Silicon Studio

シリコンスタジオ最新テクノロジーデモ

レンダリング技術解説

安田 廉
re-yasuda@siliconstudio.co.jp

田村 尚希
nk-tamura@siliconstudio.co.jp

川瀬 正樹
masa@siliconstudio.co.jp

本セッションのターゲット

- **Q. 物理ベースという単語をしっていますか**
  - LV. 1 聞いたことはある
  - LV. 2 物理ベースを謳ったツール・エンジンを使ったことがある
  - LV. 3 実装しようとしている or GDC/Siggraph資料を少し見たことがある
  - LV. 4 実装した or 最新動向は全部追っている
- **LV.2~LV.3あたりをメインに想定**
  - デモで用いた技術を網羅的に取り上げている関係で多少知識があることを前提としています
  - 疑問点は是非ご質問下さい（口頭でもメールでも）

まずはデモ動画をご覧ください

www.youtube.com/watch?v=sYX9I3ONHc4

パフォーマンス

- ハイエンドPC: 60fps弱
  - GPU: NVIDIA GeForce GTX 780 Ti
- モバイルPC: 20fps
  - GPU: NVIDIA GeForce GTX 860M
- 解像度: フルHD

補足

- デモ名称: Museum
- GDCで公開したものを
  ブラッシュアップ
- ブラッシュアップ内容
  - 背景モデルを大きく変更
  - パフォーマンスを最適化

アジェンダ

- 第一章: デモの概要説明
- 第二章: 技術説明
- 第三章: まとめ

アジェンダ

- 第一章: デモの概要説明
- 第二章: 技術説明
- 第三章: まとめ

技術デモ開発の背景

近年の技術の進化・ハードウェアの性能向上により
一段上のグラフィクスを実現できると感じた

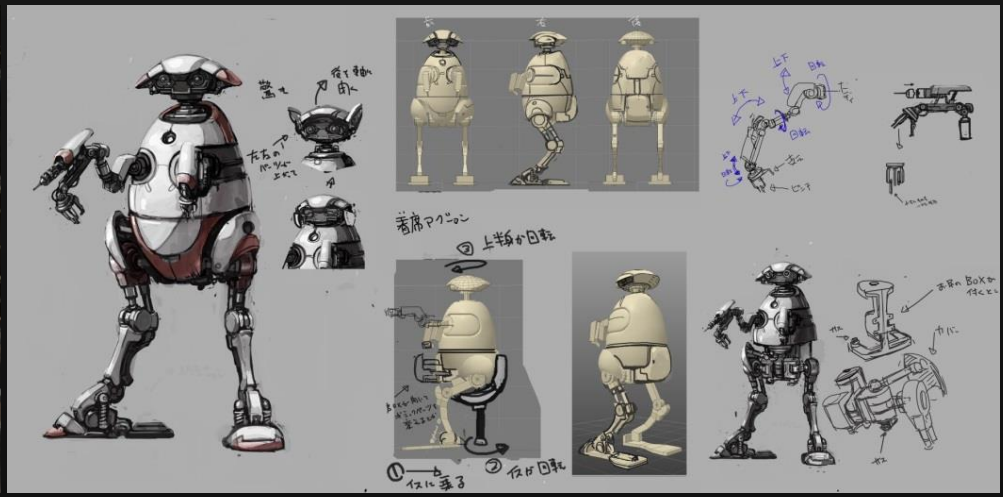
それを実証してみた

デモの方向性

- “まるで実写” & 映像作品のような絵作り

“The Third & The Seventh” from Alex Roman http://vimeo.com/7809605

コンセプトアート



注) GDC2014と現在では背景モデルが大きく異なります

技術デモ

- どういう所が大変だったのか?
  - 多くの技術要素をカバーする必要があった
  - 技術もアートも高いレベルが必要
  - 両者を高い水準で融合する必要がある

⇒一定の成果はあった

⇒技術面で重要なポイントを、この後説明したいと思います

アジェンダ

- 第一章: デモの概要説明
- 第二章: 技術説明
  - Lighting/Shading
  - ポストエフェクト
- 第三章: まとめ

Lighting, Shading

まずはこの絵ができるまで順に表示

LightMaps

+AlbedoMaps (+Directional Light)

+Image-Based Lighting

+小物/ロボット

+Local Reflection

+Fog

要求は「まるで実写」

- 実写とは言えないが一定の水準は満たせた
- 実現のために特に重視した要素
  - ライティング
  - 材質感
  - エイリアスの除去

要求は「まるで実写」

- ライティング
  - 現実に即した照明・環境マップ
  - Global Illumination

要求は「まるで実写」

- 材質感
  - つるつる・ざらざらの表現
  - 金属の表現
  - 視点やライティング
    に依存しない一貫性
  - 接写に耐えうる解像感

要求は「まるで実写」

- エイリアスの除去
  - エイリアス量と実写感は反比例
  - “物理ベース”なシェーディングでは
    エイリアス量はむしろ増大
  - Anti-Aliasingがきわめて重要に
    - これは十分に想定できていなかった
    - 後で苦労する羽目に

AAなし

AAあり

解説概要

- 絵が出来上がる順に解説
  - レンダラ概要
  - GIベイク
  - マテリアルシステム
  - Image-Based Lighting
  - Local Reflection
  - Anti-Aliasing
  - まとめ

※ Post Processingの解説はさらにその後

レンダラ概要

PBR, Shading Model, G-Buffer, SSAO, Fog etc.

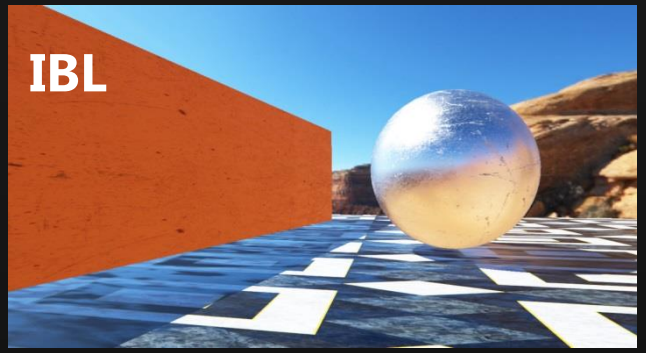
“物理ベース”

- 光の反射に関わる全てを“正しく”扱う
  - Shadingの計算式
  - マテリアルパラメータ
  - ライティング

- 材質感を追求する上で必然的に採用
  - Image-Based Lightingと合わせて
    最初からR&Dを行った

間接光

- Image-Based Lighting
  - 材質感の再現において必須
- Light Map
  - IBL用Cubemapの撮影のために必要
  - 背景オブジェクトは実行時もDiffuseに使用
- Local Reflection
  - a.k.a. Screen-Space Reflection
  - IBLで捉えられない細かいスペキュラを扱う



Shading Model

- BRDF
  - Cook-Torrance (NDF – GGX, G Term – Smith, Fresnel –Shlick)
  - Lambertian

- マテリアルパラメータ
  - Unreal Engine 4のパラメータセットを使用[Karis13]
    - ただしRoughness の代わりにShininess (=1-Roughness)を使用
    - BaseColor / Normal / Shininess / Metallic / Cavity

細かなこと1

- Deferred Renderer
  - 40枚弱のRender Target
- いまのところデモ特化仕様
  - 半透明なし
  - 平行光のみ
    - Spot Light, Point Lightは未対応

細かなこと2

- Fog
  - Shadow map の ray-marching
  - Cubemapを 1 つだけ考慮

- SSAO
  - Alchemy AO
  - 少しだけGI的な乗算方法
    - $Color \ *= lerp(BaseColor, 1, AOTerm);$

G-Buffer

- 法線エンコーディングにはOctahedron normal vector encoding [Meyer10]
- LightMapColorには代わりにEmissiveを格納する場合もある
- Local ReflectionのためにVertexNormalを格納していた時期もあったが今はない
- Clothnessは布用パラメータだがまだ研究中

GIベイク

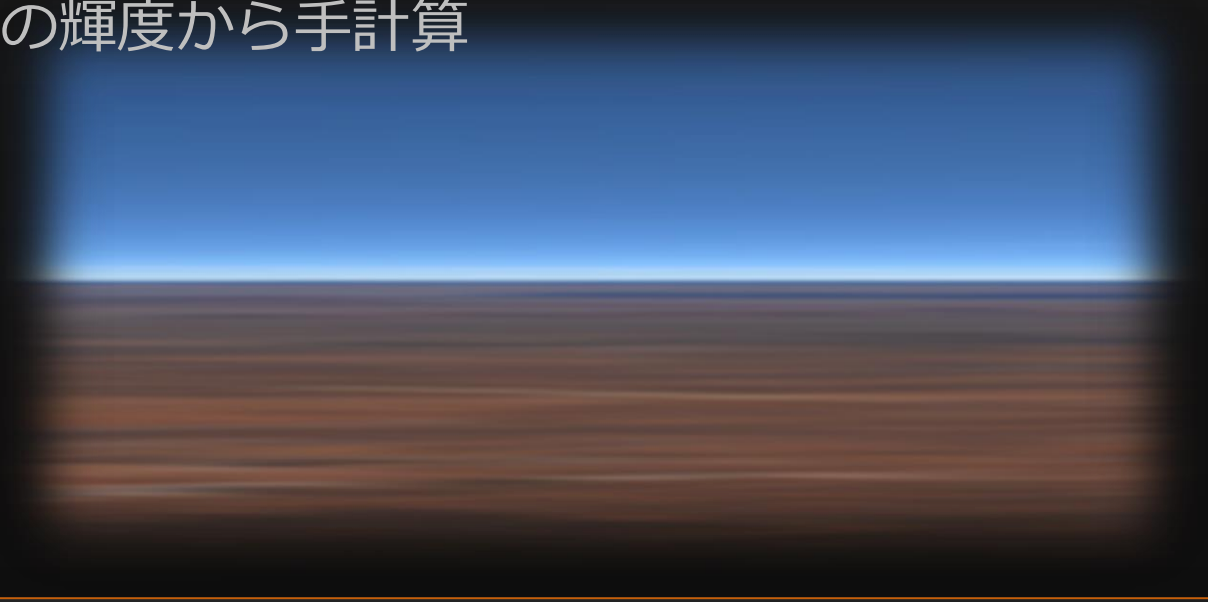
Light Maps, Sun/Sky

Light Map Baking

- 3ds Max + V-ray
- 背景オブジェクトのみ
  - マテリアル設定はAlbedoだけ
- 方向非依存
  - HL2 basisなどではない
- ~50MB (BC6H)

Sun/Sky

- デザイナの選んだHDRI + 手動で古典的平行光を設定
  - ランタイムへの持ち込みやすさ優先
- 平行光の強度は空の輝度から手計算



注意点

- Punctual Light
  - 古典的光源は内部で強度がπ倍されている
  - 平行光には手計算した強度をπで除算して指定
  - ランタイム側もPunctual Light

- V-Rayでベイクする場合はV-Rayマテリアルで
  - 標準マテリアルでも正しくベイクできたように見えるが
    - 輝度が2倍になっていたりする

マテリアルシステム

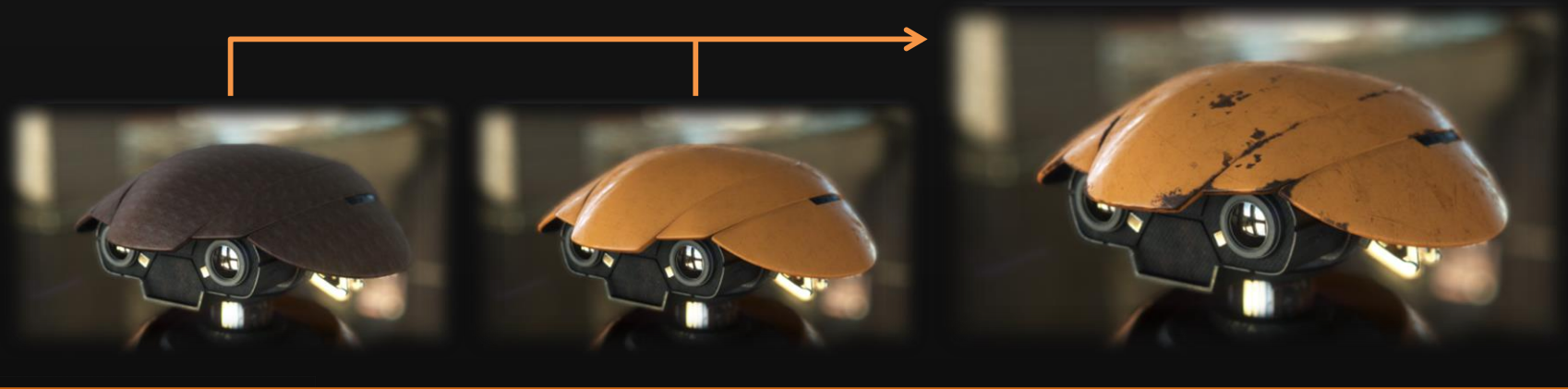
マテリアルレイヤリング，リニアワークフロー

要求

- 複雑なマテリアルの表現
  - 光沢感の違いや金属/非金属の混在など
- 接写に耐えうる解像感

Material Layeringシステム

- 5つのパラメータ(＆テクスチャ)の集まりを１レイヤーとする
  - Albedo, Normal, Shininess, Metallic, Cavity
  - 複数のレイヤーを重ねてマスクテクスチャで合成
  - 全テクスチャに独立してUV変換が設定可能



Material Layeringシステム

- 先達の一例
  - The Order: 1886のようなスタック的なマテリアルシステム[Neubelt13]
    - 手軽で理解しやすい
  - Unreal Engine 4のようなリアルタイムブレンディング[Karis13]
    - 異なるUVスケールでテクスチャ合成することによる解像感

- 本デモにおける実装
  - スタックベースのレイヤリング＋リアルタイム合成
    - 実装コストも現実的だった

Material Layeringシステム

- レイヤー合成は毎フレームレンダリング時に行う
  - オフラインで合成する術もあるが、解像感が損なわれる
    - 合成した結果をテクスチャに出力した時点で情報が失われる
    - 超接写する場合、4Kでも解像感が十分とは言えない
  - レンダリング時にUVスケールを適用しつつ合成することで解像感が出る

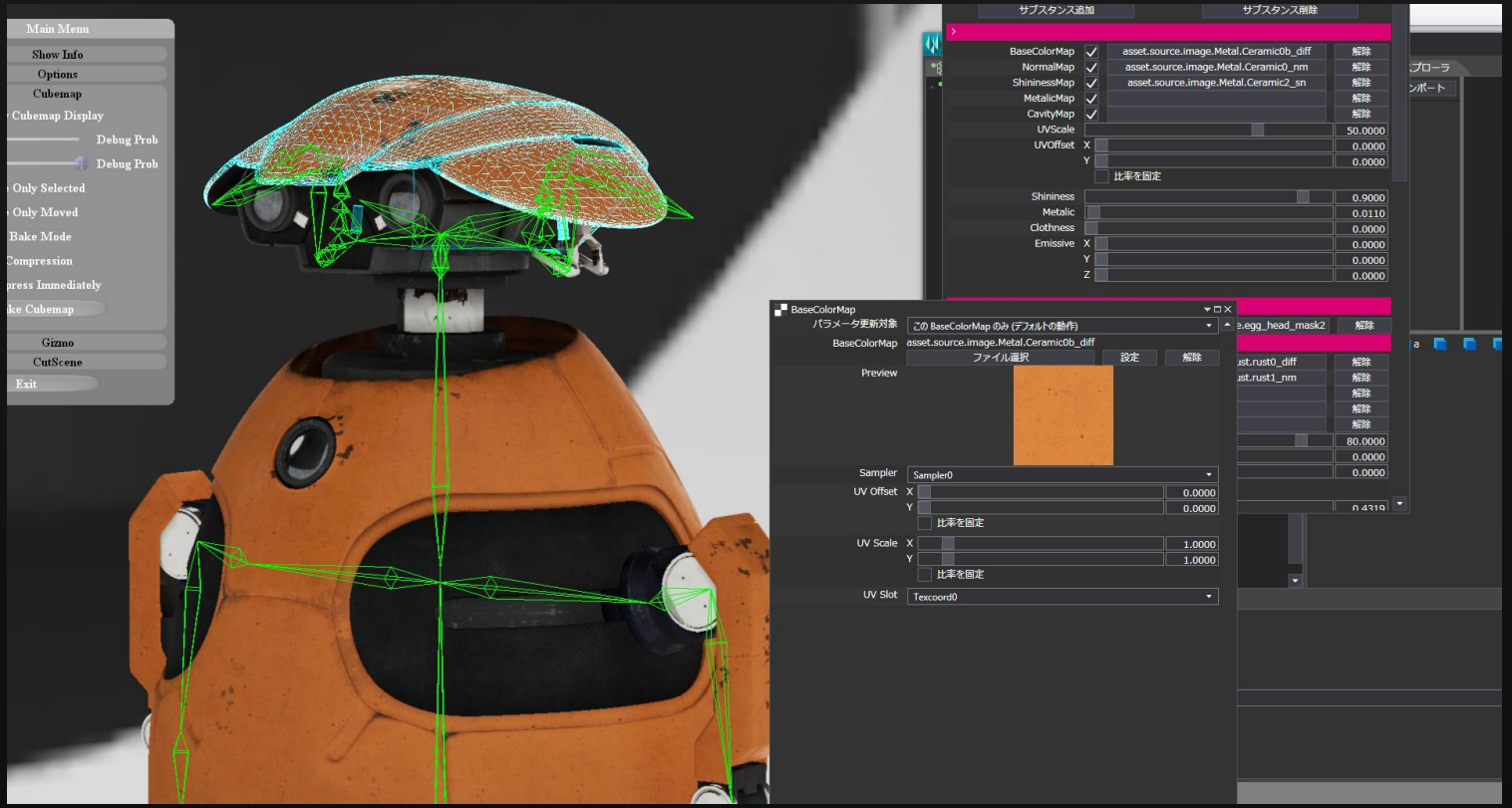
- マテリアルの合成は全てG-Bufferへの書き出し時に実行
  - それぞれのパラメータは独立して合成し、G-Bufferに格納

マテリアル編集

- 専用ツールで全てのマテリアルを設定
  - Material layering＆リアルタイムプレビュー
  - 環境Cubemapを切り替えつつ設定



マテリアル編集


Main Menu
Show Info
Options
Cubemap
Cubemap Display
Debug Prob
Debug Prob
Only Selected
Only Moved
Bake Mode
Compression
Gizmo
CutScene
Exit
サブスタンス追加
サブスタンス削除
BaseColorMap
asset.source.image.Metal.Ceramic0b_diff
NormalMap
asset.source.image.Metal.Ceramic0_nm
ShininessMap
asset.source.image.Metal.Ceramic2_sn
MetalicMap
CavityMap
UVScale
50.0000
UVOffset
比率を固定
Shininess
0.9000
Metalic
0.0110
Clothness
0.0000
Emissive
BaseColorMap
パラメータ更新対象
この BaseColorMap のみ (デフォルトの動作)
asset.source.image.Metal.Ceramic0b_diff
ファイル選択
設定
解除
Preview
Sampler
Sampler0
UV Offset
UV Scale
1.0000
UV Slot
Texcoord0

リニアワークフロー

- 当然全部リニア
- Raw画像からリニア現像するプラグインも作成 [McAuley12]
  - が、ほとんど使っていない(レトロカメラだけに使用)

※これは写真です

リニアワークフロー

- プラグインを用意しても素材撮影は手間が多すぎた
  - 常設の＆十分な広さの撮影スペースが必要
  - rawが重い、現像が遅い
  - 小規模のチームだとかなり厳しい
    - 全て自動化しないかぎりコストが高すぎる

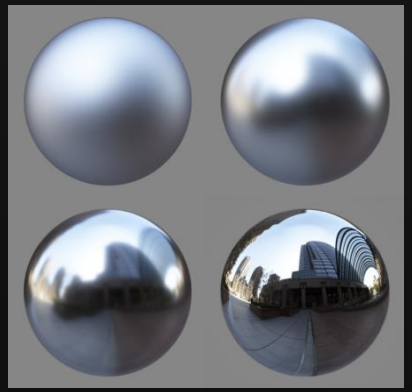
- ほぼすべてのテクスチャを今まで通りの素材集などから使用
  - それでもリニアワークフロー＆PBRだと満足できるクオリティが得られた
  - 全てちゃんとした素材を使ったらどうなるかは試してみたい

Image-Based Lighting

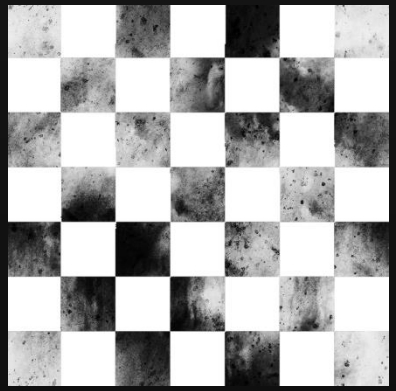
Localized cubemaps, cubemap blending, parallax correction

Image-Based Lighting

- Cubemapを用いた間接ライティング[Karis13]
  - 材質の表現力が随一
- 事前計算はコストが高め



Cubemap



Shininess Map

結果

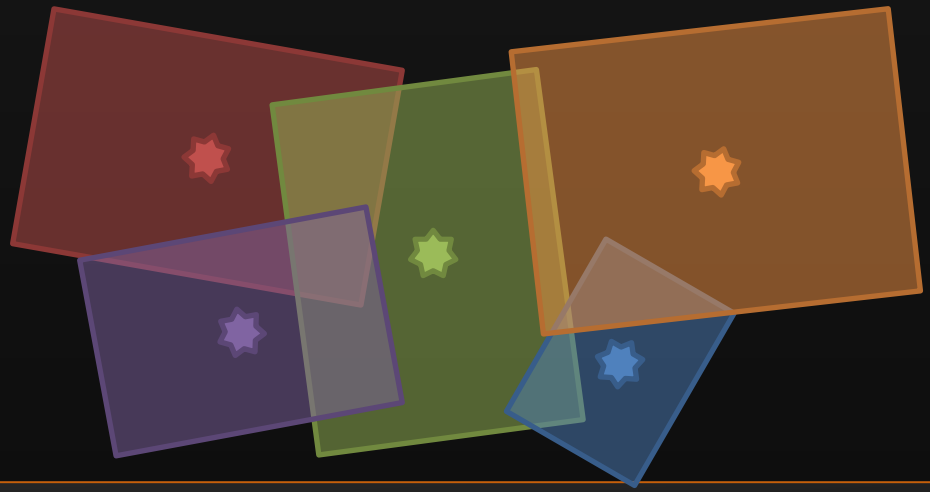
Image-Based Lighting

- 最も支配的な間接光の要素
  - 背景以外 … DiffuseもSpecularもIBL
  - 背景オブジェクト … DiffuseにLight Map, SpecularにIBL
- 動的なオブジェクトや細かい反射はLocal Reflectionで補う

- Parallax Correctionはなし
  - 実装はしたが完全な鏡面以外で起こるエラーが許容できず不採用
  - 改良案はあるが今回は間に合わず

Localized Cubemaps

- シーン全体でCubemap一つだけというわけにはいかない
  - ライトのようにシーンに配置できるようにする
  - 影響範囲(BoxないしSphere)も設定

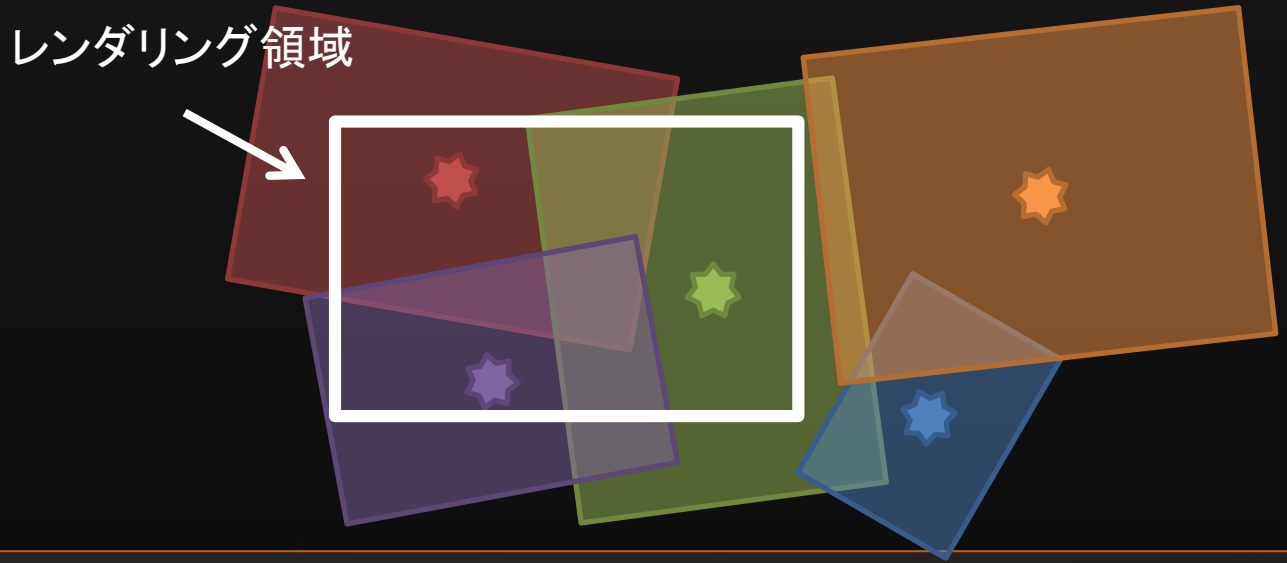


デモにおけるLocalized Cubemaps

- 全体で90個ほど配置
  - 各面 256 × 256 pixels
  - ~50MB (BC6H)

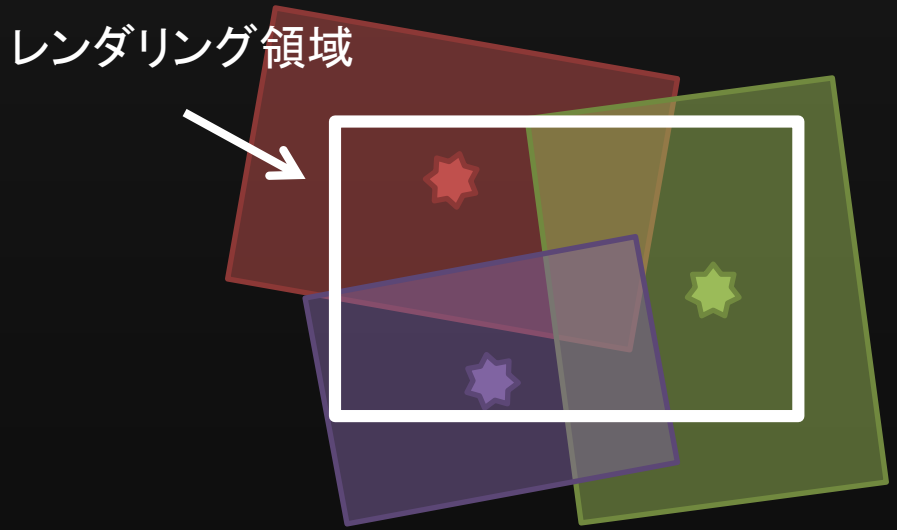
Localized Cubemapを用いたShading

- Cubemap(の影響範囲)を順々に描画
  - 影響範囲内のピクセルのみShadingを行う



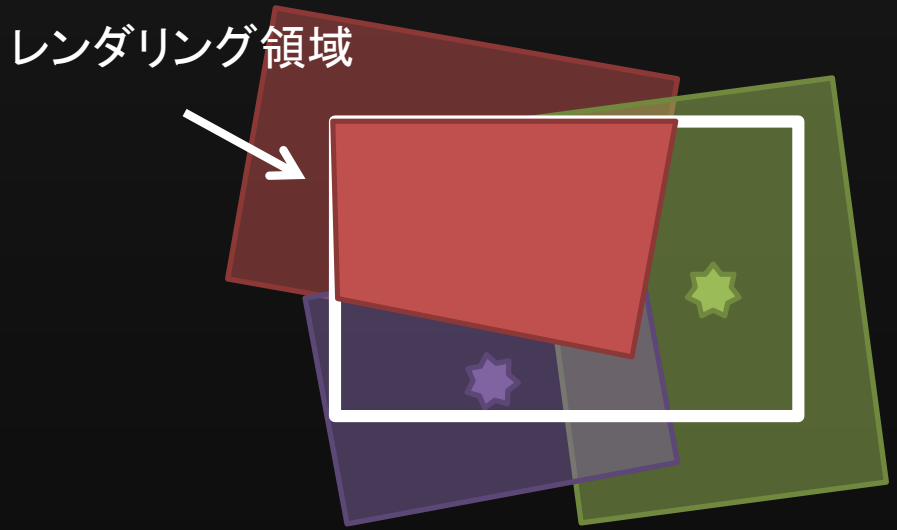
Localized Cubemapを用いたShading

- Cubemap(の影響範囲)を順々に描画
  - 影響範囲内のピクセルのみShadingを行う



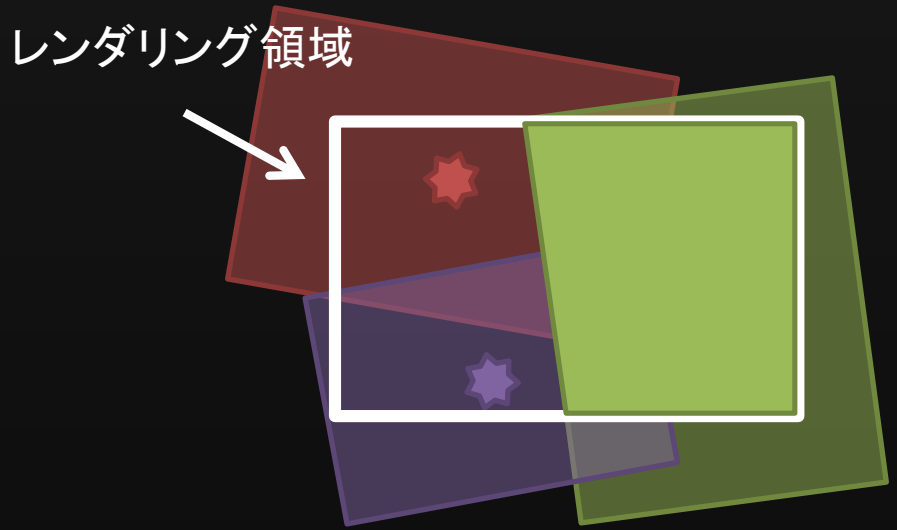
Localized Cubemapを用いたShading

- Cubemap(の影響範囲)を順々に描画
  - 影響範囲内のピクセルのみShadingを行う



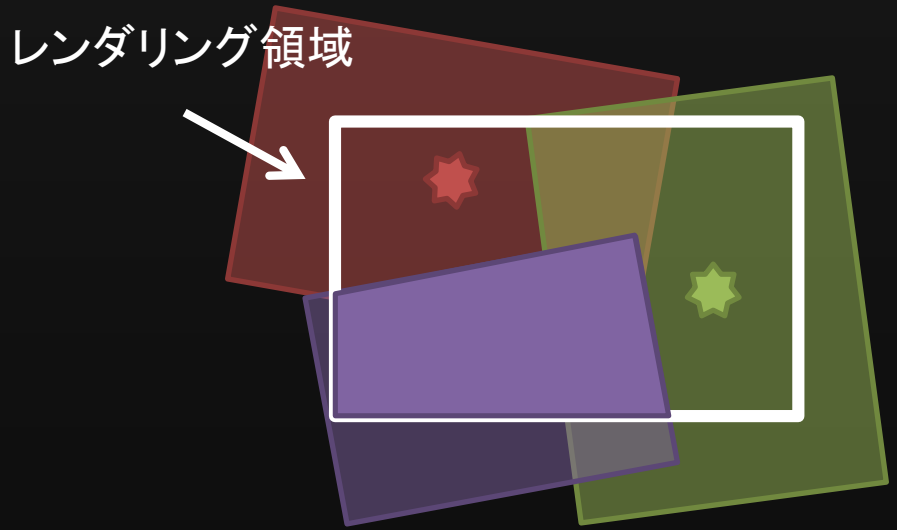
Localized Cubemapを用いたShading

- Cubemap(の影響範囲)を順々に描画
  - 影響範囲内のピクセルのみShadingを行う



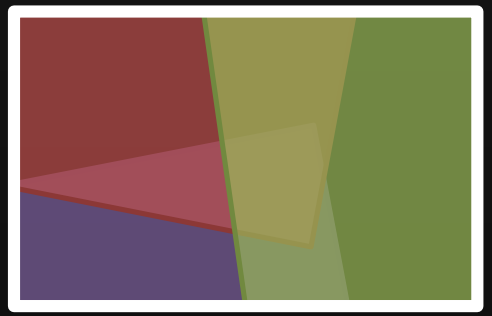
Localized Cubemapを用いたShading

- Cubemap(の影響範囲)を順々に描画
  - 影響範囲内のピクセルのみShadingを行う



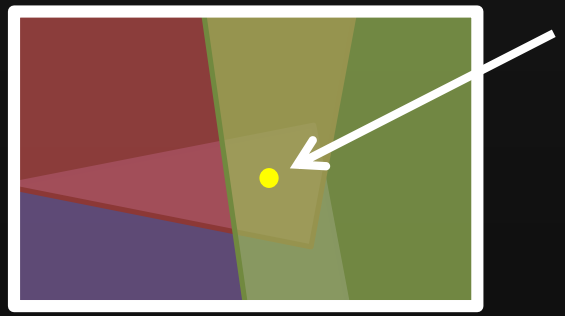
Localized Cubemapを用いたShading

- Cubemap(の影響範囲)を順々に描画
  - 影響範囲内のピクセルのみShadingを行う



Cubemapの結果のブレンディング

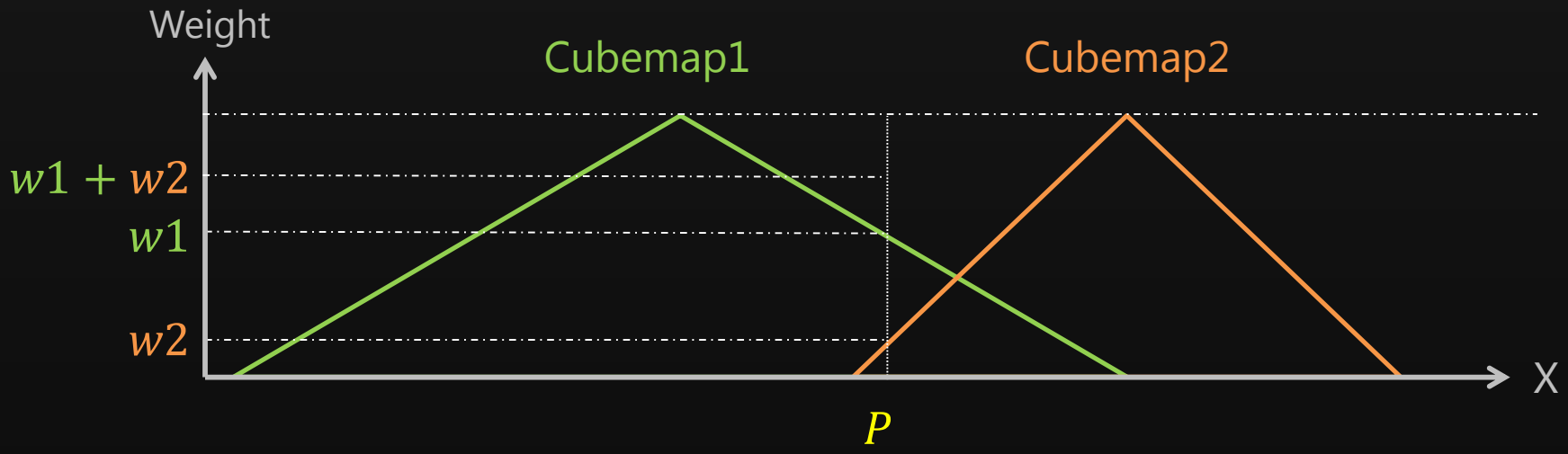
- Cubemapが重なった個所は結果をブレンドする必要がある



このpixelの色は？

Cubemapの結果のブレンディング

$$(Ir)radiance_{P,\omega_i} = \frac{Cubemap1_{\omega_i} * w1 + Cubemap2_{\omega_i} * w2}{w1 + w2}$$



Cubemapの結果のブレンディング

- Deferred Shadingの際に分母と分子を独立して加算ブレンド
  - 最後にアルファ値で除算して正規化

$$\frac{Cubemap1_{\omega_i} * w1 + Cubemap2_{\omega_i} * w2}{w1 + w2} \dashrightarrow \begin{cases} RGB\ channel \\ Alpha\ channel \end{cases}$$

- 簡単＆基本的には充分な実装
  - ただ精度の問題が起こりうるので多少工夫できる

Cubemapブレンドの精度問題

- 16bit floatでは精度が足りないケースがある
  - Cubemapの輝度と重みが両方小さいor大きい場合
  - 分子が16bit floatの表現幅を超える
  - デモでは輝度が$1e^{-4}$になる場所もあった

$$\frac{Color_{cubemap1} * w1}{w1} \Rightarrow \frac{1e^{-4} * 0.001\ (16bit)}{0.001(16bit)} \Rightarrow 5.96e^{-5}(16bit)$$

Cubemapブレンドの精度改良

- 重みの総和を求めるだけのパスを先に走らせておく
  - R16_UNORMバッファへ出力
  - 2パス目ではalphaには出力せず、シェーダ内で総和で除算してrgbに出力

$$\frac{Color_{cubemap1} * w1}{w1} \Rightarrow \frac{1e^{-4} * 0.001(32bit)}{0.001(16bit)} = 9.95e^{-5}(16bit)$$

シェーディング回数

- Cubemapが重なった個所はPixel Shaderがその回数走る

Cubemap

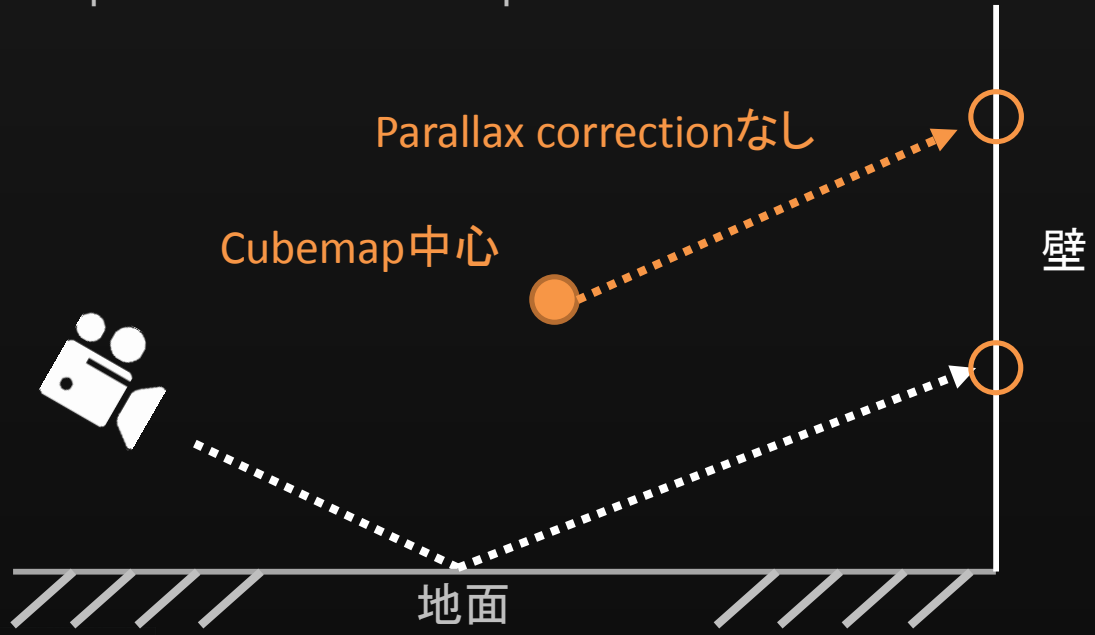
青　1回
黄　2回
橙　3回
赤　4回以上

シェーディング回数

- 重ならないよう配置すればパフォーマンスへの影響は少ない
  - Pixelが起動しないような最適化は必要
- 前の画像だと画面内に45個Cubemapが映っている
  - Pixel Shaderの起動回数は画面の解像度×1.5程度？

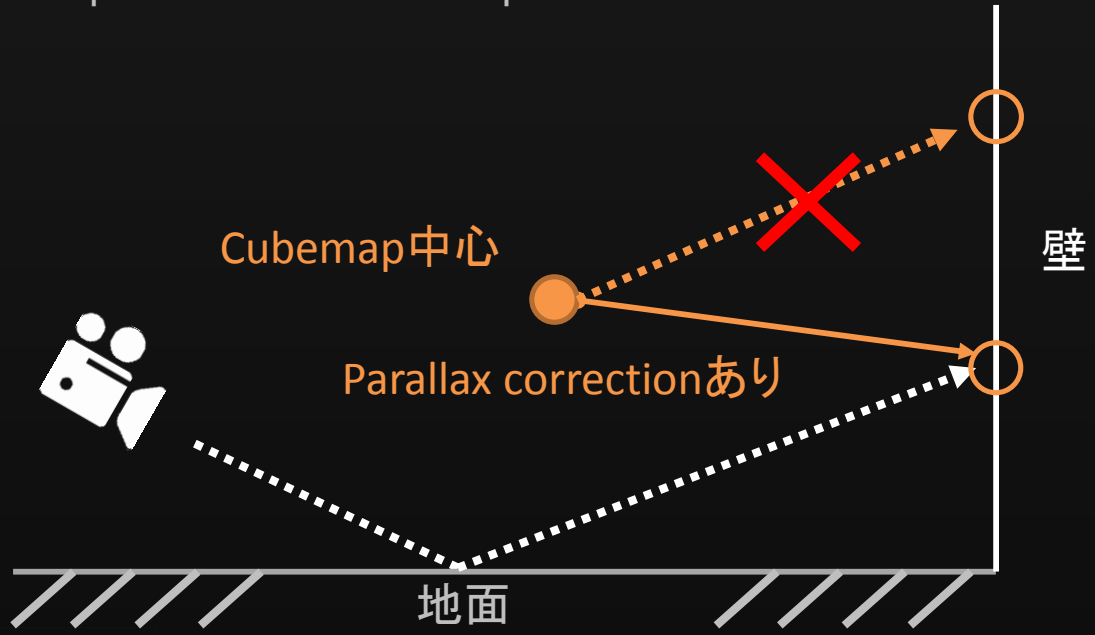
Parallax Correction

- Cubemapのサンプル方向をある程度修正[Lagarde2012]
  - Cubemapのフェッチ時に全pixelシェーダで処理



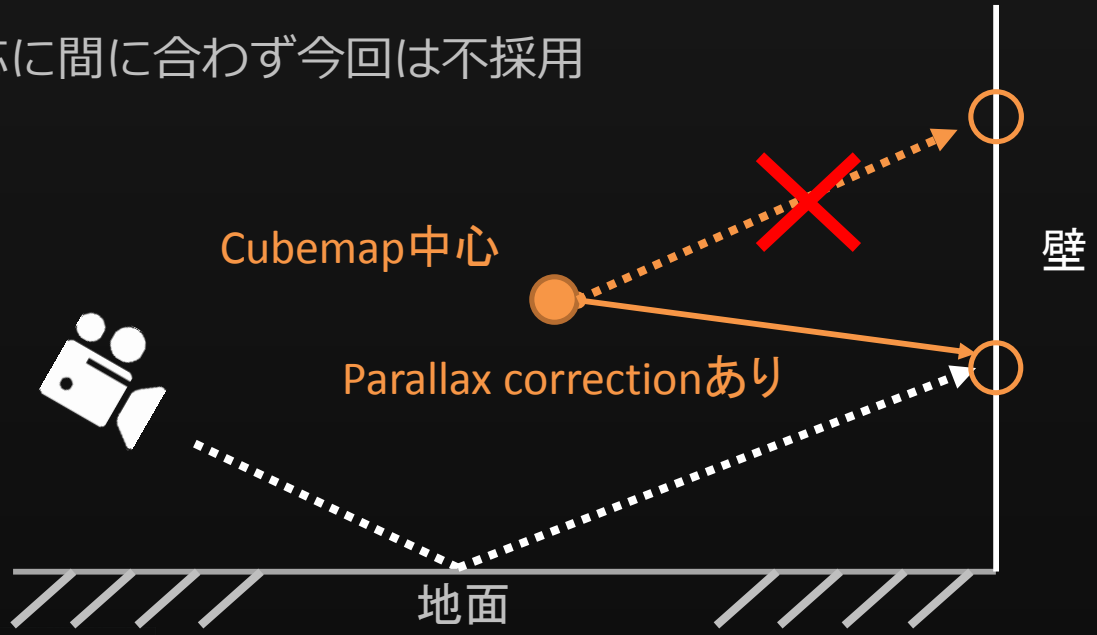
Parallax Correction

- Cubemapのサンプル方向をある程度修正[Lagarde2012]
  - Cubemapのフェッチ時に全pixelシェーダで処理



Parallax Correction

- あまり良い結果を得られなかった
  - 完全鏡面だとうまくいくが、Shininessが低いとエラーが大きくなる
  - 対応に間に合わず今回は不採用



Local Reflection (Screen-Space Reflection)

BRDFを考慮したLocal Reflection, Temporal Super Samling

IBLのみ

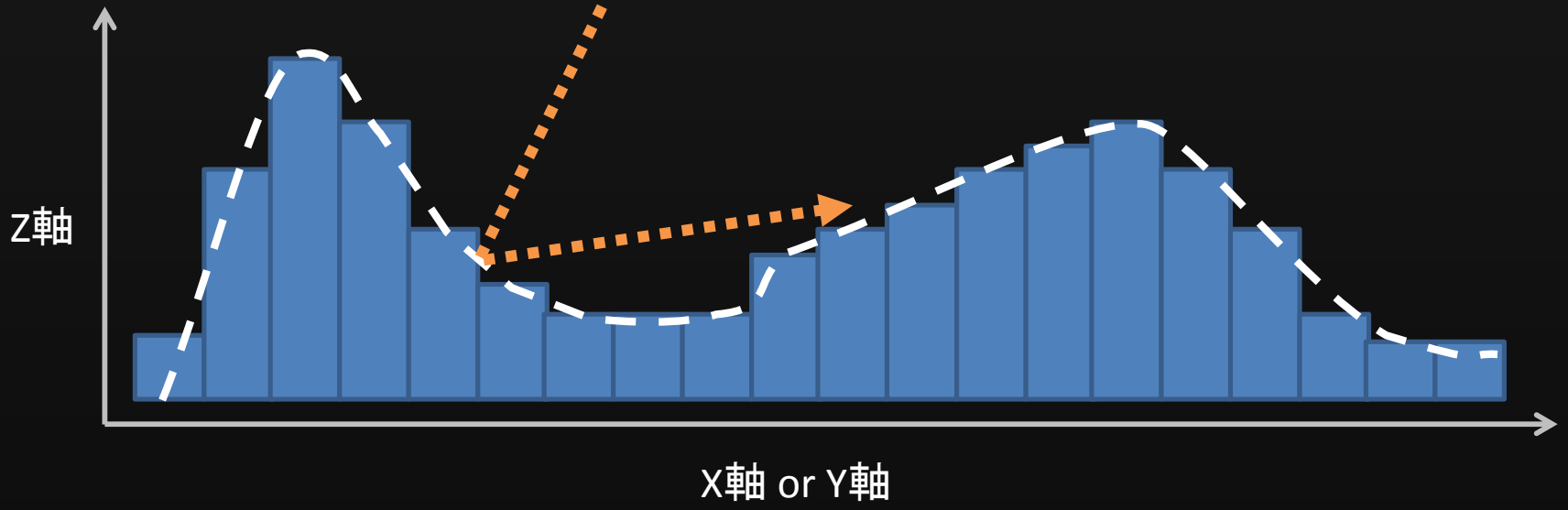
Local Reflectionあり

IBLのみ

Local Reflectionあり

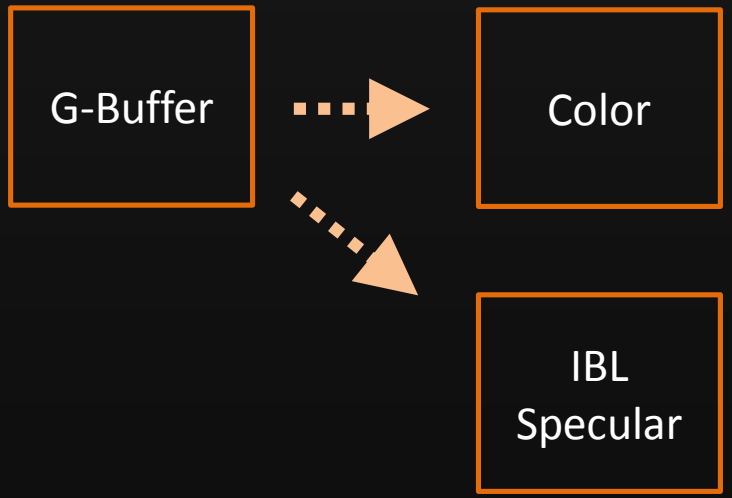
Local Reflectionとは

- Depth BufferとNormalを用いたScreen Spaceのリフレクション
  - 基本的にはDepth Bufferに対するRay Marching



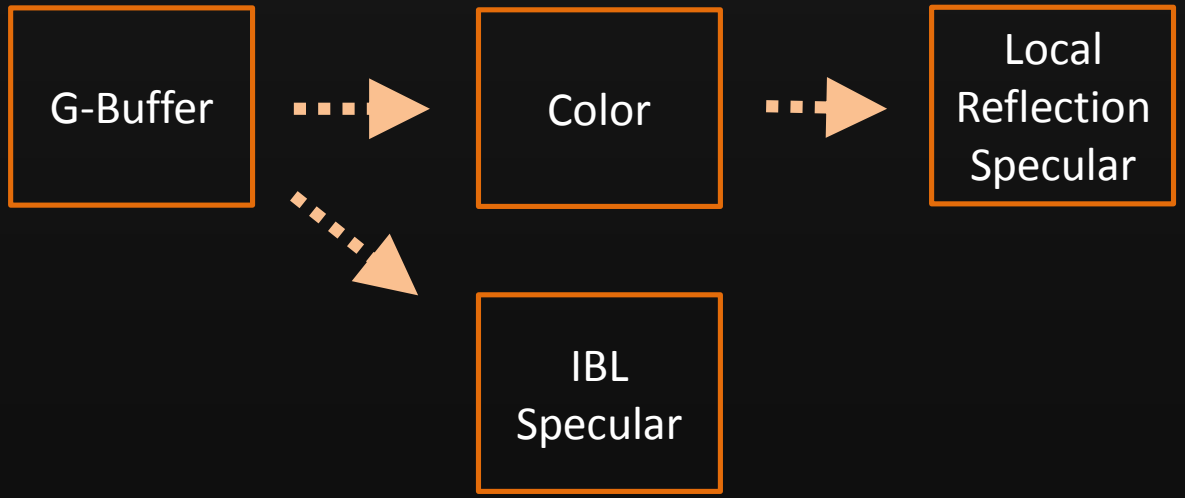
基本手順

- 直接光 & IBL (&LightMaps)でDeferred Shading
  - その際IBLによるSpecularは別途書き出し



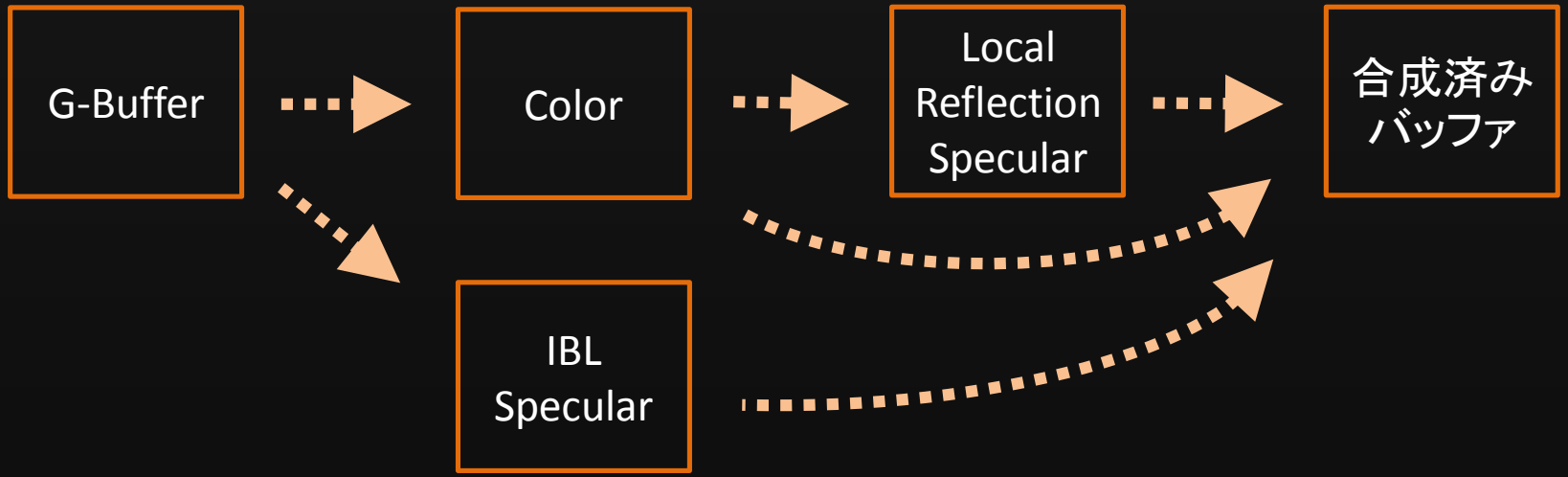
基本手順

- Colorバッファ (およびDepth Buffer)を使ってLocal Reflection



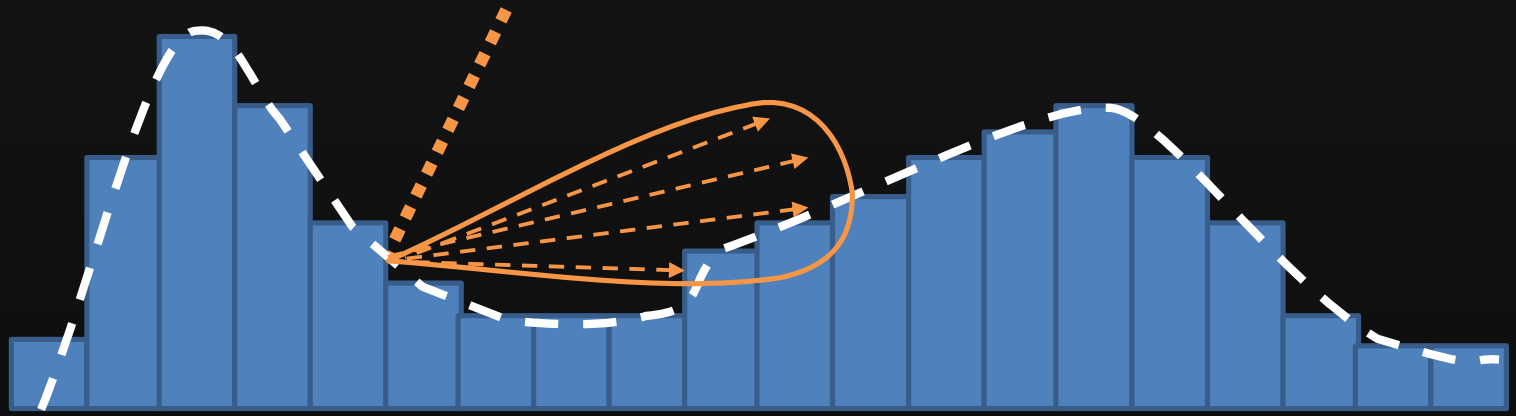
基本手順

- 合成
  - IBL SpecularとLocal Reflection Specularをブレンド
  - ブレンド結果をColorバッファに合成



Local Reflection & BRDF

- 完全鏡面反射だけでなく、BRDFを考慮したい
  - ざらざらな表面に鏡のような反射が起こるのは避けたい
  - いくつかのアプローチが存在[Valient14][Gotanda13][Uludag14]



デモにおけるLocal Reflection

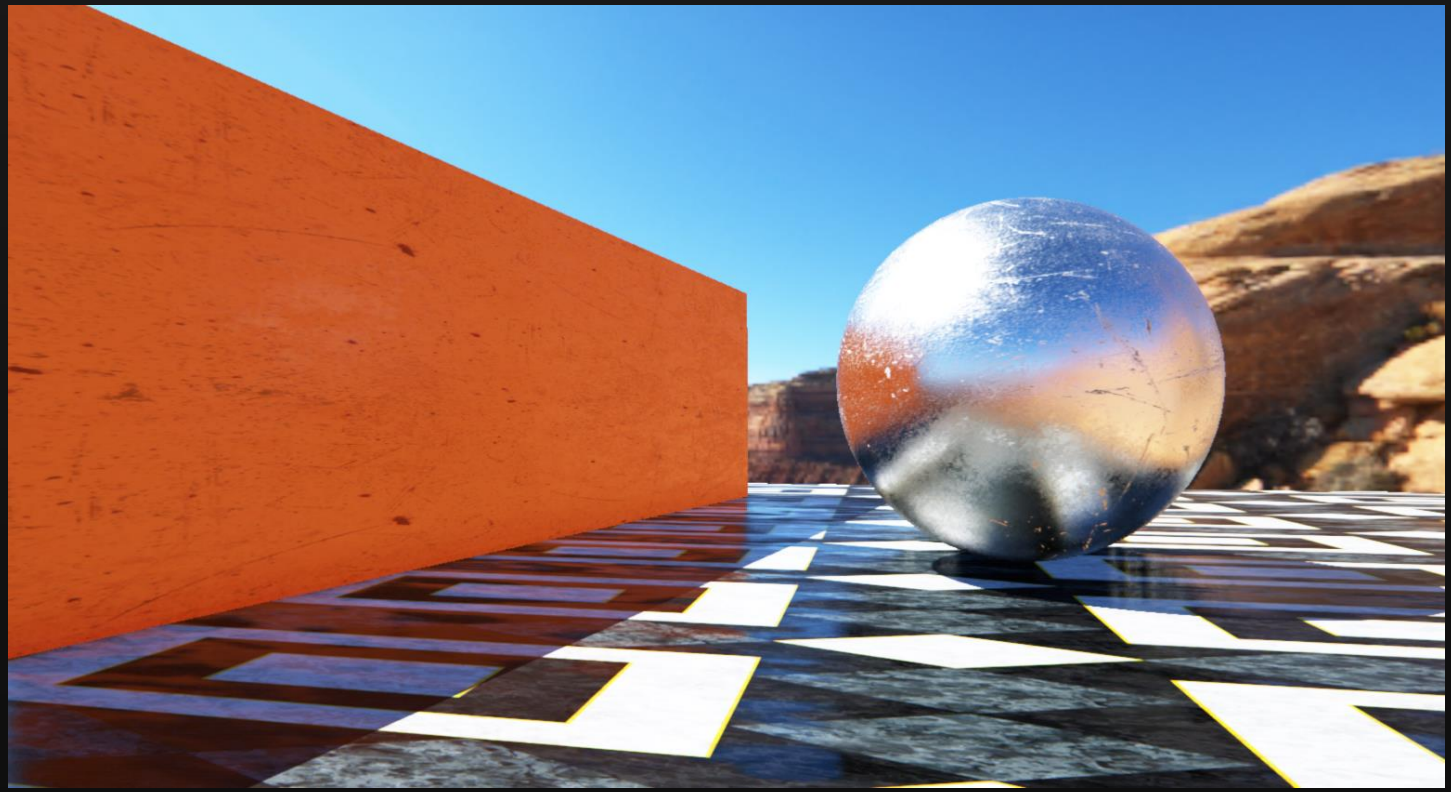
- BRDF Importance Sampling
  + Hi-Z Cone tracing
  + Mip-mapped Color Buffer
  + Temporal Super Sampling

- 他のアプローチも試したかったが、時間がなかった
  - Hi-Zのトラバースなどはまだかなりアドホック
    - ちゃんとしたCone-Tracingにはなっていない

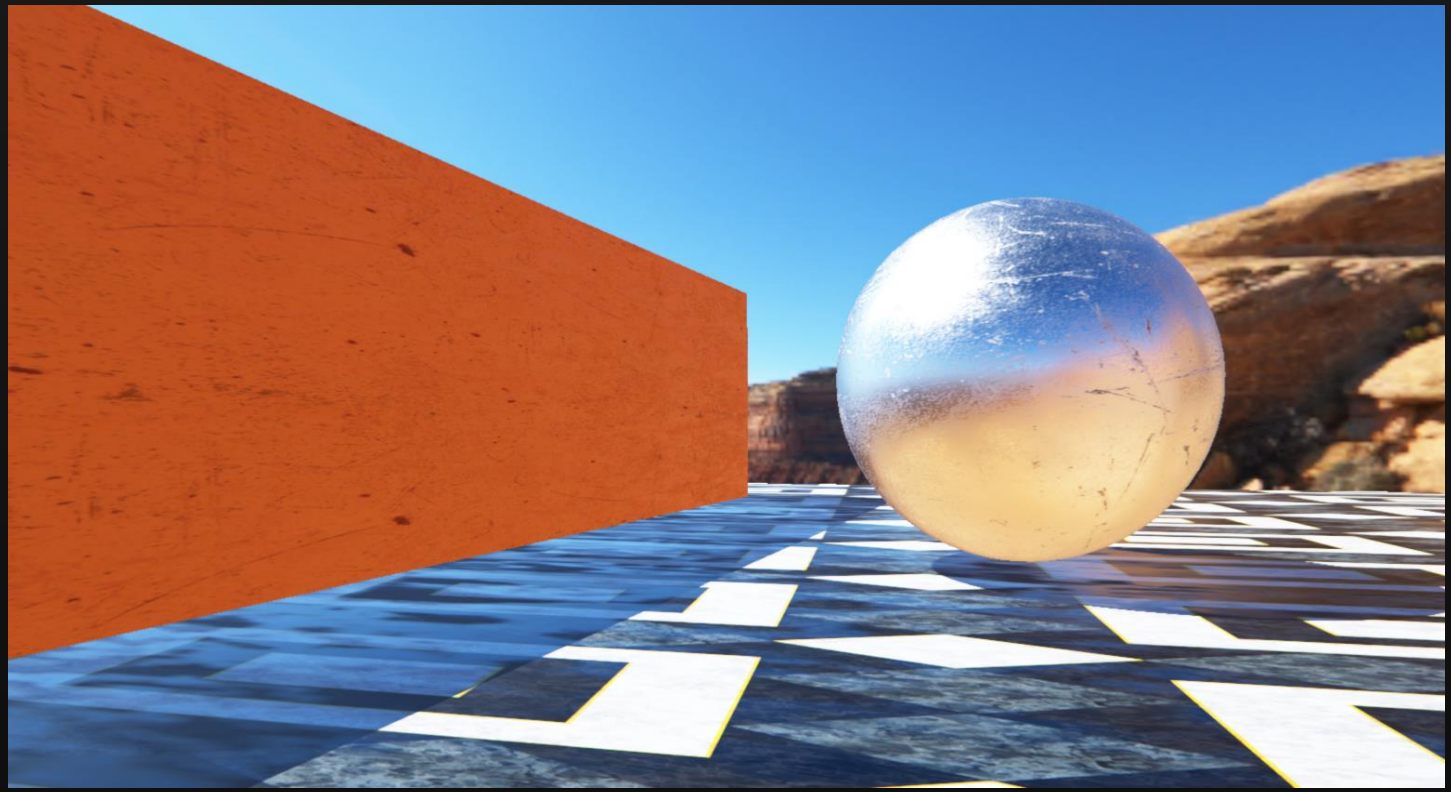
その他詳細

- Half解像度, FullHDの半分で3~4ms(Gefore GTX660)
- 1フレーム毎のrayの数 1~4本
  - Shininess依存。
  - 1ray当たり最大16tap
- ブラー処理はなし

Local Reflection+材質感



IBLのみ



Temporal Super Sampling

- 時間軸方向にBlur
  - 過去の数フレーム～十数フレーム分を再利用してサンプルをふやす
    - ちなみにFog/SSAOも同様の処理をしている
- 残像対策
  - Velocityの差による重みづけ[Sousa13]
  - 輝度・ブレンド比による重みづけ

Anti-Aliasing

Toksvig Normal Mapping, Temporal G-Buffer Filtering

理想

物理ベース導入した

IBL実装した

モデルもハイポリで作りこんだ

リアルなライティングもした

最高のデモが出来た！

現実

PBR + IBL + ハイポリ+高コントラストなシーン

エイリアシング地獄

エイリアシングがひどすぎる問題

- ここまでとは想定できていなかった
  - 酷過ぎてエイリアス以外に目がいかないほど
  - 特にスペキュラエイリアス

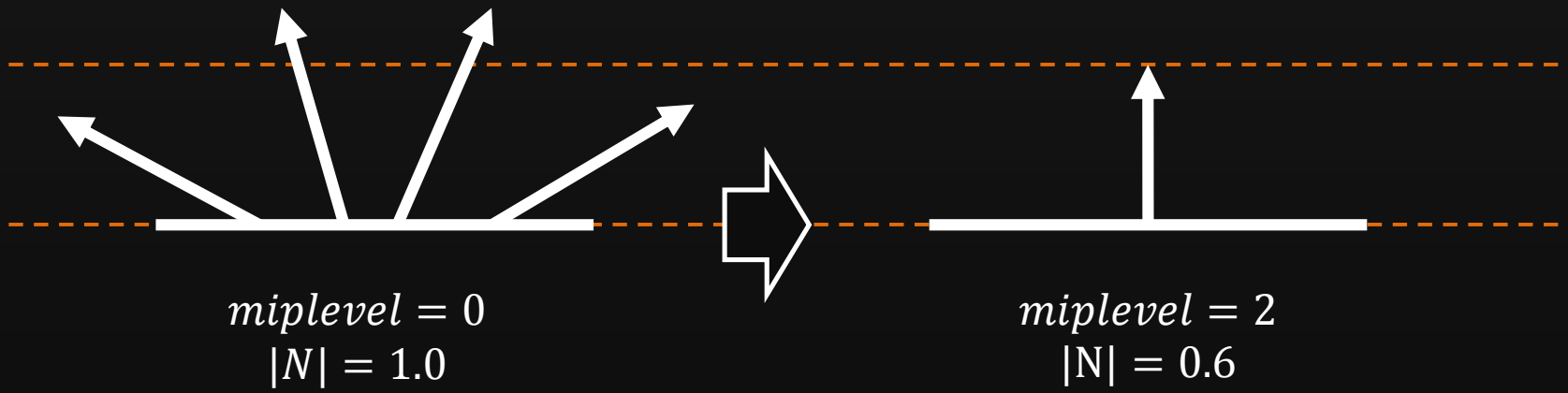
- これからPBRに取り組む場合はご注意を
  - Anti-Aliasingにかなり時間を割かざるをえない

対応

- Toksvig Normal Mapping
- Temporal G-Buffer Filtering
- FXAA
- Temporal Anti-Aliasing
  - TemporalAAに関してはPost-Processingの方で説明

Toksvig Normal Mapping [Toksvig04]

- Normal MapのMipmap構築の際、ベクトルを正規化しない
  - 単純な平均を取る ⇒ 下の階層の法線の分散が大きいほど短くなる
- 使用時にはベクトルの長さに応じてshininessを抑え込む



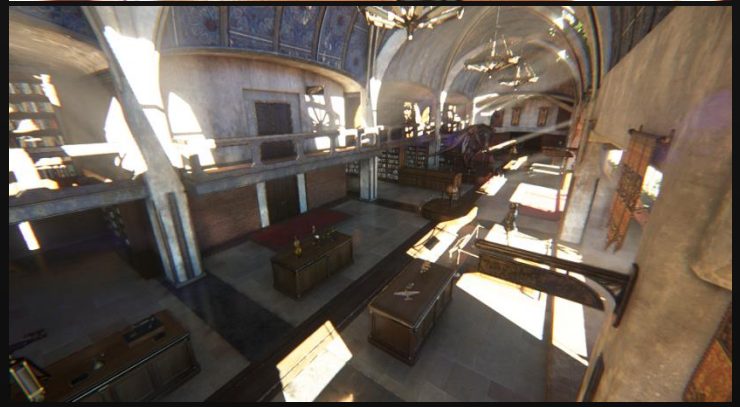
Temporal G-Buffer Filtering

- 法線を時間方向でブレンド + Toksvigの要領でshininess低減
  - ジオメトリエイリアシング対策
  - Toksvigの手法はジオメトリエイリアスには無力

- 後述のTemporal AAと併用でも効果がある
  - 強力なTemporal AAでは空間的・時間的な高周波成分をかき消してしまう
  - Temporal G-Buffer Filteringである程度スペキュラが平滑化される
    - TemporalAAでハイライトがかき消されないように

アジェンダ

- 第一章: デモの概要説明
- 第二章: 技術説明
  - Lighting/Shading
  - ポストエフェクト
- 第三章: まとめ



Post Processing

デモで使っているポストプロセス

- テンポラル・アンチエイリアス
- 被写界深度／絞り／ボケ味シミュレーション
- ポストプロセス型アンチエイリアス
- エアリーディスクシミュレーション（小絞りボケ）
- モーションブラー
- レンズディストーション
- 色収差
- 周辺減光
- レンズフレア
- フィルム／トーンマッピング
- カラーグレーディング
- フィルムグレイン／ノイズ
- Etc.

ポストプロセスに含まないもの

- ポストプロセスは一通り完成した画像に対する処理とする
- ここで述べる「ポストプロセス」に含まないもの
  - スクリーンスペース・ローカルリフレクション
  - スクリーンスペース・アンビエントオクルージョン
  - スクリーンスペース・フォグ
- これらはシェーディングの一環として処理
  - カラーバッファ合成以前に必要になるため
- ポストプロセス適用前のカラーバッファには既に含まれている
  - これらに対するテンポラル・スーパーサンプリングも適用済み

ポストプロセスなし

ポストプロセスあり

ここで取り上げる内容

- テンポラル・アンチエイリアス
  - メイン
- おまけ
  - ボケ味シミュレーション
  - カメラシミュレーション

テンポラル・アンチエイリアス

エイリアシングとは？

- エイリアシング
  - サンプリング周波数の1/2を超える周波数を表現できない
    - ナイキスト周波数と呼ばれる
- 空間周波数に対してサンプリング周波数が低いと発生
  - ピクセルのジャギー
  - モアレ現象

アンチエイリアシング

- 何らかの方法でサンプリング数を増やす
  - スーパーサンプリング
  - これが本来の意味でのアンチエイリアシング

従来のアンチエイリアシング

- ハードウェアマルチサンプリング
  - MSAA
  - カバレッジのみのスーパーサンプリング
    - シェーディングは1サンプルのみ
    - ポリゴンエッジにしか効果がない
  - シェーディングによって生じるエイリアスには効果なし
    - スペキュラーエイリアスには無力
- ポストプロセス型アンチエイリアス
  - FXAAなど
  - サンプリングを増やす訳ではない
  - 時間方向へのエイリアシングには効果なし
    - 静止画としてのジャギーにしか効果がない
    - 一枚絵では綺麗でも動かすとウネウネする
- これらは画像としてのジャギー除去に最適化した技術
  - 本当にサンプリングが増えている訳ではない
  - 本質的な解決方法ではない

物理ベース時代のエイリアス

- 極めて鋭いスペキュラーが発生し得る
  - 空間周波数が際限なく高くなる
  - 輝度が際限なく高くなる
    - HDR にも程がある
- 頂点数の増大との相乗効果
  - エッジの増加
    - 1ピクセル内にエッジが10個くらい重なることも
  - ピクセル毎の法線がほとんど乱数のように異なる
- 極めて高輝度のノイズに近いスペキュラーエイリアスが生じる

こんなことになりました

スーパーサンプリングの重要性

- 高輝度高周波スペキュラーに対するアンチエイリアス
  - MSAA や FXAA はほぼ効果なし
    - ディファードなので MSAA とは相性が悪い
  - Toksvig で多少は改善可能
    - エッジ部分はまだまだ厳しい

- サンプル数を物理的に増やさないとどうにもならない

ブルートフォース・スーパーサンプリング

- 真面目にサンプル数を増やす
- GDCデモではフルシーン・アンチエイリアス（FSAA）を適用
  - 内部では3k～4kレンダリングをおこなっていた！
  - 効果は限定的で、見苦しいエイリアスがかなり残っていた
    - そもそも数サンプル程度ではまったく足りない
- 非常にコストパフォーマンスが悪い
  - サンプル数に比例して負荷が高くなる

テンポラル・スーパーサンプリング

- 描画量を増やさずにサンプル数を増やす工夫

- 時間方向への積分でサンプルを増やす
  - 複数フレームの画像を合成
    - 現在のピクセルの過去フレームにおける位置のカラーをブレンド
  - ピクセルは通常サブピクセル単位で移動
    - 自動的にフレーム毎のズレが積分される
      - 確率的にサンプリング数が増える
    - ピクセルの位置がフレーム間で全く動いていない場合は一切効果なし

テンポラル・アンチエイリアスの実装

- Unreal Engine 4 をベースに拡張
  - [Karis14] SIGGRAPH 2014
  - テンポラルエイリアスへの効果は非常に高い
- 効果の大きさ故のデメリットも
  - 高周波の高輝度成分はほぼスポイルされる
    - スペキュラーエイリアスはスペキュラー自体が無くなる
    - 本来の物理的な輝度を維持できない
    - DOFやモーションブラーへの影響が大きい
  - 映像のディティール感が失われやすい
- 総合的にはエイリアス除去のメリットの方が大きかった
  - 物理ベースレンダリングのスペキュラーエイリアスは異常

アンチエイリアスなし

アンチエイリアスあり

アンチエイリアスなし／あり

もう少し詳細な tips 等

- 履歴バッファの合成
  - モーションブラー用のベロシティマップで過去位置を求める
  - 深度等を比較して物理的に同じ場所とみなせる場合だけブレンド
    - 単純に過去位置のピクセルとブレンドすると残像が発生
  - あまりに距離が離れたピクセルは合成しない
- 残像は消しきれなかった

エッジ部分の高輝度スペキュラー問題

- 所詮10や20程度のサンプル数では全く解決できない
  - 露出調整後のピクセルが仮に100くらいの輝度になると
    - 100サンプル近くまで効果なし
    - 水平／垂直に近いエッジ以外はどれだけサンプル数を増やしても白か黒に…

エッジ部分の高輝度スペキュラー問題

- 解決策
  - ローパスフィルタなどの処置
    - やはりディティールは失われやすい
  - トーンマップ後の色空間で合成して最後に線形空間に戻す
    - トーンマップの段階でクランプされないように注意
      - 線形空間に完全に戻せる変換を使うこと
    - モニタ用ガンマ補正は適用しないように注意
      - ガンマ補正はあくまでモニタ出力ガンマに対応するための補正
      - ただしカラーグレーディングの一環としてのガンマカーブ調整は適用するのが正しい
    - 周波数の高い高輝度成分は本来より暗くなるので注意

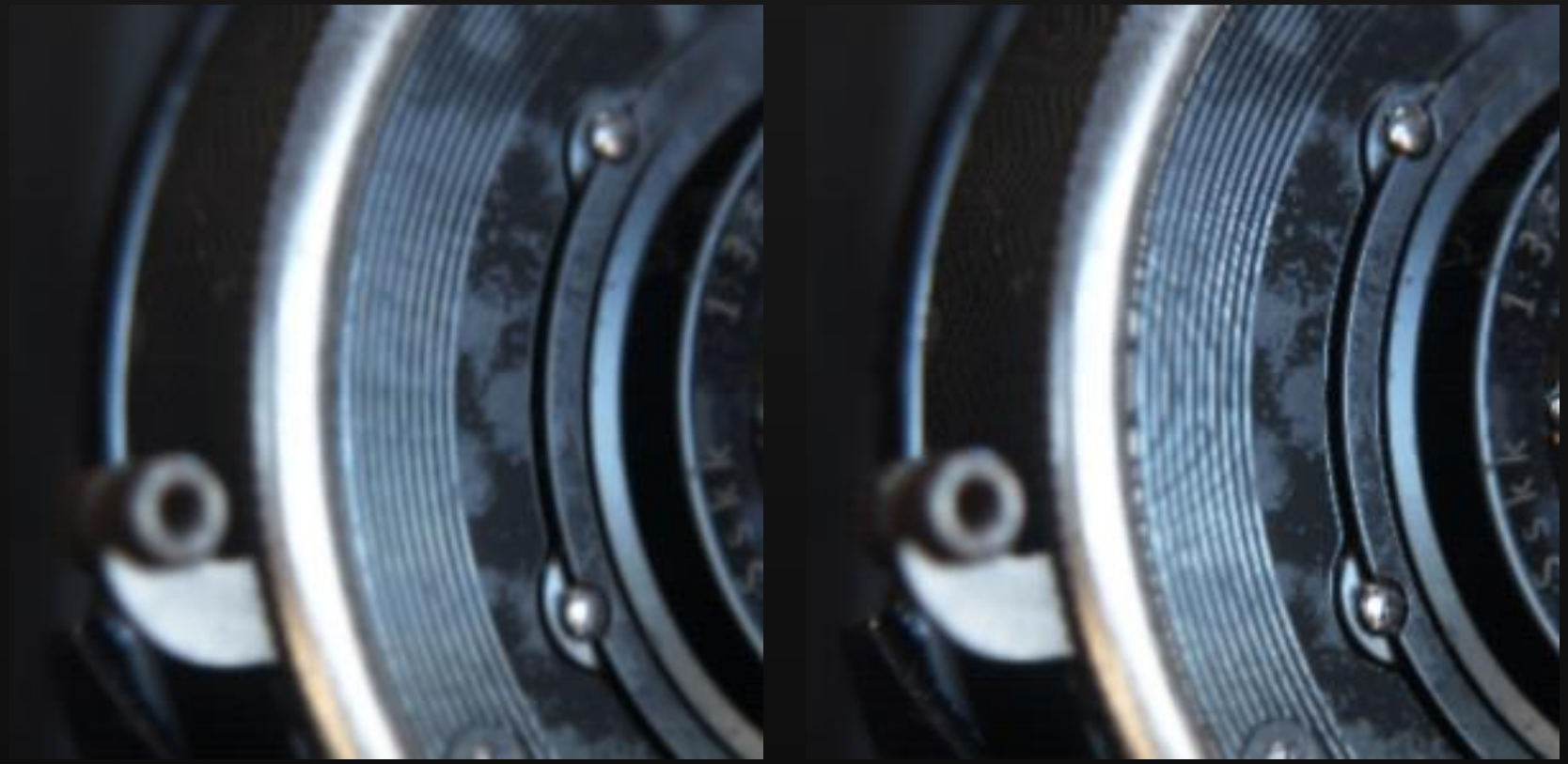
静止ピクセルの改善

- 静止ピクセルもスーパーサンプリングしたい
  - ここまでの方法だけでは動かないピクセルには全く効果なし
  - 画面が止まるとエイリアスが復活する

動いているピクセル

静止しているピクセル

動いているピクセル／静止ピクセル



テンポラル・ジッタリング

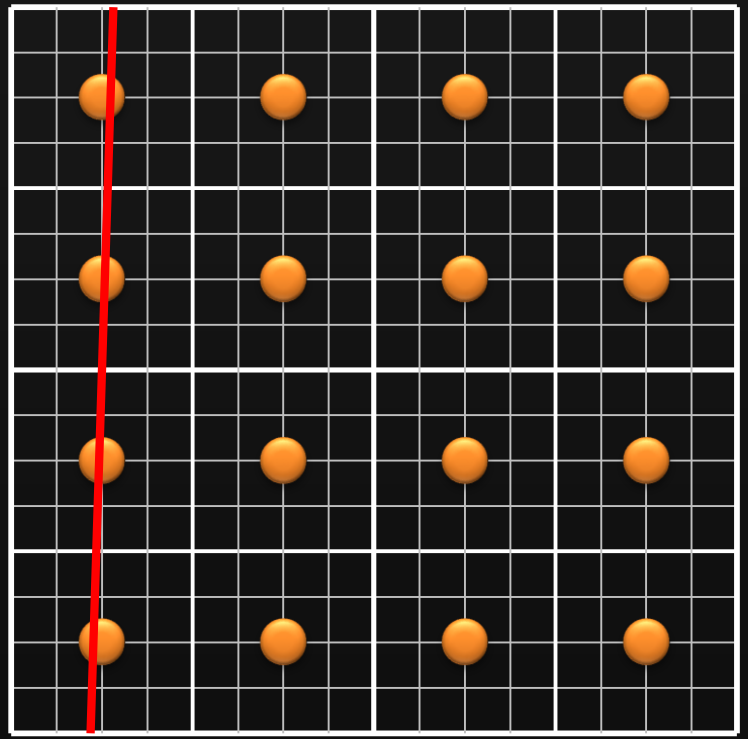
- 基本的な考え方
  - サブピクセルサンプリングを増やすために視錐体をジッタリング
    - 昔から知られるレガシーな手法
  - 1フレーム内ではなくフレーム毎にジッタリング
    - 複数フレームが徐々に合成されてサンプルが増える
- 最終的にデモには使用しなかった
  - 後述

サンプリング点の注意

- 理想的なサンプリング点
  - できるだけまばらに点在する
    - 偏ったサンプルにならないように
    - 乱数は偏りが大きいため有限サンプル数では望ましくない
  - どの方向のエッジが横切ってもまばらにサンプリング
    - グリッド状だと特定方向のエッジだけサンプリングが偏る
  - ある程度以上のサンプリング数
    - エイリアスを充分除去できるように
    - ただし増やし過ぎるとテンポラルではフリッカリングの原因に

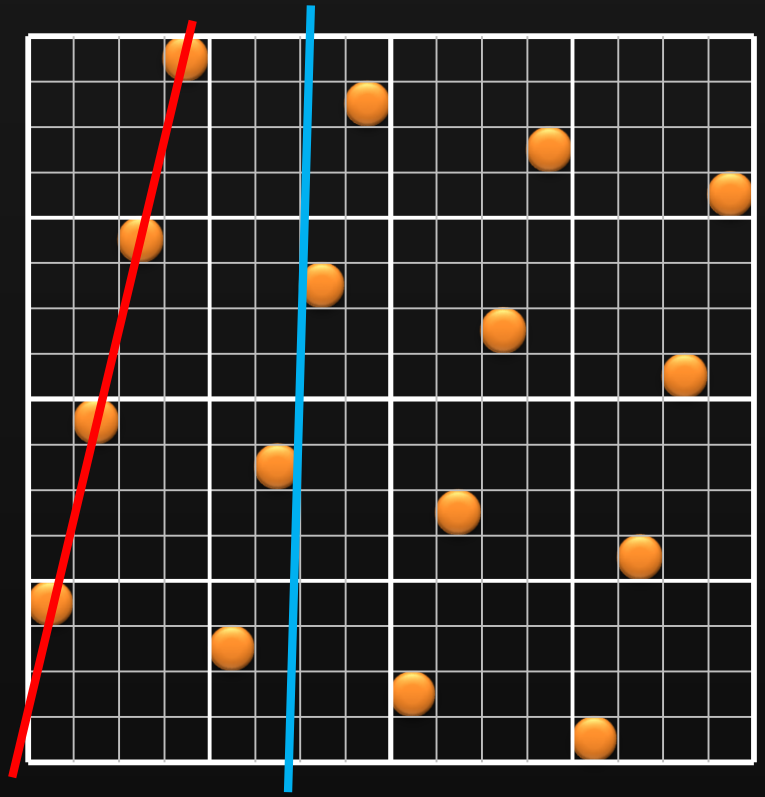
16サンプリングの例

- Ordered-Grid
- 水平／垂直に近いエッジに弱い
  - サンプルに偏りが発生する



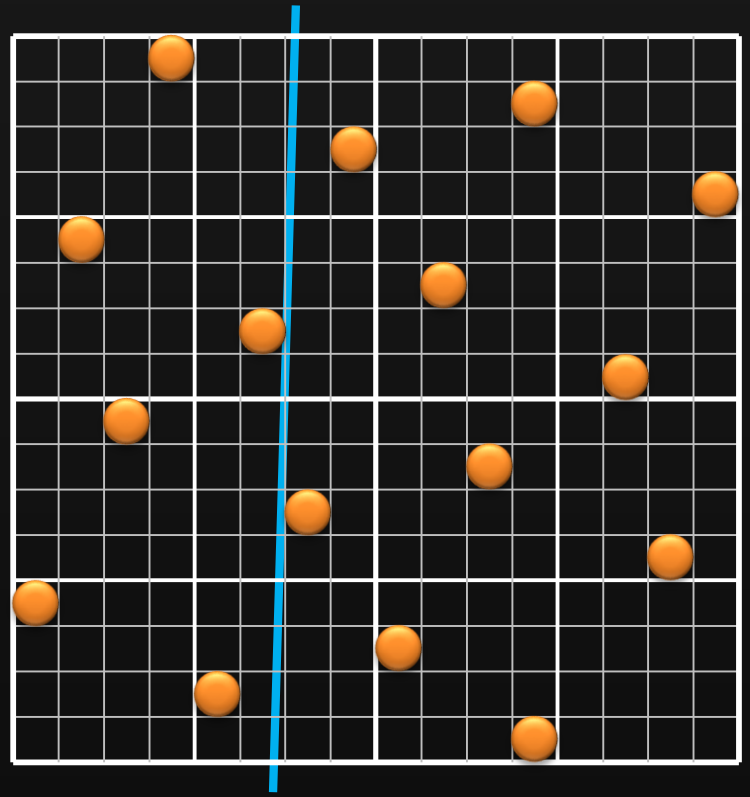
16サンプリングの例

- Rotated-Grid
  - N-Rooks 条件
- 水平／垂直に近いエッジに強い
  - 特定方向のエッジに弱い



16サンプリングの例

- Rotated-Grid を改善
  - N-Rooks 条件
- どの方向にもおおよそ強い
  - 水平／垂直に近いエッジに強い
  - 極端に弱い方向がない

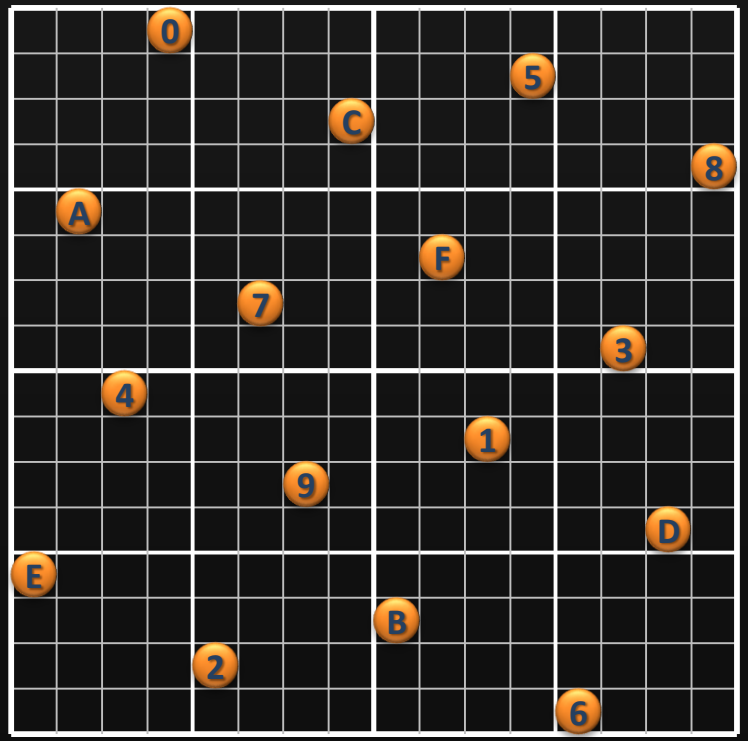


サンプリング点の注意

- サンプリングの順序の考察
  - 複数フレームに跨って合成されることを考慮
  - 任意のポイントから任意の数を合成しても偏りが少ないように
    - 例えば上から順にサンプリングすると上下方向に偏りが生じる
    - 複数フレームで徐々に合成されるため偏りが可視化される
  - 1フレーム内のジッタリングならこのような問題は起こらない
    - テンポラル処理の特性

採用したサンプリングパターン

- 実際には16サンプルでは不足
  - エッジ部のスペキュラーなど
  - あまり増やしてもフリッカリングが…
    - あまり強くブレンドできないため
- サブグリッドレベルでずらす
  - 16サンプル毎にパターンを変える



頂点単位ジッタリング

- 視錐体ジッタリング
  - 微妙に動いているピクセルがガタつく
  - 動いているピクセルにはジッタリング不要
    - すでにテンポラル・スーパーサンプリングが効いている
- 頂点単位ジッタリング
  - 頂点単位にベロシティベクトルを計算
  - 静止しているとみなせるピクセルだけジッタリングを行う

ジッタリングによるがたつき

- シーンによってはがたつきが目立つ
  - 被写界深度処理に悪影響がある
    - 深度によるマスク処理を行うため
    - 少しでも深度にばらつきがあると強いフリッカリングとなる
- 調整しきれなかったためデモ中では無効に
  - ⇒静止するとエイリアスが現れる
  - 静止部分への効果は絶大なので改善したい

テンポラル・アンチエイリアスまとめ

- 物理ベースレンダリングはエイリアス地獄！

- 非常に奥が深い
  - 今回はアドホックな調整が多かった

- まだまだまだ改善の余地あり
  - 残像の解消
  - ジッタリングをロバストに利用できるように
  - ディティール表現ができる限り失われないように
  - Etc.

ボケ味シミュレーション

絞り／収差シミュレーション

- 基本的な方法
  - 光学に基づいたボケ味の表現
    - [Kawase08] CEDEC 2008
    - [Kawase12] CEDEC 2012
      - フォーカス／前後ボケで最大15レイヤー
  - アンチ縮小バッファアーティファクト
    - [Kawase09] CEDEC 2009

絞り／収差シミュレーション

- 最適化
  - エッジ部分を検出してボケに対して重要なピクセルをマーク
    - 重要ピクセルのみをスキャッターベースで
    - 重要でないピクセルはギャザーベースで
  - 高速化により解像度を向上

ギャザーベース

ギャザー×スキャッター

カメラシミュレーション

カメラパラメータの制御

- アーティストが設定できるパラメータの管理
- ある程度カメラに似た仕組みを導入
  - レンズ固有のパラメータ
  - 動的な絞りパラメータ

カメラパラメータの制御

- レンズ毎の設定
  - 開放Ｆ値／画角による開放Ｆ値の変化
  - フォーカスグレージング（フォーカス距離による画角変化）
  - レンズ収差補正タイプ／収差の強さ
  - 絞り羽根枚数

- 動的な調整
  - 一眼レフのプログラムシフト的な動作
  - 露出によって絞りとシャッタースピードを自動調整
    - 絞りとシャッタースピードのバランスはアーティストがシフト値として調整
    - シャッタースピードはモーションブラーには影響させていない

カメラパラメータの制御

- これにより自然な表現が自動的に行われる
  - 明るさや画角によって絞りが自動的に調整される
  - 絞りの状態でボケ味が自動的に変化する
  - 開放Ｆ値との関係で円形絞りが自動的に変化する
  - 絞りの状態で光条の強さや長さが自動的に変化する

露出と絞りの変化（露出低）

露出と絞りの変化（露出高）

絞りと光条の変化（露出低）

絞りと光条の変化（露出高）

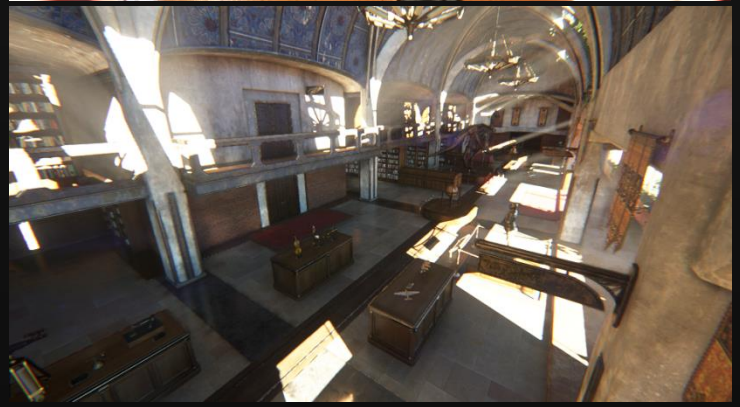
ポストプロセスまとめ

ポストプロセスまとめ

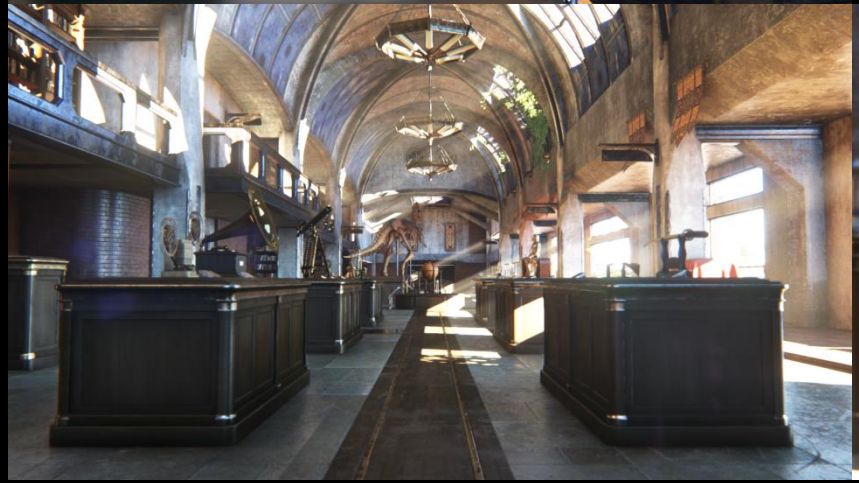
- 映像のリアリティが増すと従来よりもさらに重要に
  - 物理ベースではアンチエイリアスが必須
  - 被写界深度やモーションブラーが無いと不自然
  - カメラパラメータのアニメーションによって自然な表現に
    - 絞りの形状やブラーの強さが自動的に変化する

アジェンダ

- 第一章: デモの概要説明
- 第二章: 技術説明
  - Lighting/Shading
  - ポストエフェクト
- 第三章: まとめ








本日のまとめ

本日のまとめ

- 目標
  - まるで実写の映像作品のようなデモ
- リサーチ／開発
- 成果
  - 一定レベルの成果を得た
    - 少なくとも当初想定していた映像レベルは達成
  - 実装上のさまざまなノウハウを得た
    - 実装にあたっての想定内／想定外のさまざまな問題の発覚と解決方法
    - 本日紹介した他にもさまざまなノウハウの蓄積
  - とは言えまだまだ課題も多い

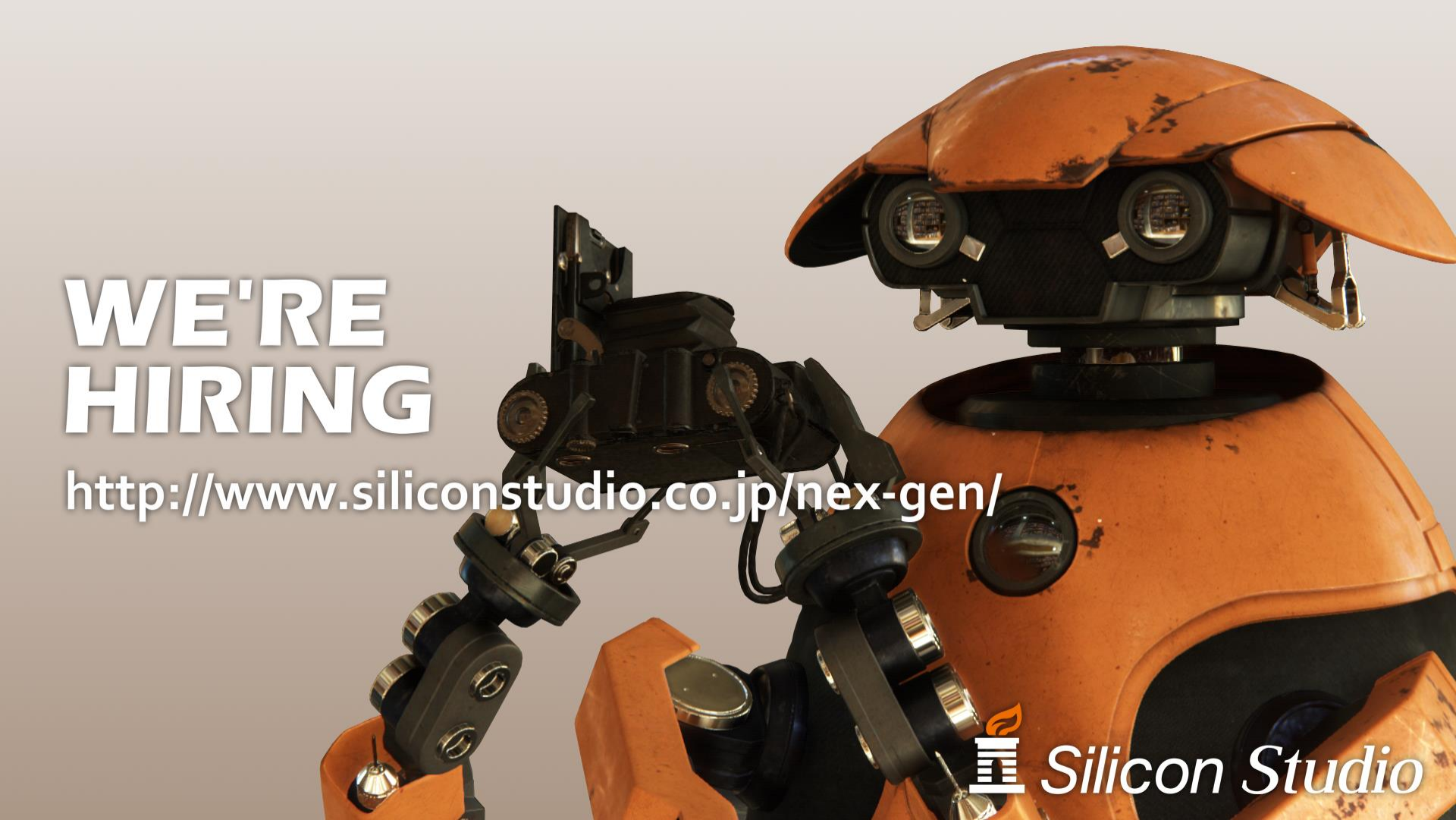
ご質問／ご感想は？

- 今回の発表を面白いと感じてくださった方
- PBRを実装してみようと思われている方
  - ノウハウを共有できることもあろうかと思います

- ご意見／ご感想／ご質問など何でも歓迎

お待ちしています！

re-yasuda@siliconstudio.co.jp


WE'RE
HIRING
http://www.siliconstudio.co.jp/nex-gen/
Silicon Studio

Shading/Lighting 参考文献

- [McAuley12] Stephen McAuley, “Calibrating Lighting and Materials in Far Cry 3”, SIGGRAPH 2012 Course: Practical Physically Based Shading in Film and Game Production
- [Meyer10] Quirin Meyer1, Jochen Süßmuth1, Gerd Sußner2, Marc Stamminger1 and Günther Greiner1 “On Floating-Point Normal Vectors” Eurographics Symposium on Rendering 2010
- [Karis13] Brian Karis, “Real Shading in Unreal Engine 4”, SIGGRAPH 2013 Course: Physically Based Shading in Theory and Practice
- [Neubelt13] David Neubelt, Matt Pettineo, “Crafting a Next-Gen Material Pipeline for The Order: 1886”, SIGGRAPH 2013 Course: Physically Based Shading in Theory and Practice
- [Valient14] Michal Valient “Taking Killzone Shadow Fall Image Quality into the Next Generation”, GDC 2014
- [Robison09] Austin Robison, Peter Shirly , “Image Space Gathering”, HPG 2009
- [Gotanda2013] Yoshiharu Gotanda, “Physically Based Reflectance Model Design for Next Generation”, CEDEC 2013
- [Gotanda2011] Yoshiharu Gotanda, “Real-time Physically Based Rendering - Implementation”, CEDEC 2011
- [Uludag14] Yasin Uludag, “Hi-Z Screen-Space Cone-Traced Reflections” GPU Pro 5
- [Sousa13] Tiago Sousa, “Graphics Gems from CryENGINE 3”, Siggraph 2013 Advances in Real-Time Rendering in Games Course
- [Lagarde12] Sébastien Lagarde “Lighting approaches and parallax-corrected cubemap” http://seblagarde.wordpress.com/2012/09/29/image-based-lighting-approaches-and-parallax-corrected-cubemap/
- [Toksvig04] Michael Toksvig, "Mipmapping Normal Maps", nVIDIA Technical Brief

ポストエフェクト 参考文献

- [Karis14] Brian Karis, *"High Quality Temporal Supersampling"*, SIGGRAPH 2014 Course: Advances in Real-time Rendering in Games: Part I
- [Wroński14] Bartłomiej Wroński, *"Temporal supersampling and antialiasing"*, http://bartwronski.com/2014/03/15/temporal-supersampling-and-antialiasing/
- [Kawase08] Masaki Kawase, "レンダリスト養成講座 *2.0* ～光学に基づいたボケ味の表現～", CEDEC 2008
- [Kawase09] Masaki Kawase, "続・レンダリスト養成講座 ～アンチ縮小バッファアーティファクト～", CEDEC 2009
- [Kawase12] Masaki Kawase, "実践！シネマティックレンズエフェクト ー光学に基づいたボケ味の表現ー", CEDEC 2012