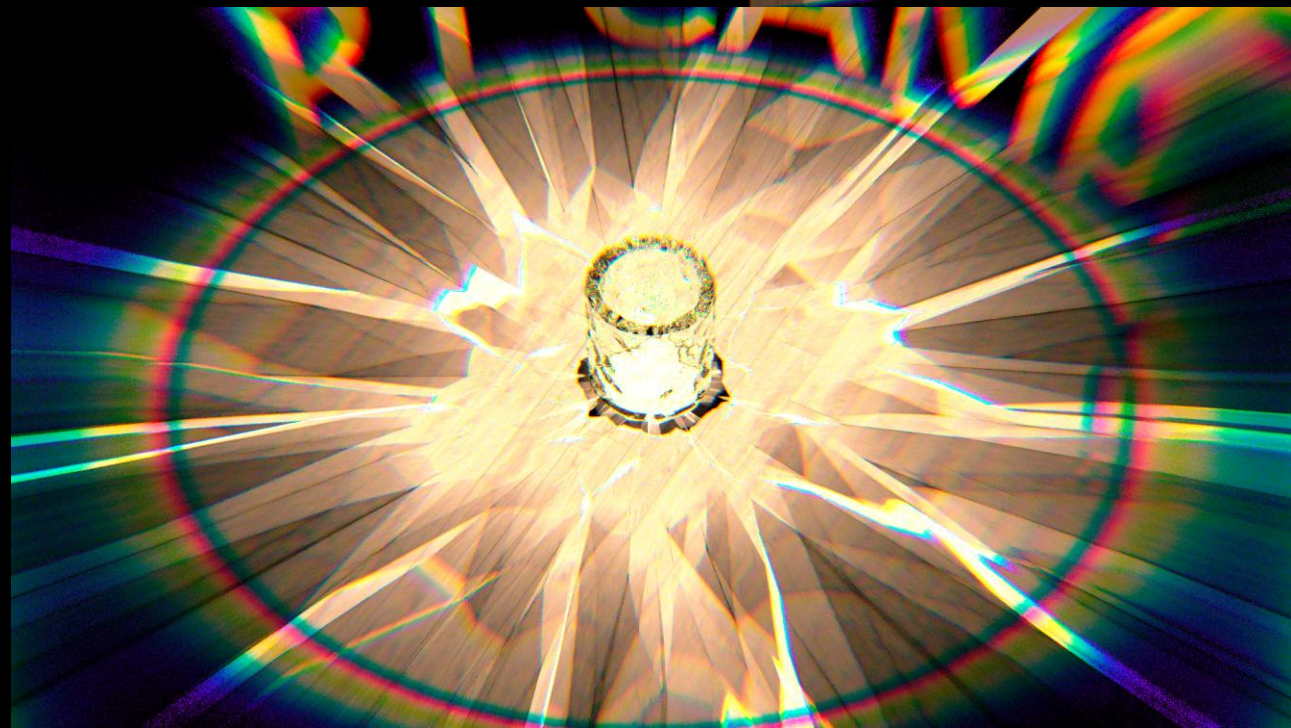
PATH CUTS 風味フルスペクトルコースティクスレンダラー

ushiostarfish

All 32spp







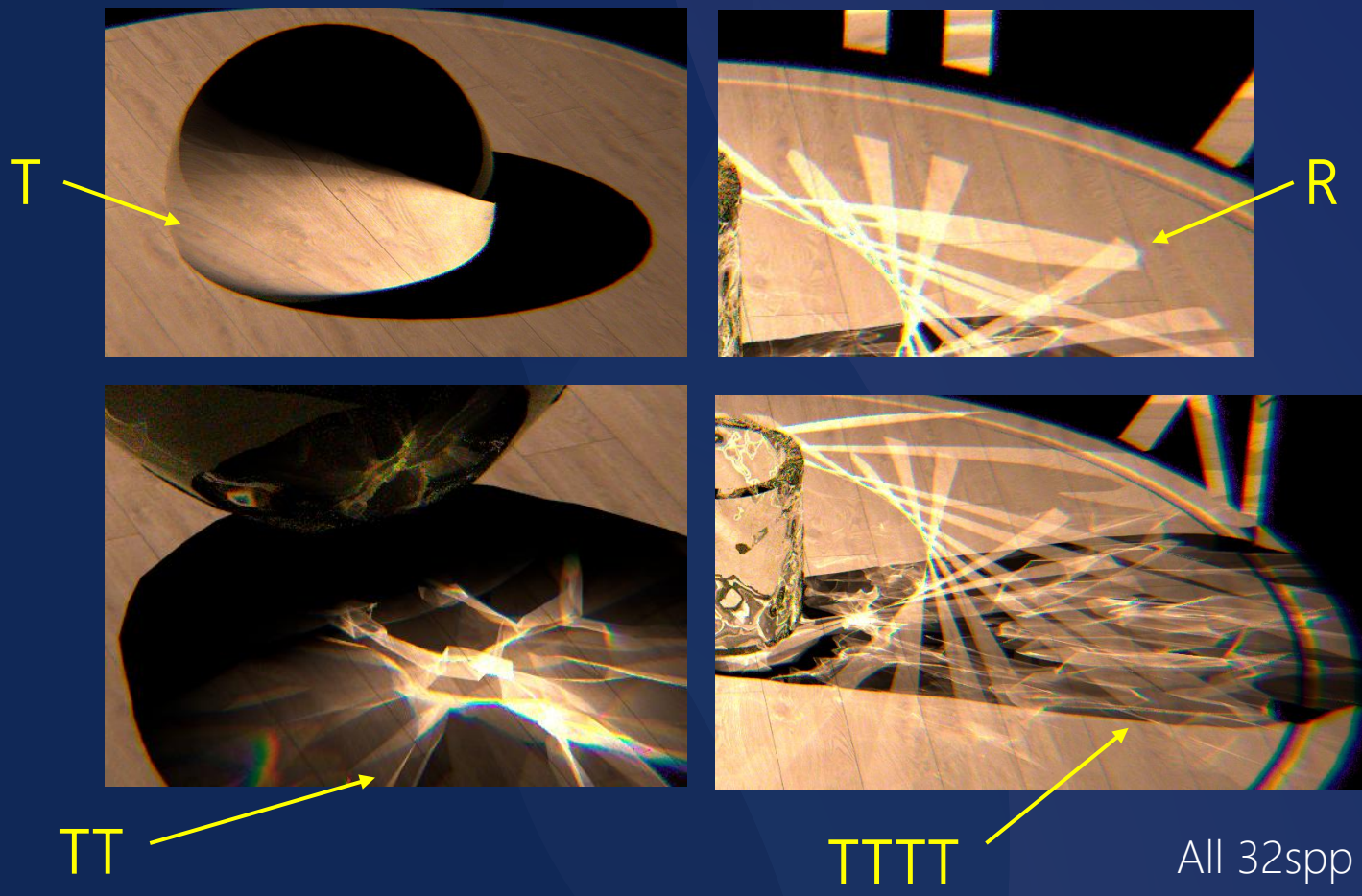
推しポイント

- Path Cuts をベースにした完全決定論的なスペキュラパスの計算
  - ポイントライトからの寄与はノイズ０
  - ノイズの原因は波長とカメラのピクセル内サンプリングだけ (HD 32spp)
- フォトントレーシングによる実装によるディープなパスのサポート
  - 今回は最長 TTTT
  - でも経路長を伸ばしても計算が爆発しない
- 線形システムソルバーにEigenなどの線形代数ライブラリ不使用
- 完全なソフトウェアレイトレ
  - 何の工夫もないバイナリLBVH (Ciprian Apetrei, Fast and Simple Agglomerative LBVH Construction)
  - Stack Freeトラバーサルをテスト
    - Ingo Wald, A Stack-Free Traversal Algorithm for Left-Balanced k-d Trees
    - Hapala et al, “Efficient Stack-less BVH Traversal for Ray Tracing”
    - コードはシンプルだが２倍遅くなったので普通にスタックフルに実装

計算している経路

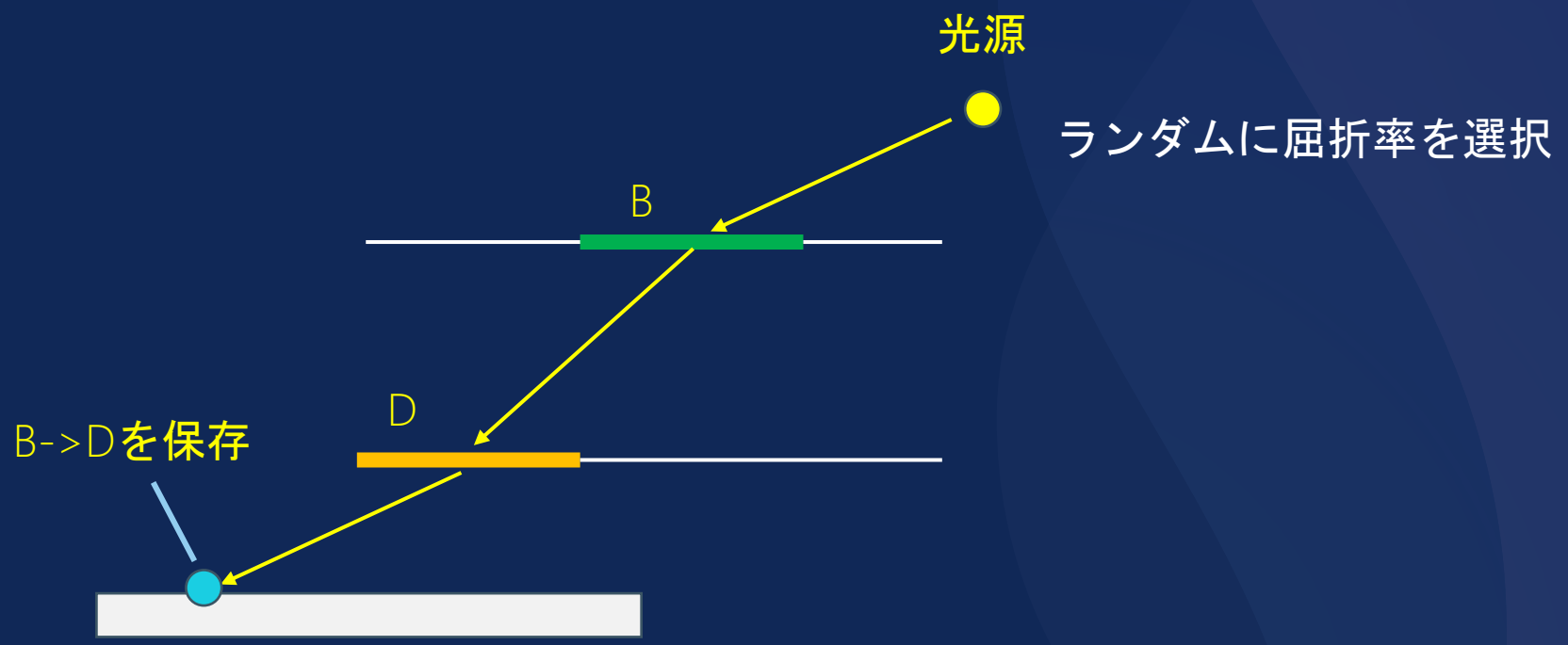
- 全部計算するとやや重いので、視覚的に重要なものだけ
  - T
  - R
  - TT
  - TTTT
  - (F103 から F153までのみ) TRT
  - (F103 から F153までのみ) TTT

※ライトからの表記



フォトントレーシングのフルスペクトル対応

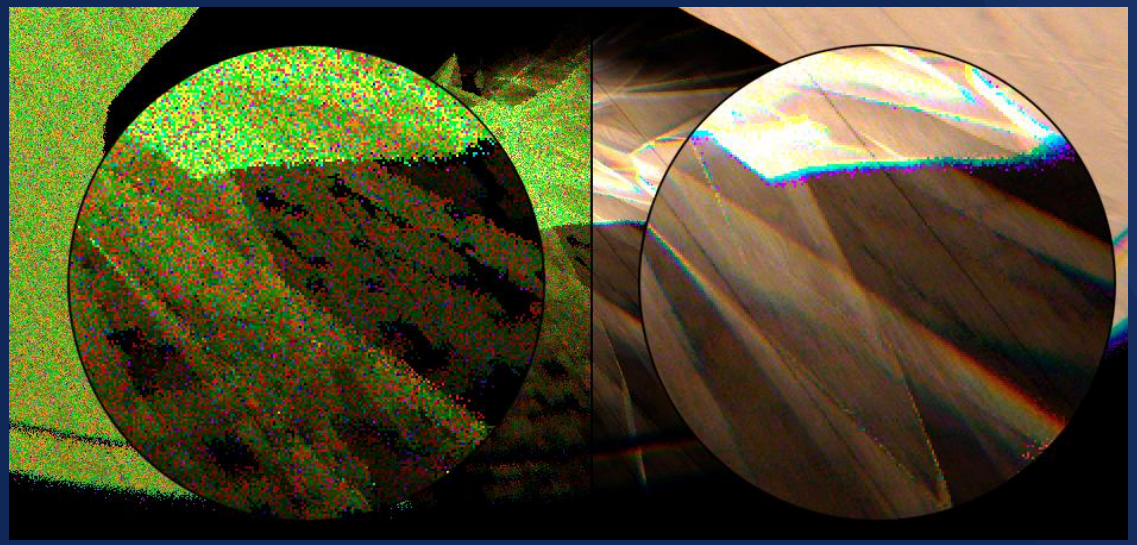
- そもそもフォトントレーシングでキャッシュするのは寄与候補のタプル
  - 単純にランダムに屈折率を選択してキャッシュを構築すればよい
  - カメラパスで再度最終寄与の波長をサンプリング、ニュートン法でパスをピクセルごとに解決可能



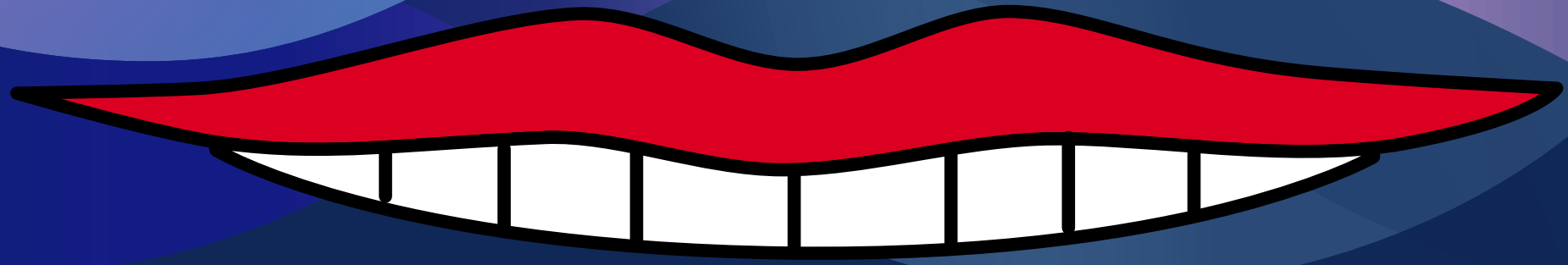
バイアス

- フォトントレーシングに十分な数が無い場合に経路を見落とす
  - 寄与が取れないのでダークニングバイアス
- 少量のフォトントレーシングをサンプルごとにやり直す
  - 視覚的なアーティファクトは低減される
  - そもそも失敗するエリアは集光しないエリアなのでどうせ暗い

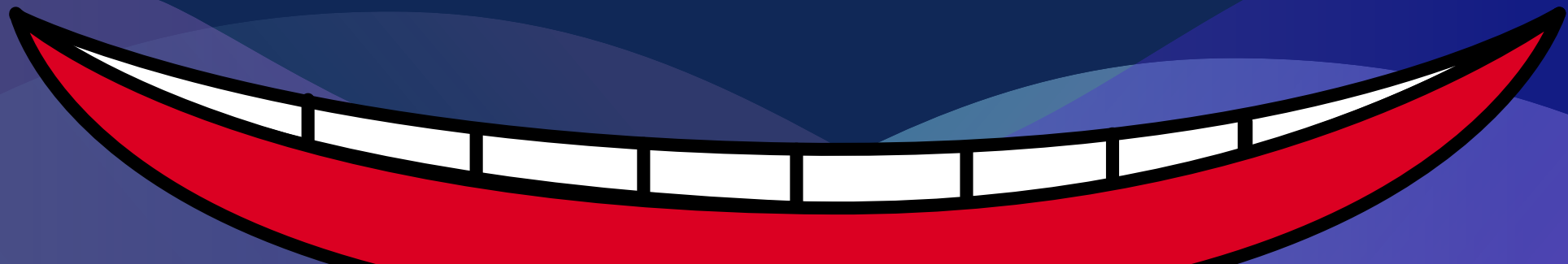
1 SPP　　32 SPP



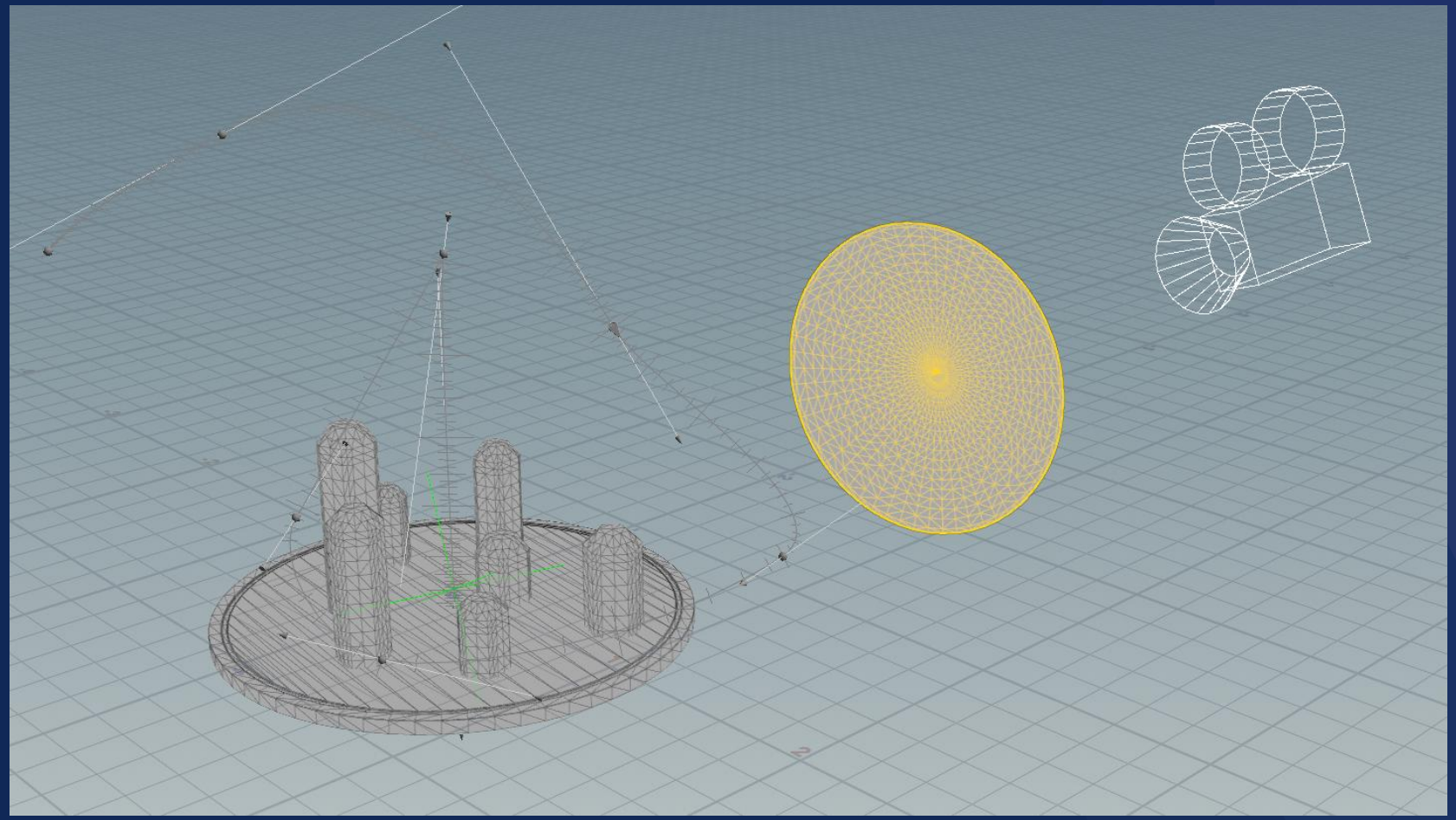
※F58 少し明るくしています



SDSパスはともかく、
他はライトトレーシングでもいい説



大丈夫！
カメラの前にレンズを置いておきました！



今回のコースティクス、**すべてがSDSパス**


32spp
説立証ならず

使用ライブラリ・3Dモデル

glm(cpu only), helper_math.h(https://github.com/NVIDIA/cuda-samples), Orochi, alembic, stb-image, libattopng, onesweep radix sort(https://github.com/wolfgangfengel/GPU-Zen-3, 自作といえば自作)

3D モデル: 自作

木目テクスチャ(New): https://polyhaven.com/a/laminate_floor_02