© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

確率的ライトカリング

—基礎から動的コースティクスの描画まで—

徳吉 雄介

株式会社スクウェア・エニックス

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

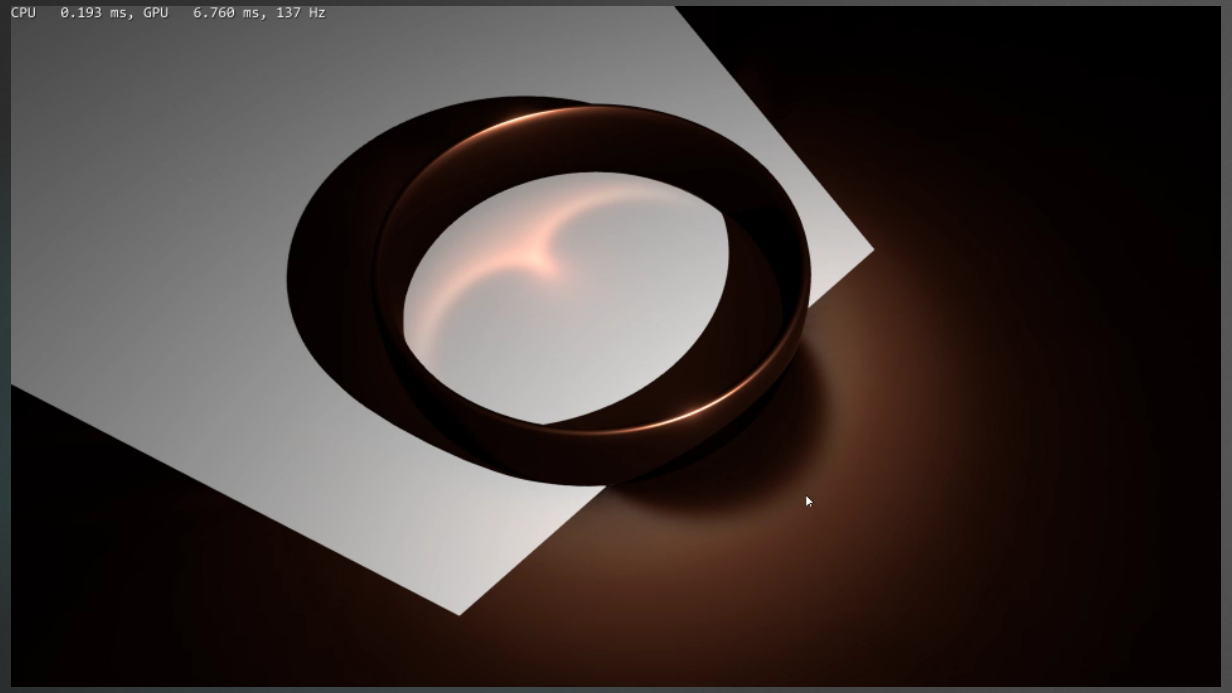
本講演の目的

- 数万の光源で照らされたシーンを比較的小さい誤差で描画する方法のご紹介
- Virtual Point Lights (VPLs)を使ったリアルタイムコースティクスの解説
  - 論文: Tokuyoshi and Harada “Stochastic Light Culling for VPLs on GGX Microsurfaces” (EGSR’17)
  - http://jp.square-enix.com/tech/publications.html

- 本講演の対象
  - レンダリングの基礎知識をもっている人
  - これからレンダリングの研究を始める人

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

光沢反射による動的コースティクス



解像度: 1920×1080
65536 VPLs
GPU: NVIDIA® GeForce® GTX 970

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

本日の流れ

- ライトカリングの説明
- 確率的ライトカリングの基礎
- 動的コースティクスへの拡張
  - GGX microfacet BRDFの反射特性に基づくカリング
- GPU実装
  - コースティクス向けのカリング手法

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

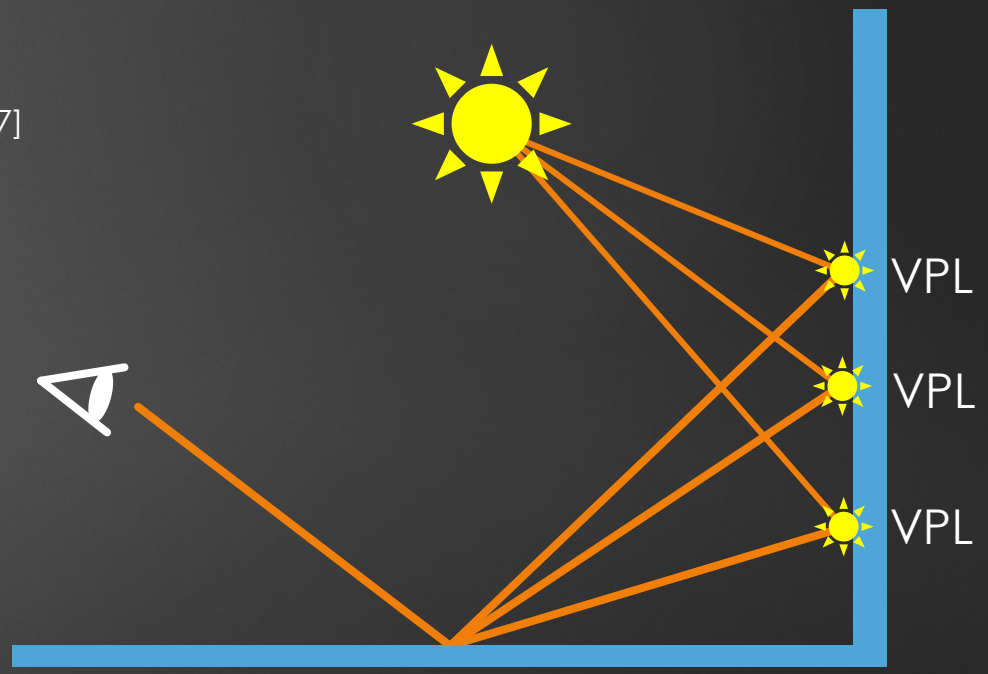
ライトカリング

LIGHT CULLING

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

多光源レンダリング

- 大量の光源を持つシーン
- 自動生成される無数の光源
  - 間接照明を表現するvirtual point lights (VPLs) [Keller97]
- まじめにやると膨大な計算量

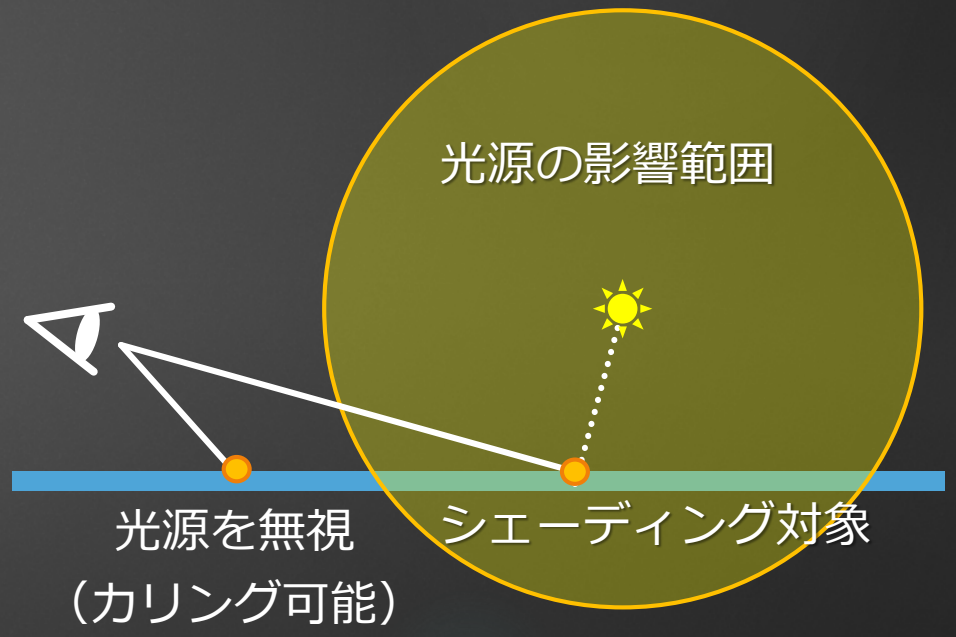


© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

ライトカリング

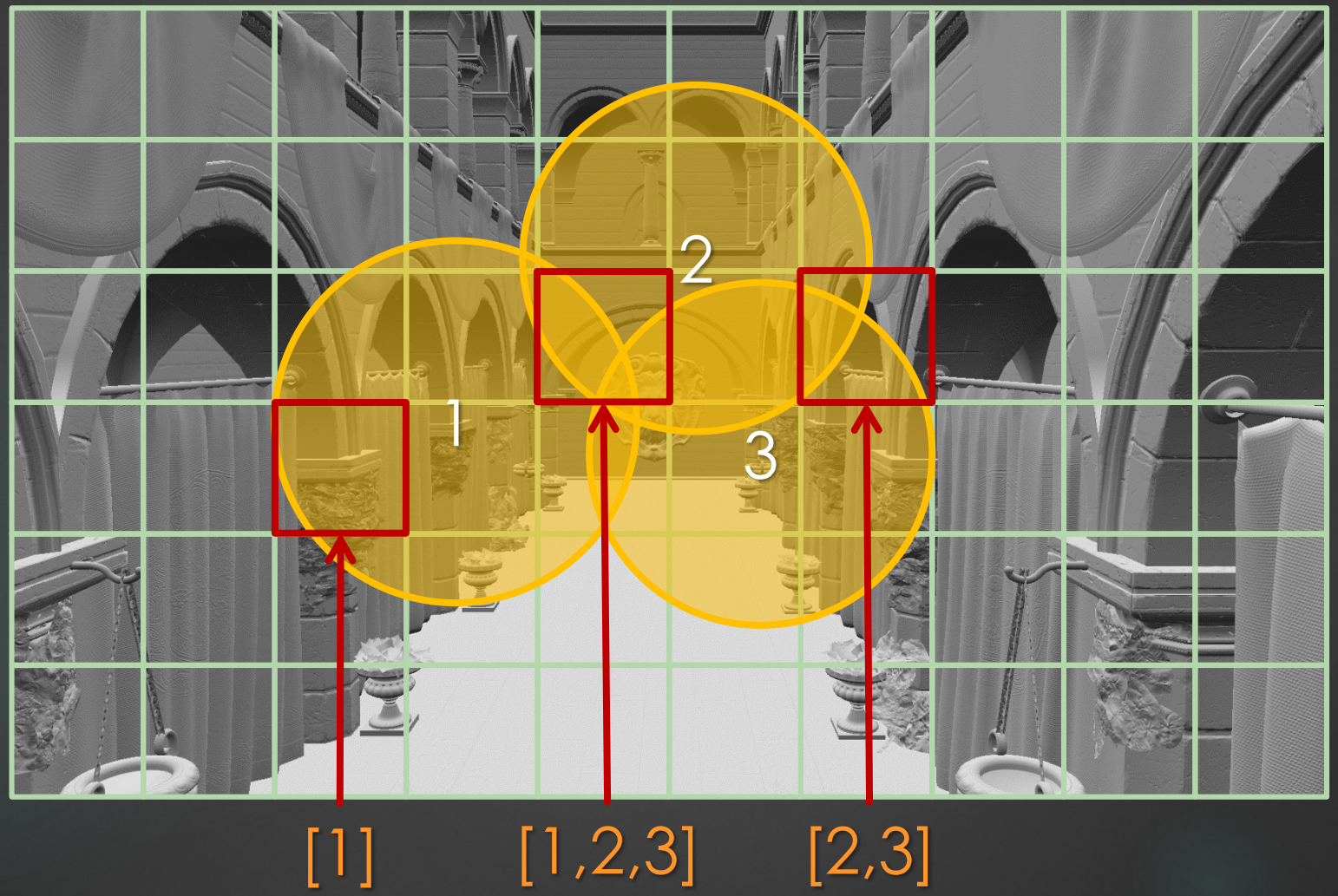
シェーディング点から遠い光源は寄与が小さいので無視

- 光源毎に光の影響範囲を制限
- 範囲外はシェーディングの対象外
- 光源を事前にカリングして高速化



© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

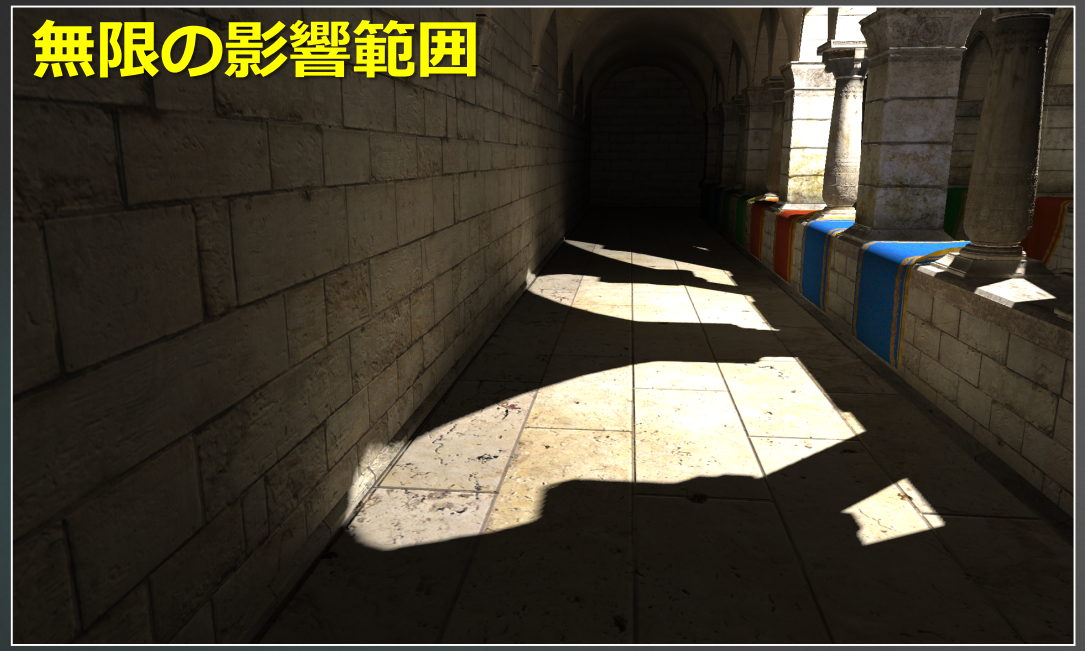
Tiled Culling [Olsson11, Harada12]

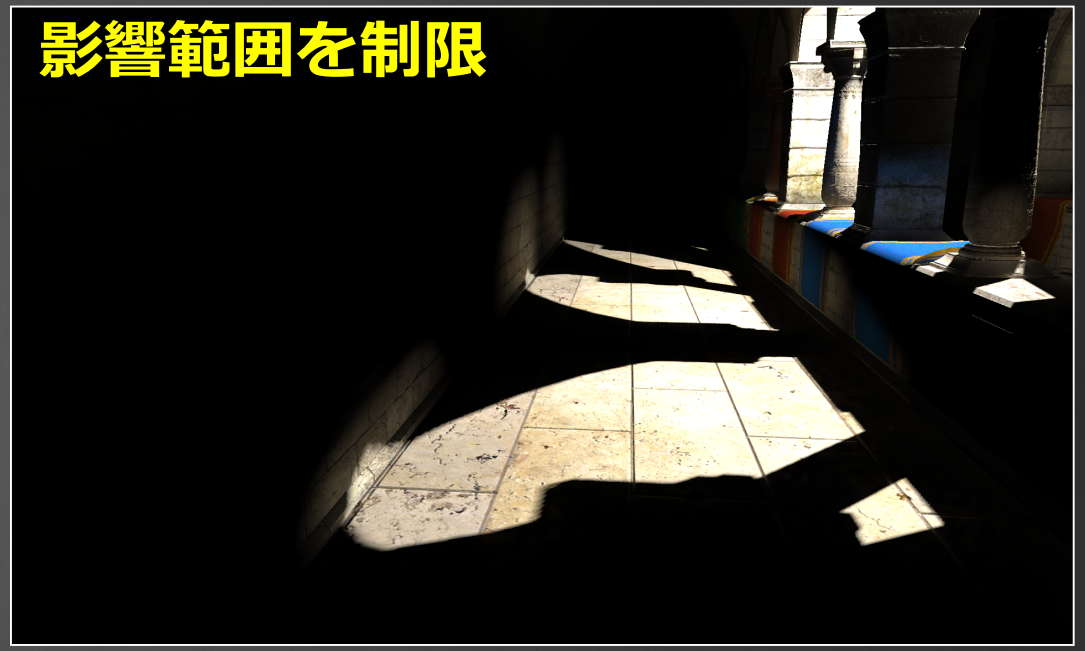


© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

問題点

65536VPLsを使った間接照明





光が遠くへ届かない☹

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

弱い光源が多くなるほど誤差(bias)が増加

各光源の明るさに応じて範囲を設定した場合…

1024 VPLs

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

弱い光源が多くなるほど誤差(bias)が増加

各光源の明るさに応じて範囲を設定した場合…

4096 VPLs

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

弱い光源が多くなるほど誤差(bias)が増加

各光源の明るさに応じて範囲を設定した場合…

16384 VPLs

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

弱い光源が多くなるほど誤差(bias)が増加

各光源の明るさに応じて範囲を設定した場合…

65536 VPLs

速度を維持しようとすると暗くなる

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

Bias

- 解に対する推定値の偏りを表す誤差
- ライトカリングは全光源が暗めに偏っている
- Biasのある推定が常に解に収束しないとは限らないがこの場合は収束しない☹

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

カリング効率と誤差

- レンダリングは速度と誤差のトレードオフ
- 速いだけでなく、少ない誤差であることも重要

65536 VPLs (shading time: 2.87 ms)

Resolution: 1920×1152, Error bound: 0.0005, GPU: AMD Radeon™ R9 290X

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

確率的ライトカリング

STOCHASTIC LIGHT CULLING

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

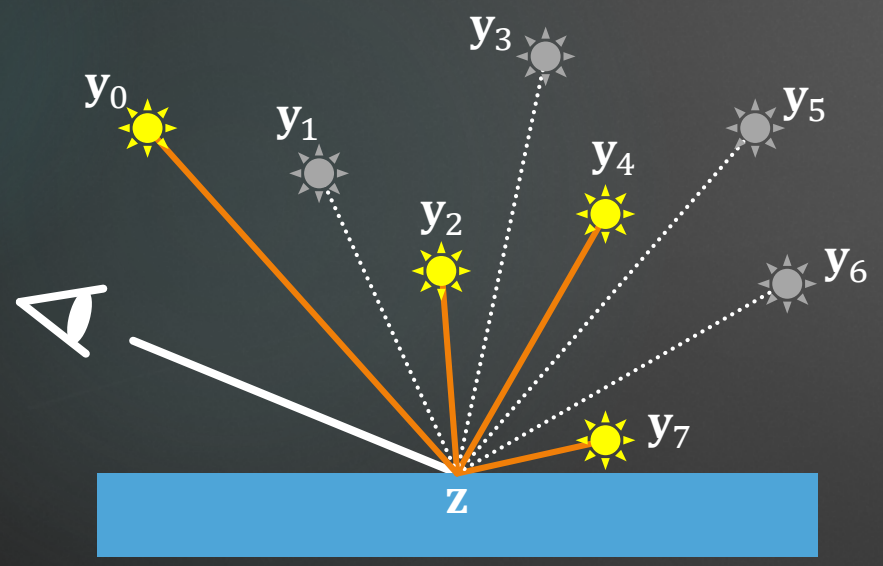
手法の概要

- ロシアンルーレット法[Arvo90a]を基に光の影響範囲をランダムに決定
  - 遠い光源も低確率でサンプリング
- Biasを分散（推定値のばらつきを表す誤差）に変える
- 光源数が多くなっても誤差が累積しない

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

ロシアンルーレット法

- 各光源を確率的に殺す
- 生存確率: シェーディング点から見た光源の明るさに比例（距離二乗減衰）
- 生き残った光源のエネルギーは生存確率で割る



各光源の放射強度

生存確率: $P(\mathbf{y}_j, \mathbf{z}) = \min\left(\frac{I(\overrightarrow{\mathbf{y}_j\mathbf{z}})}{\delta\|\mathbf{y}_j - \mathbf{z}\|^2}, 1\right)$

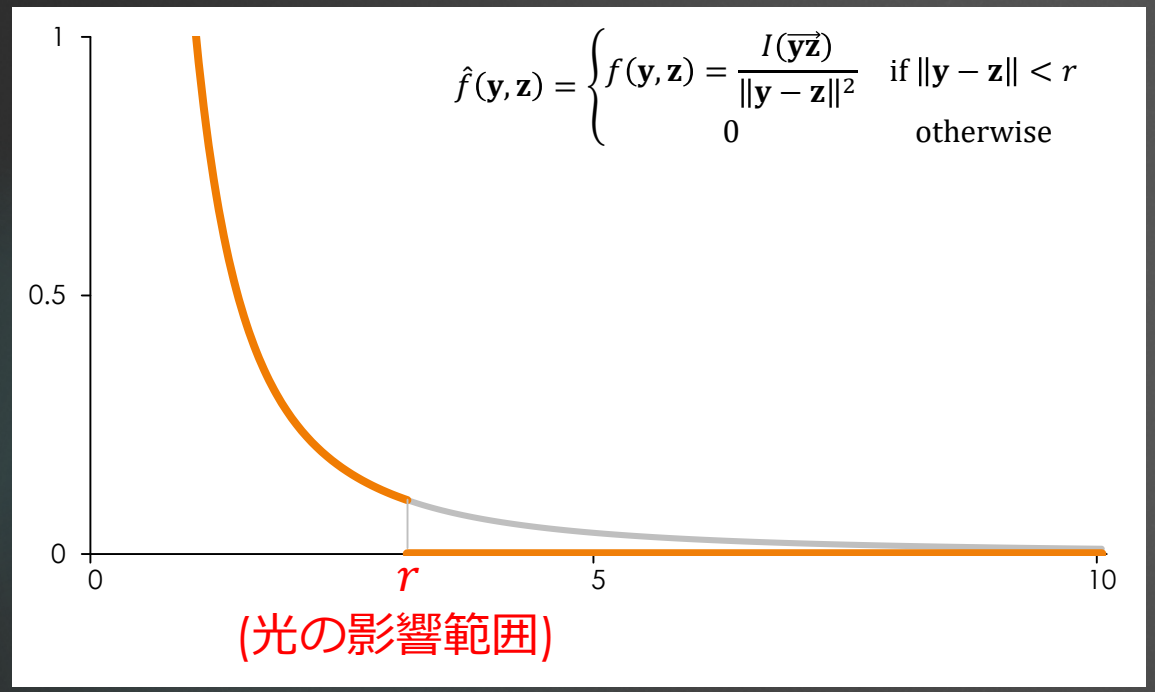
許容誤差量
（定数）

各光源までの距離

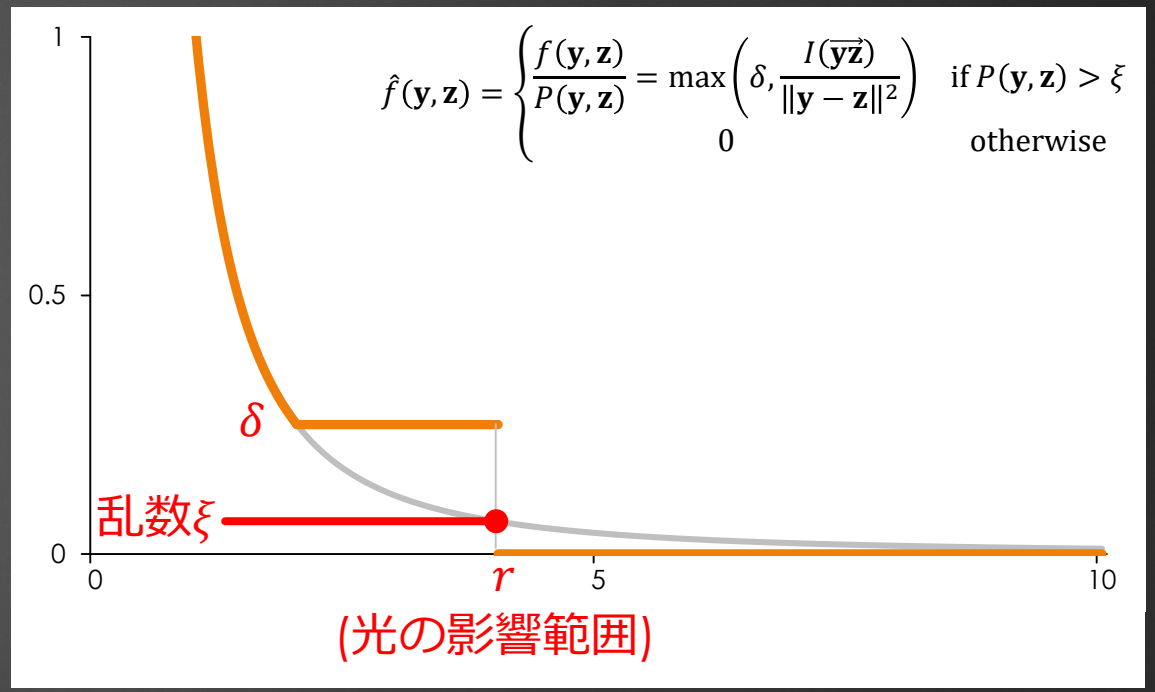
© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

ロシアンルーレット後の光源の明るさ

通常のライトカリング



確率的ライトカリング

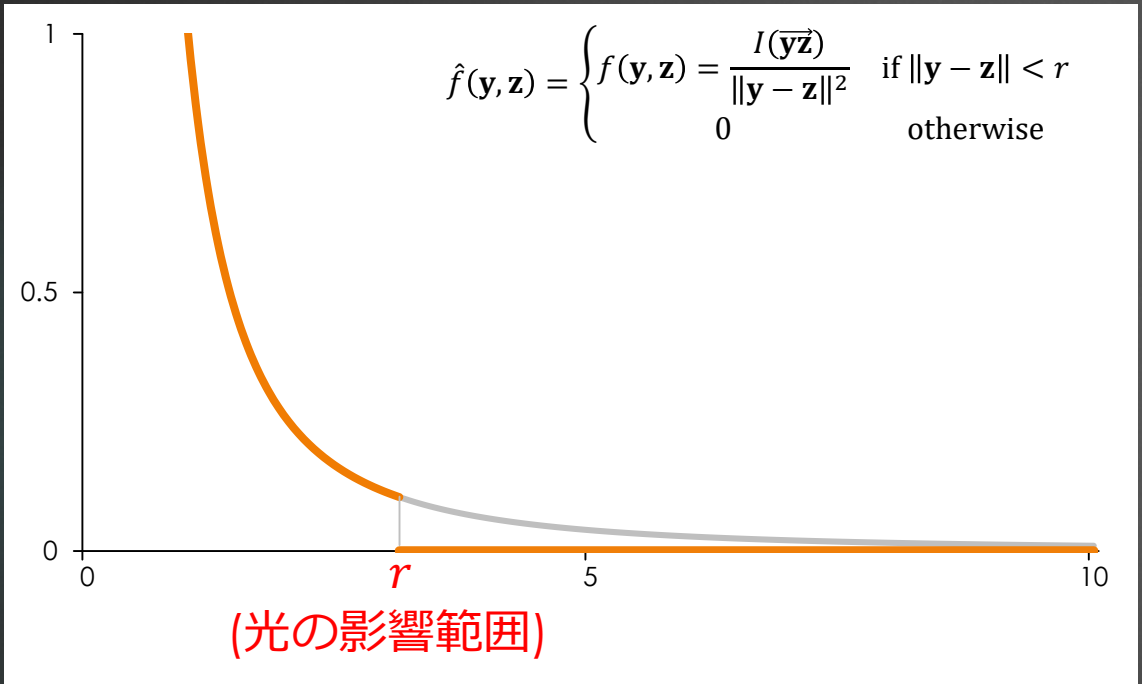


横軸: 光源からの距離$\|\mathbf{y}-\mathbf{z}\|$

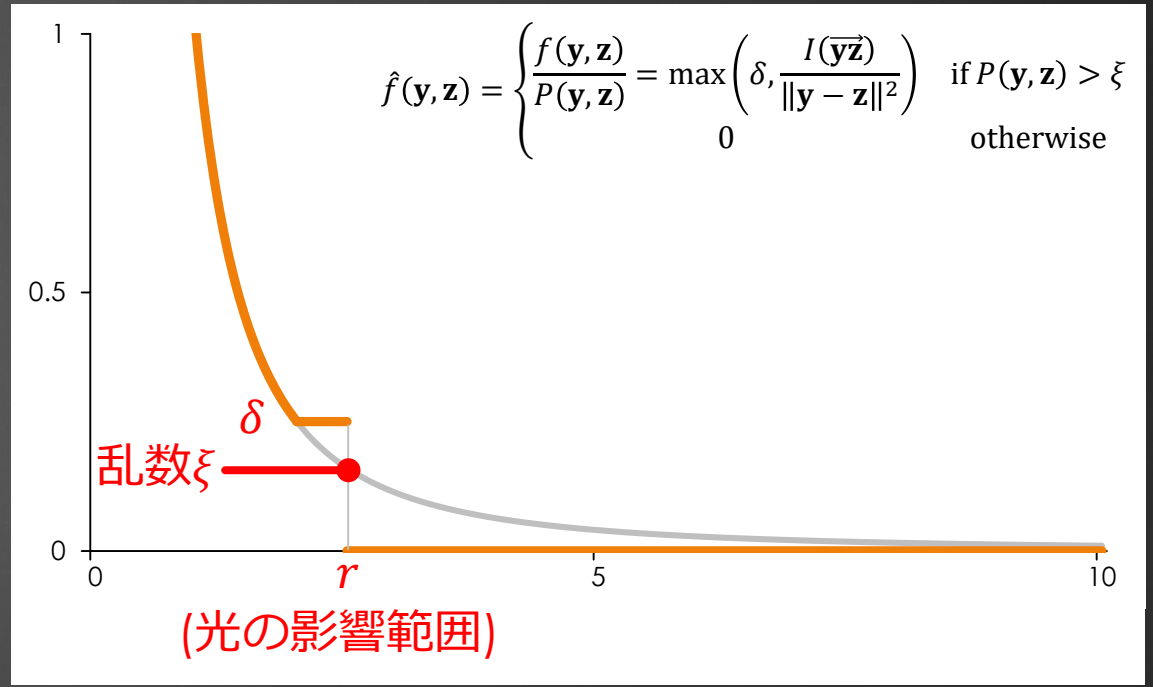
© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

ロシアンルーレット後の光源の明るさ

通常のライトカリング



確率的ライトカリング

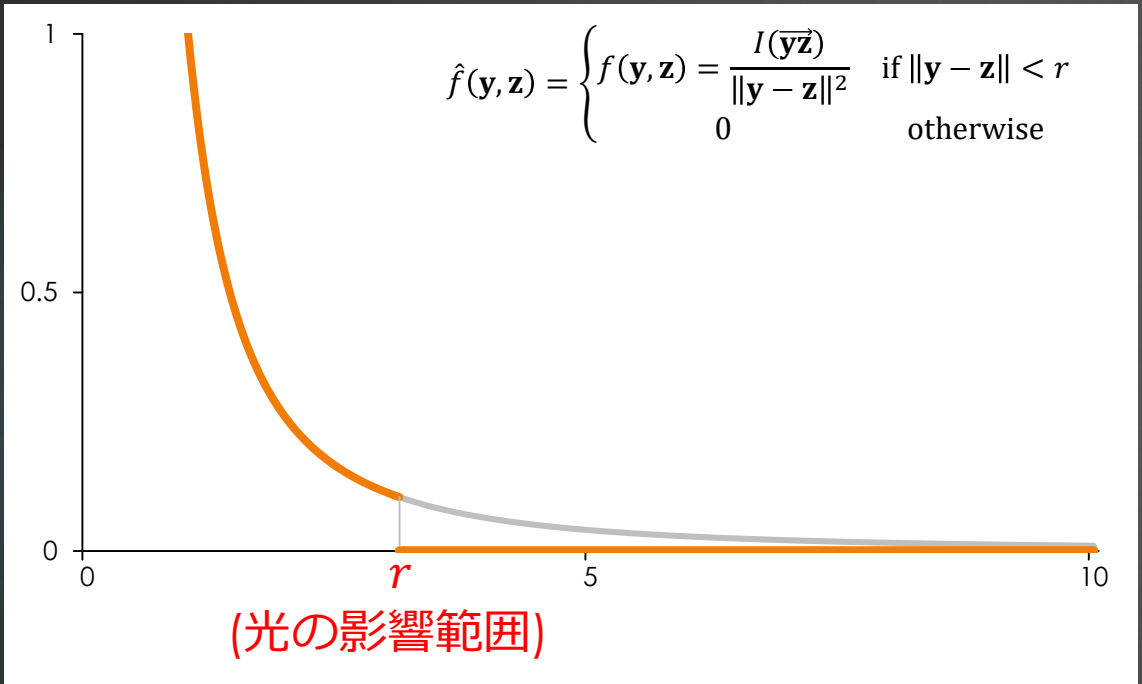


横軸: 光源からの距離$\|\mathbf{y}-\mathbf{z}\|$

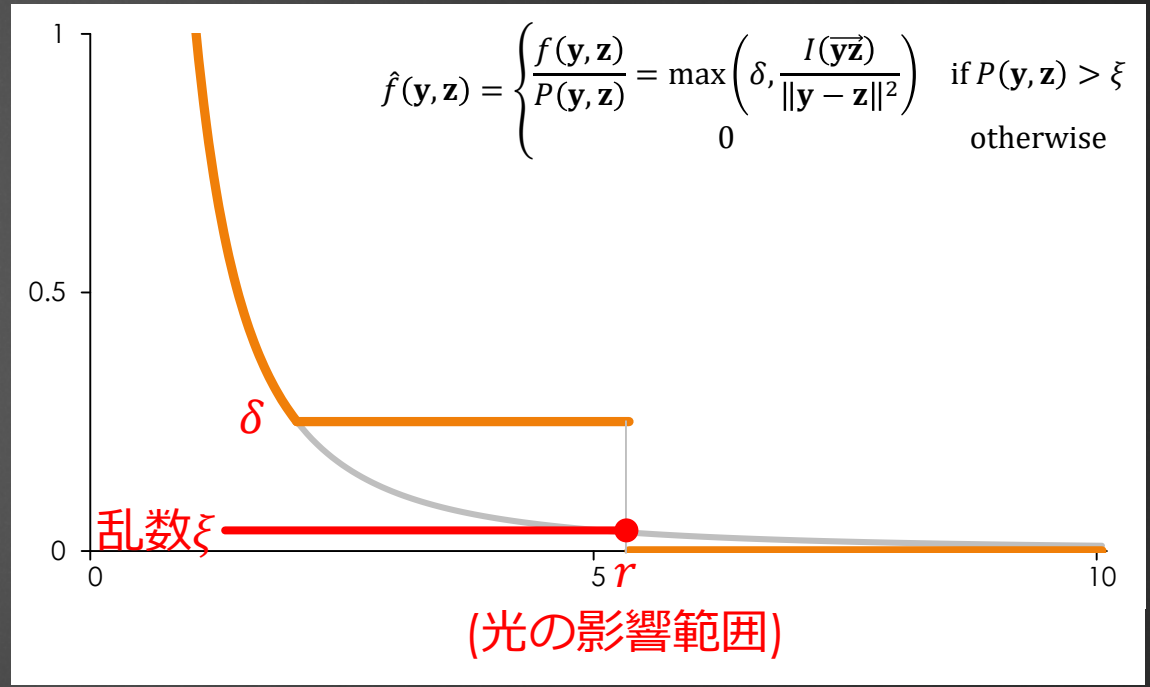
© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

ロシアンルーレット後の光源の明るさ

通常のライトカリング



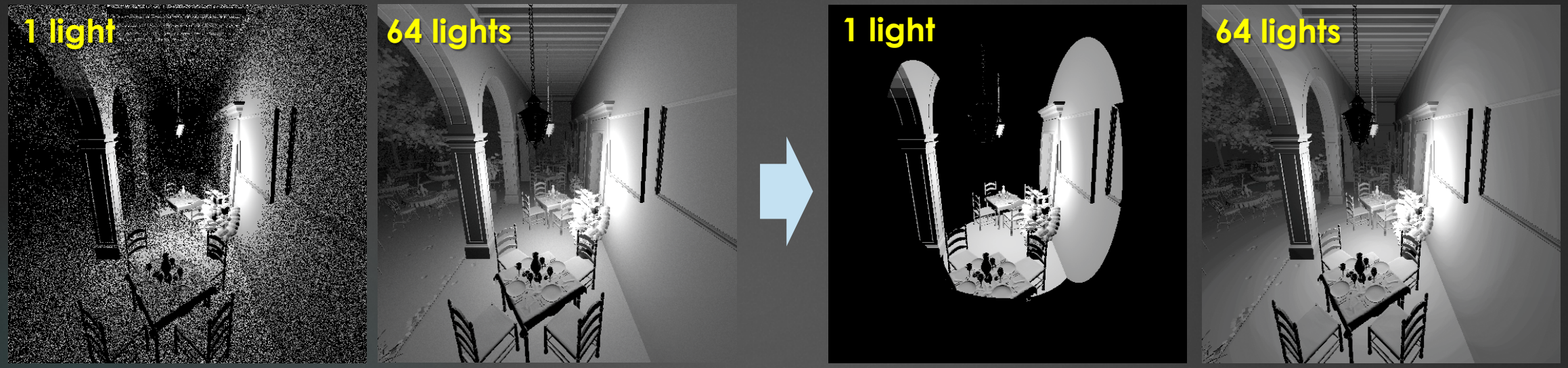
確率的ライトカリング



横軸: 光源からの距離$\|\mathbf{y}-\mathbf{z}\|$

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

ランダムな光の影響範囲



- シェーディング点毎に光の影響範囲が異なる。カリングできない☹
- 共通の範囲を使いたい
- 解決法: **全シェーディング点で共通の乱数を使用**
  - 光源毎に異なる乱数であれば誤差は偏りなくばらつく (unbiased ☺)
  - 分散はバンディングアーティファクトとして画面に現れる

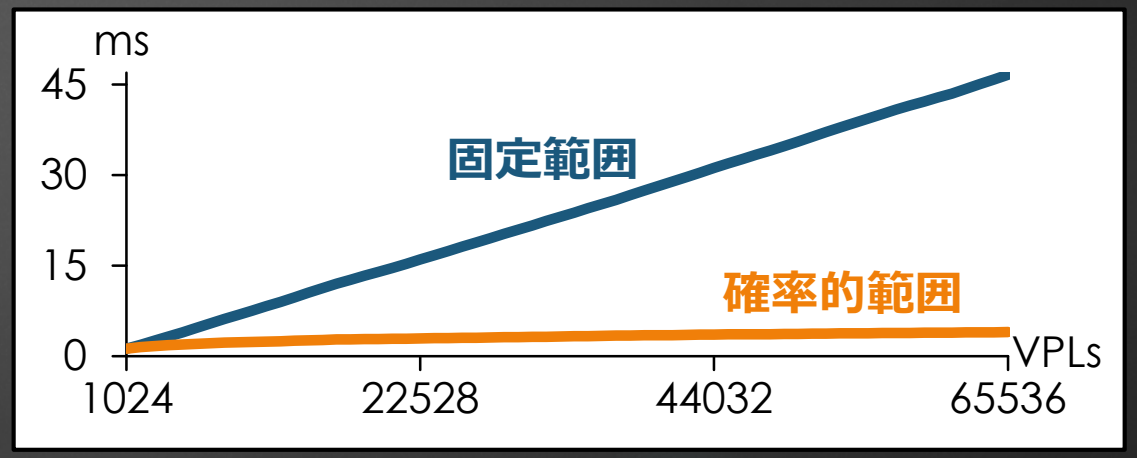
© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

許容誤差の設定

- 許容誤差δを設定してオーバーサンプリングを防ぐ
- 暗い光源ほど低確率 → 影響範囲が小さくなる
- 大きな影響範囲を持つ光源も存在するが、極めて少数
- Virtual Point Lights (VPLs)だと最終的に生き残る光源数はほぼ一定に収束 ☺

$$P(\mathbf{y},\mathbf{z}) = \min\left(\frac{I(\overrightarrow{\mathbf{yz}})}{\delta\|\mathbf{y}-\mathbf{z}\|^2}, 1\right)$$

生存確率



同じ許容誤差でのシェーディング時間

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

処理の流れ

乱数ξを生成
ロシアンルーレット法を基に光源の影響範囲を決定
(光源毎の処理)

↓

Light culling
(既存手法)

← カリング手法に依存しない

Splatting [Dachsbacher06]
Tiled culling [Olsson11; Harada12]
Clustered culling [Olasson12]
...etc.

↓

Shading
既存手法と同じ方法でシェーディング
光源毎に与えられた乱数ξを使ってロシアンルーレット

赤字が新たに追加された処理

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

結果: 拡散反射間接照明 (65536 VPLs)

解像度: 1920×1080, GPU: NVIDIA® GeForce® GTX 1070

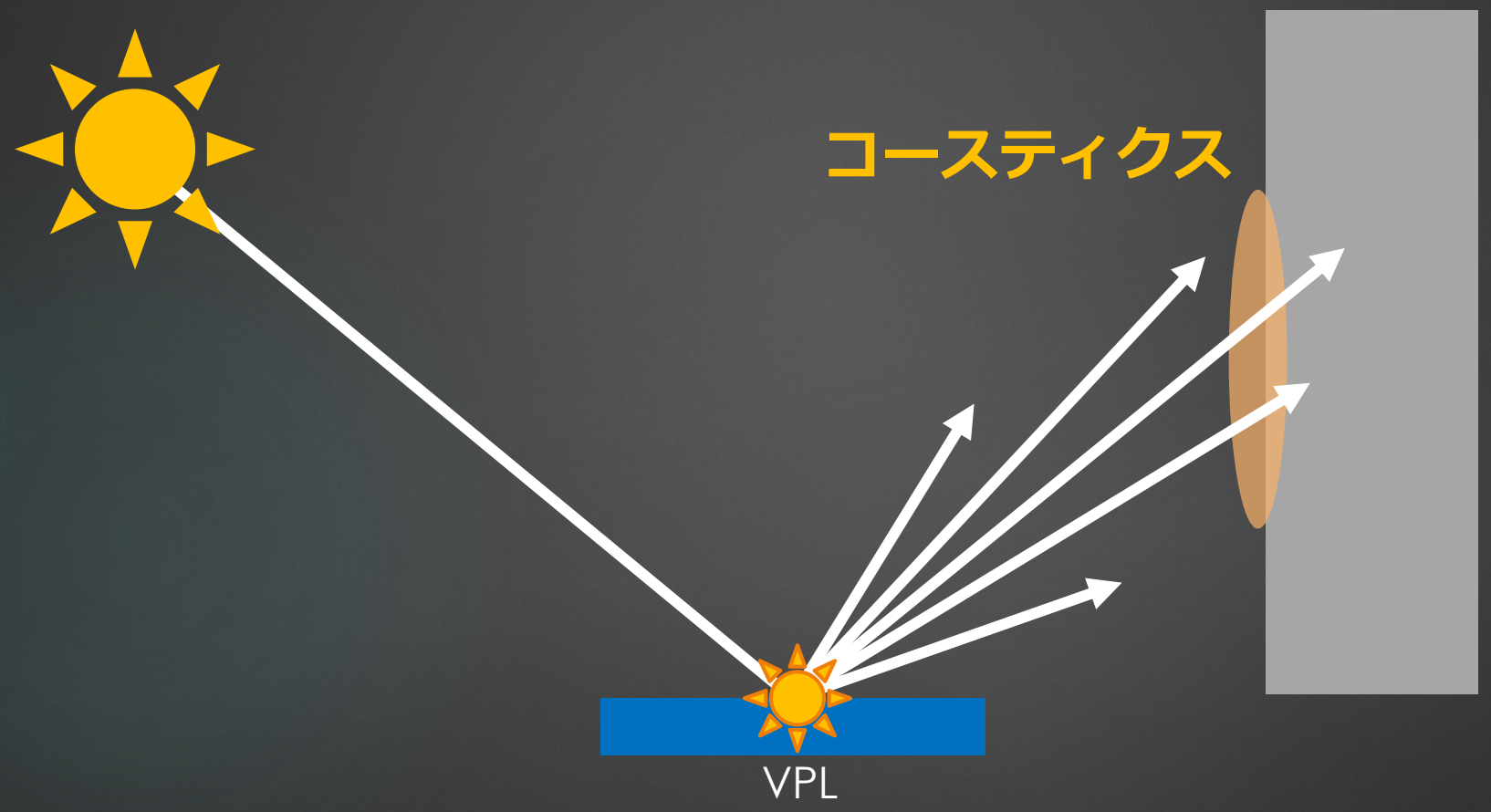
© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

光沢反射コースティクスへの拡張

EXTENSION TO DYNAMIC GLOSSY CAUSTICS

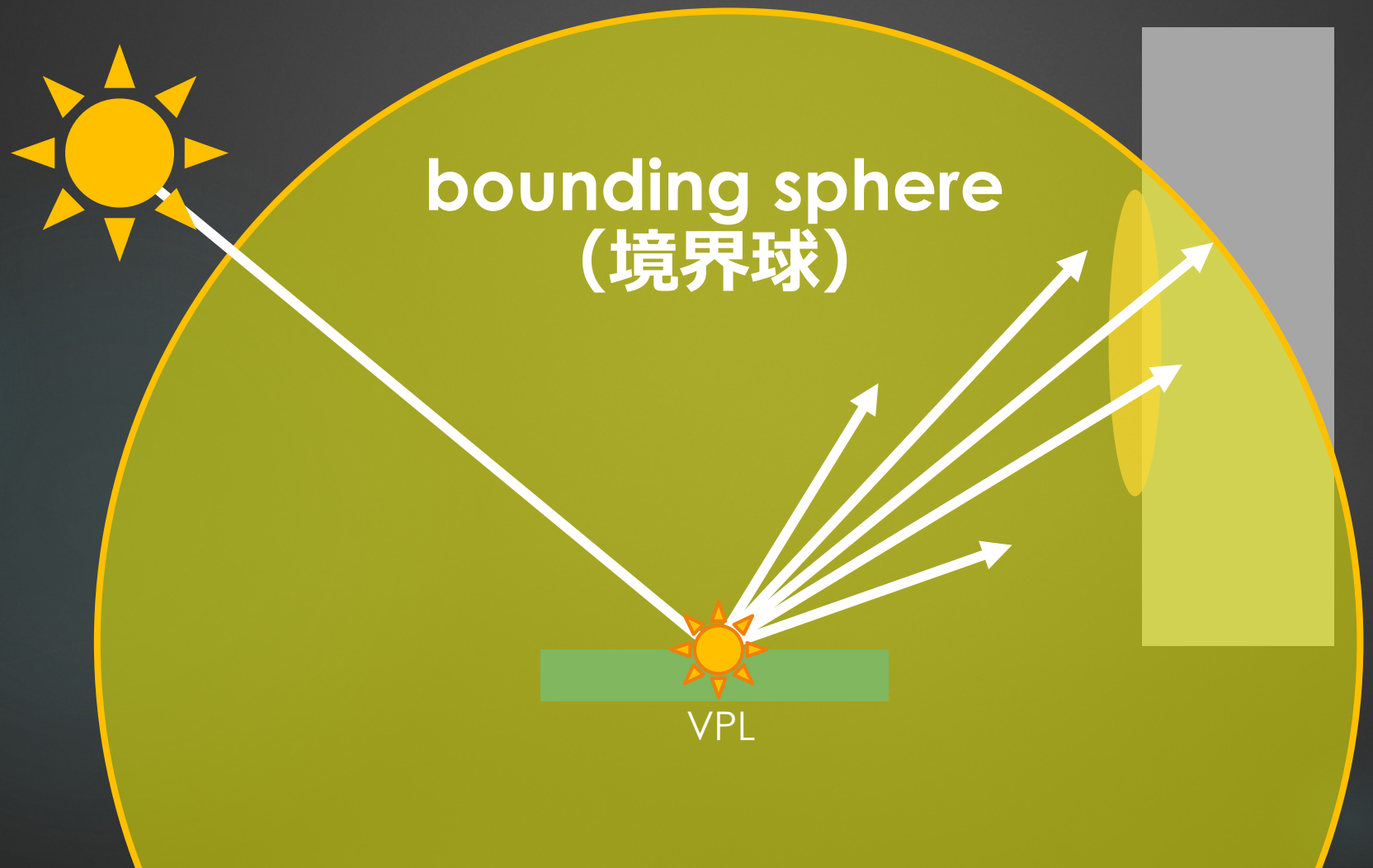
© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

光沢反射面上のVirtual Point Lights (VPLs)



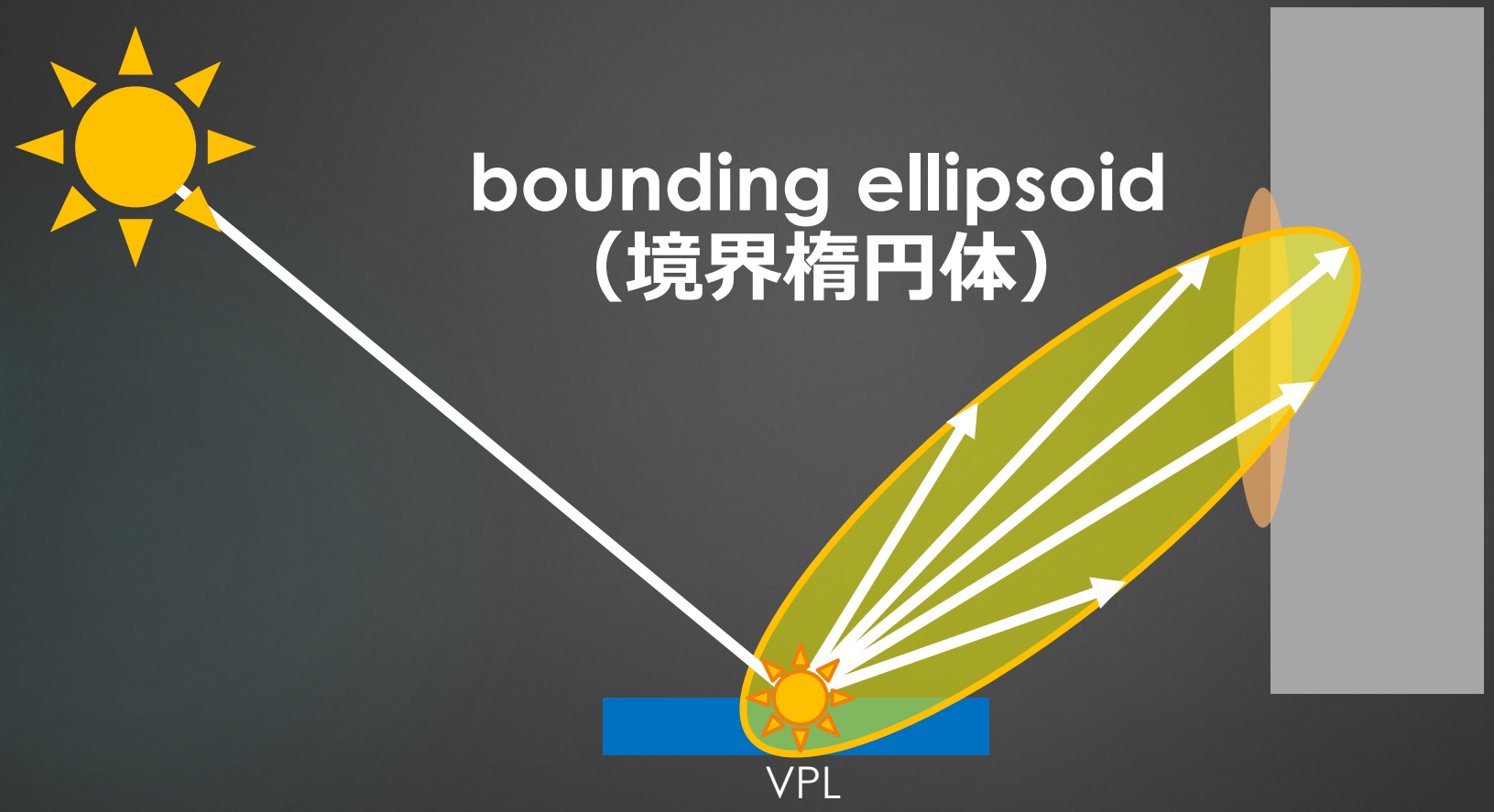
© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

光沢反射面上のVirtual Point Lights (VPLs)



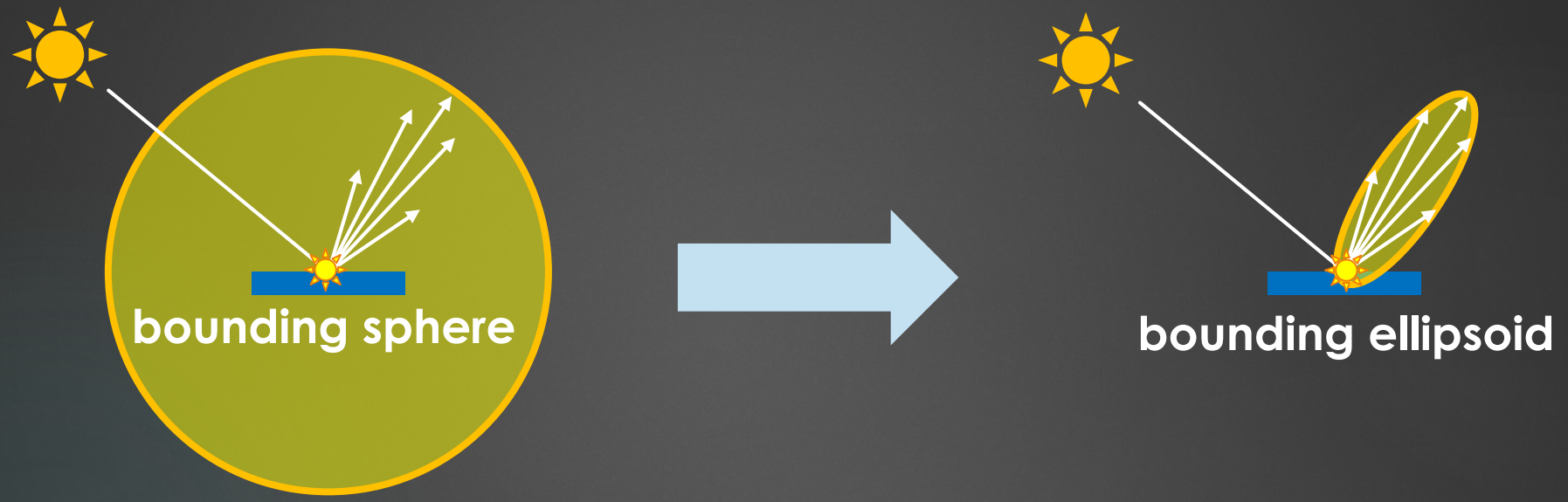
© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

光沢反射面上のVirtual Point Lights (VPLs)



© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

楕円体を使ったカリング



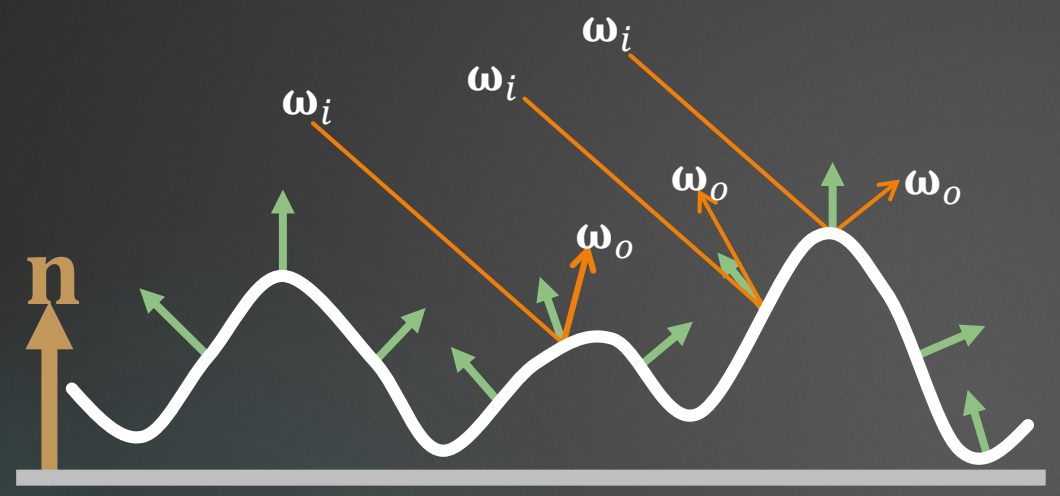
- 楕円体を使ったVPLのカリング[Dachsbacher06]はあったが
- 解析的な楕円体の構築は古典的なPhong BRDF [1975]が対象
- 最近の「物理的にもっともらしい材質」（例えばGGX）には使えない ☹

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

GGX鏡面反射に対して境界楕円体を求めたい

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

GGX Microfacet BRDF [Walter07]



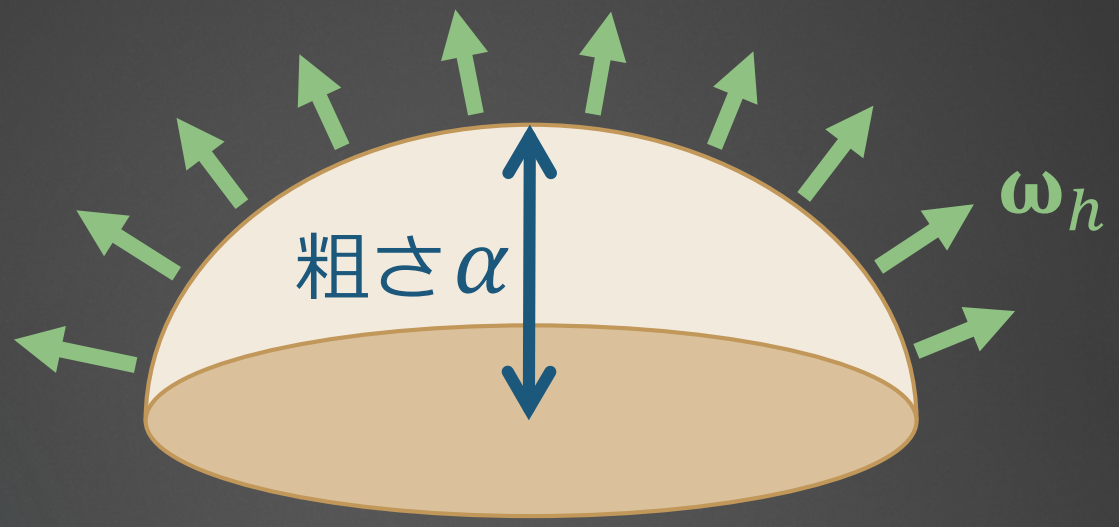
- 目に見えないくらい細かい凹凸のある鏡面 [Cook82]
- GGX法線分布関数で凹凸を表現

$$f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_h) F(\boldsymbol{\omega}_h \cdot \boldsymbol{\omega}_i) D(\boldsymbol{\omega}_h \cdot \mathbf{n})}{4|\boldsymbol{\omega}_i \cdot \mathbf{n}||\boldsymbol{\omega}_o \cdot \mathbf{n}|}$$

凹凸面の法線
（入射方向$\boldsymbol{\omega}_i$と反射方向$\boldsymbol{\omega}_o$の中間方向）

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

GGX分布＝半楕円体の法線分布[Trowbridge75]



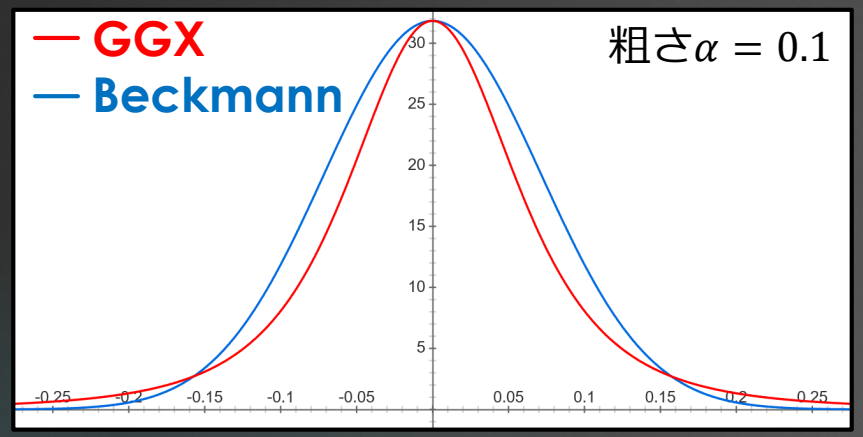
GGX: $$D(\boldsymbol{\omega}_h \cdot \mathbf{n}) = \frac{\alpha^2 \chi^+(\boldsymbol{\omega}_h \cdot \mathbf{n})}{\pi(1 + (\alpha^2 - 1)(\boldsymbol{\omega}_h \cdot \mathbf{n})^2)^2}$$

(Ground Glass Unknown)
[Walter14]

Heaviside関数: $\chi^+(\boldsymbol{\omega}_h \cdot \mathbf{n}) = \begin{cases} 1 & \text{if } \boldsymbol{\omega}_h \cdot \mathbf{n} > 0 \\ 0 & \text{otherwise} \end{cases}$

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

GGX法線分布の特徴

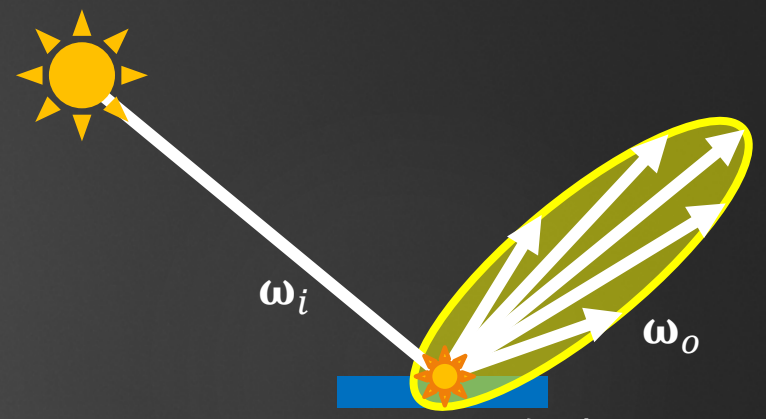


- 粗さ$\alpha = 1$で等方分布、 $0 < \alpha < 1$ で中心からの角度に対して単調減少
- BeckmannやPhongの法線分布（≠ Phong BRDF）よりテールが強い
- Smith[1967]のmasking-shadowing function $G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_h)$が解析的に求まる ☺
- 効率的なインポータンスサンプリング[Heitz14]が可能 ☺

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

反射光の影響範囲

- ロシアンルーレットの生存確率の逆関数
- ランダムに選ばれた反射放射輝度の等値面



フォトンの放射束Φ

$$l_{\max}(\boldsymbol{\omega}_o) = P_{\boldsymbol{\omega}_i}^{-1}(\xi) = \sqrt{\frac{\Phi f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)\max(\boldsymbol{\omega}_o \cdot \mathbf{n}, 0)}{\delta\xi}}$$

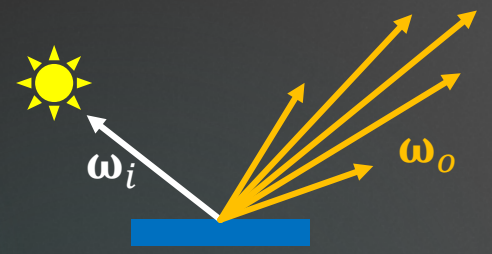
反射ローブ

一様乱数∈ [0,1)

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

反射ローブの上限

$$f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)\max(\boldsymbol{\omega}_o \cdot \mathbf{n}, 0) = \frac{G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_h)F(\boldsymbol{\omega}_h \cdot \boldsymbol{\omega}_i)D(\boldsymbol{\omega}_h \cdot \mathbf{n})}{4|\boldsymbol{\omega}_i \cdot \mathbf{n}|\cancel{|\boldsymbol{\omega}_o \cdot \mathbf{n}|}}\max(\cancel{\boldsymbol{\omega}_o \cdot \mathbf{n}}, 0)$$



反射ローブ

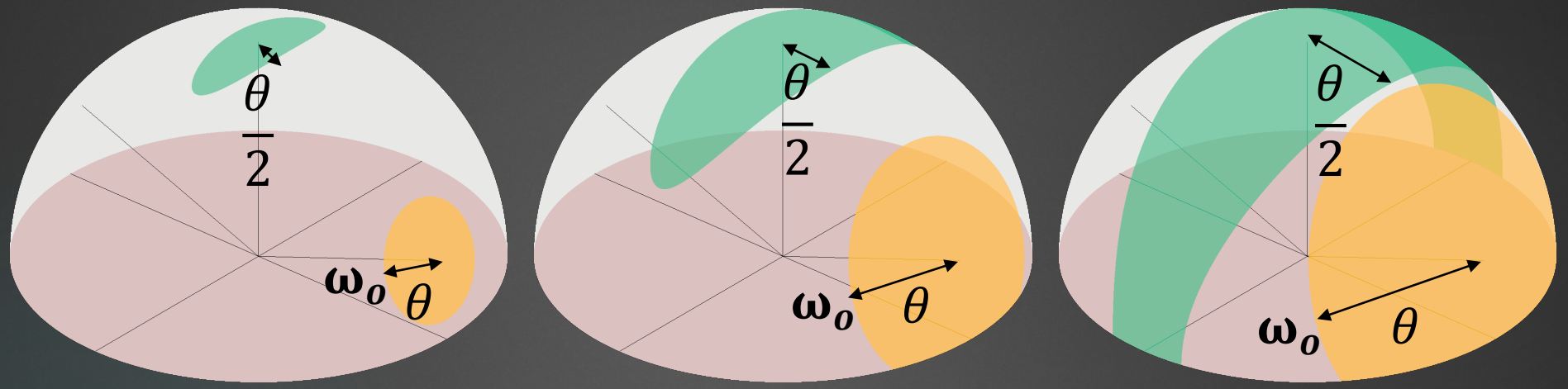
$$\leq \frac{\overbrace{G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_h)F(\boldsymbol{\omega}_h \cdot \boldsymbol{\omega}_i)}^{\text{低周波}}D(\boldsymbol{\omega}_h \cdot \mathbf{n})}{4|\boldsymbol{\omega}_i \cdot \mathbf{n}|}$$

$$\leq \frac{\overbrace{G_{\max}(\boldsymbol{\omega}_i)F_{\max}(\boldsymbol{\omega}_i)}^{\text{最大値}}D(\boldsymbol{\omega}_h \cdot \mathbf{n})}{4|\boldsymbol{\omega}_i \cdot \mathbf{n}|}$$

中間方向$\boldsymbol{\omega}_h$だけに依存

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

中間方向$\boldsymbol{\omega}_h$と反射方向$\boldsymbol{\omega}_o$の関係



反射方向$\boldsymbol{\omega}_o$が完全鏡面反射方向を中心とする球面円上にあるとき

中間方向$\boldsymbol{\omega}_h$は球面楕円 or 球面双曲線上にある [Jakob14]

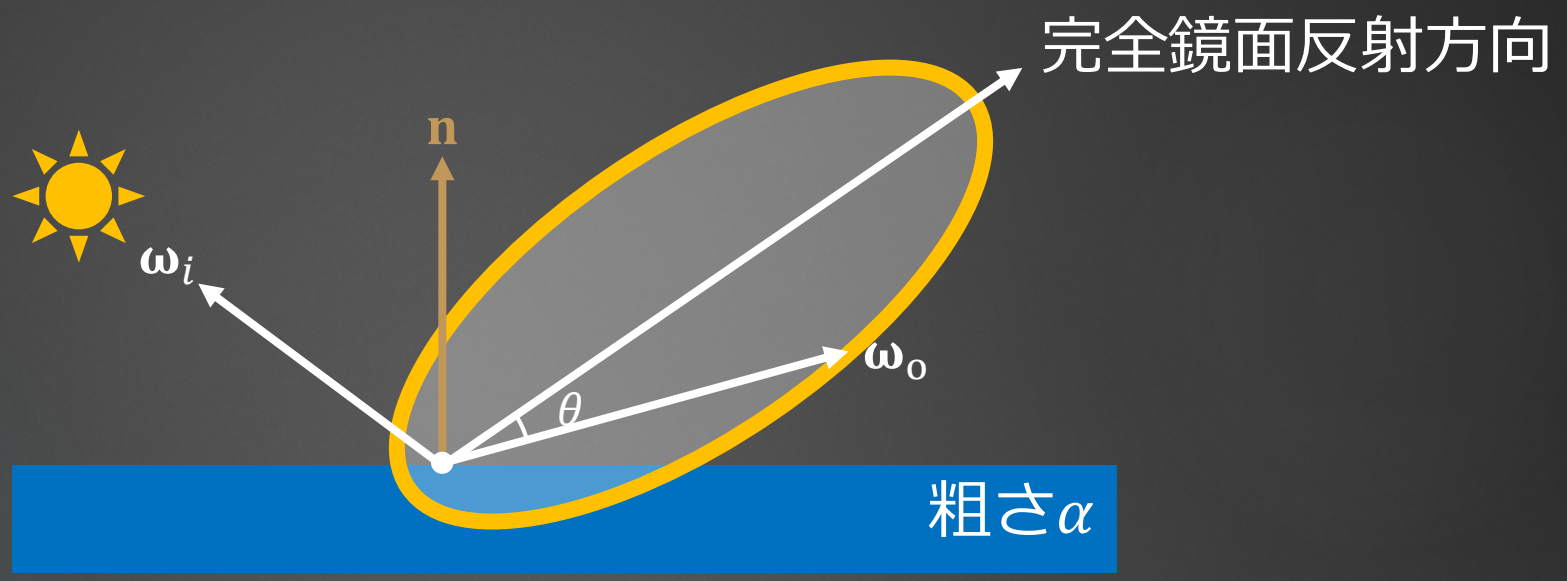
$$\arccos(\boldsymbol{\omega}_h \cdot \mathbf{n}) \geq \frac{\theta}{2}$$

GGXは角度に対して単調減少

$$D(\boldsymbol{\omega}_h \cdot \mathbf{n}) \leq D\left(\cos\frac{\theta}{2}\right)$$

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

反射方向だけに依存するBounding Volume

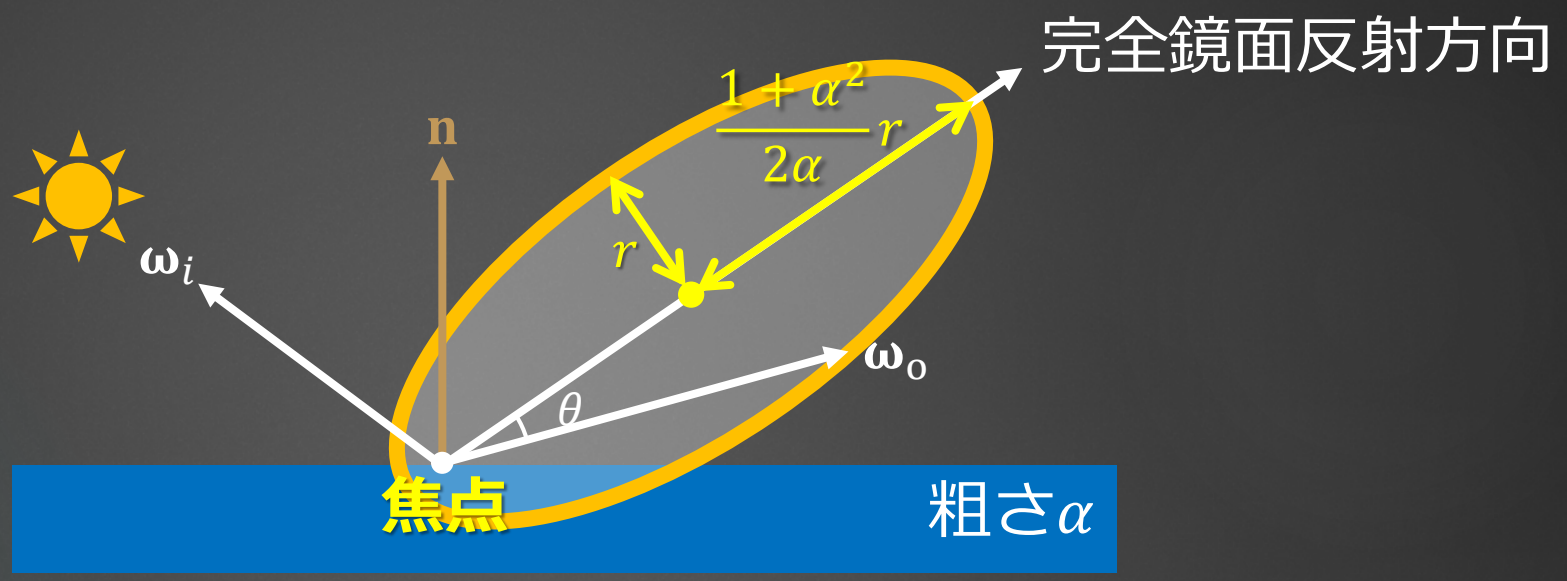


$$l_{\max}(\boldsymbol{\omega}_o) \leq \sqrt{\boxed{\frac{\Phi G_{\max}(\boldsymbol{\omega}_i) F_{\max}(\boldsymbol{\omega}_i)}{4\delta\xi|\boldsymbol{\omega}_i \cdot \mathbf{n}|}} D\left(\cos\frac{\theta}{2}\right)} = r\sqrt{\pi D\left(\cos\frac{\theta}{2}\right)}$$

$\boldsymbol{\omega}_o$に非依存

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

反射方向だけに依存するBounding Volume



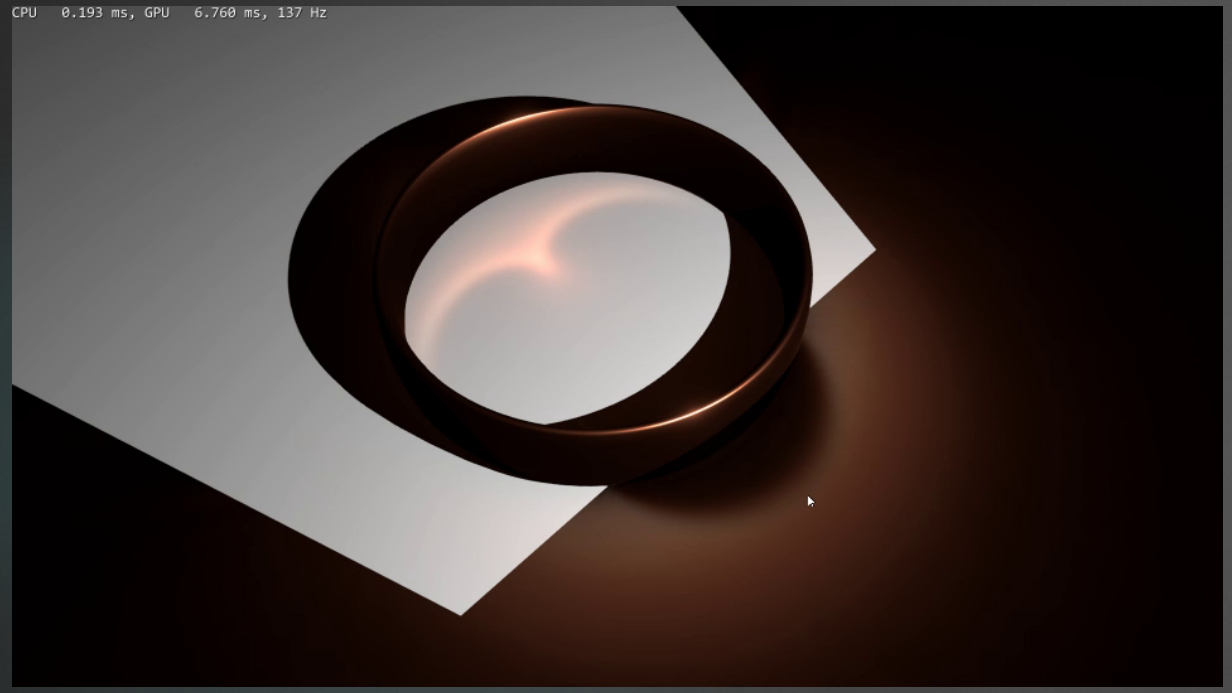
$$l_{\max}(\boldsymbol{\omega}_o) \leq \sqrt{\boxed{\frac{\Phi G_{\max}(\boldsymbol{\omega}_i) F_{\max}(\boldsymbol{\omega}_i)}{4\delta\xi|\boldsymbol{\omega}_i \cdot \mathbf{n}|}} D\left(\cos\frac{\theta}{2}\right)} = r\sqrt{\pi D\left(\cos\frac{\theta}{2}\right)}$$

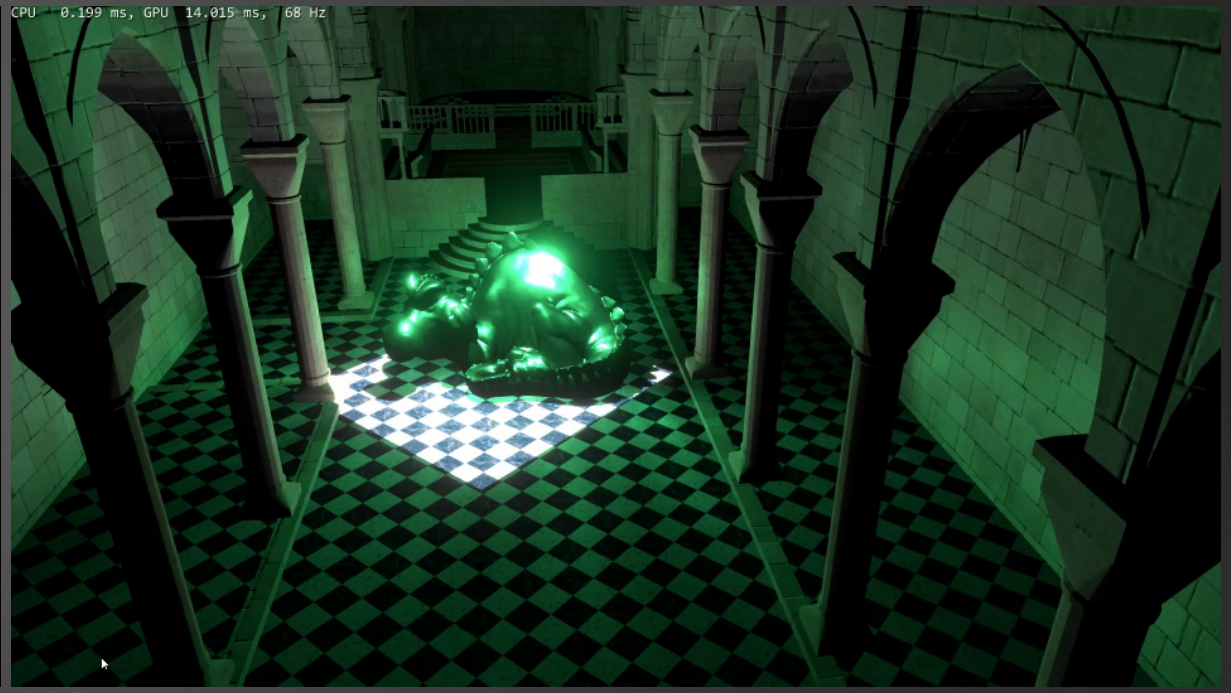
$\boldsymbol{\omega}_o$に非依存

回転楕円体

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

結果

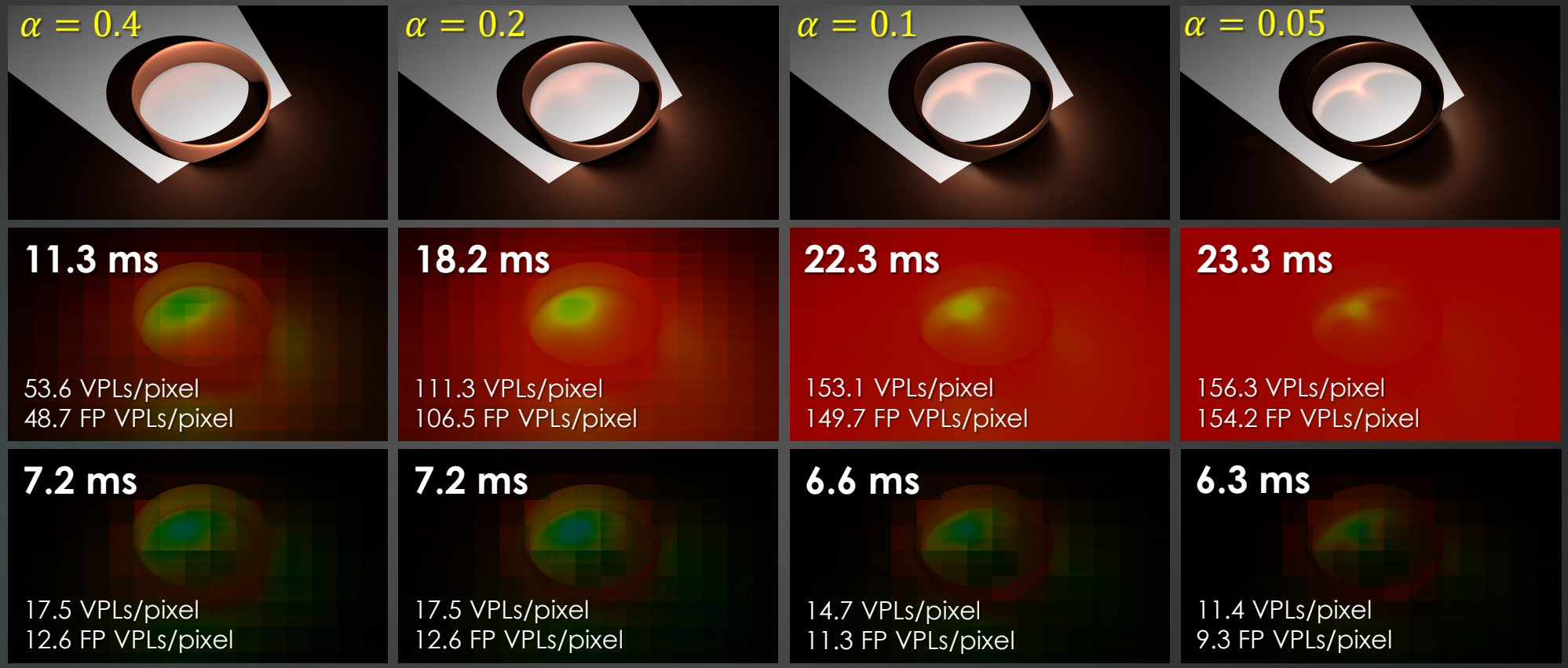




解像度: 1920×1080
65536 VPLs
GPU: NVIDIA® GeForce® GTX 970

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

False Positives (カリングできなかった光源)



明るさ: カリング後のVPL数, スペクトル: false positives, 各時間: 全レンダリング時間（コースティクス以外を含む）

解像度: 1920×1080, 65536 VPLs, GPU: NVIDIA® GeForce® GTX 970

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

GPU実装

GPU IMPLEMENTATION

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

Virtual Point Lights (VPLs)の生成

- Reflective shadow map [Dachsbacher05]を描画
- 各テクセル = VPL
- 256×256 texels = 65536 VPLs

深度　法線　拡散反射率　鏡面反射率　面の粗さ

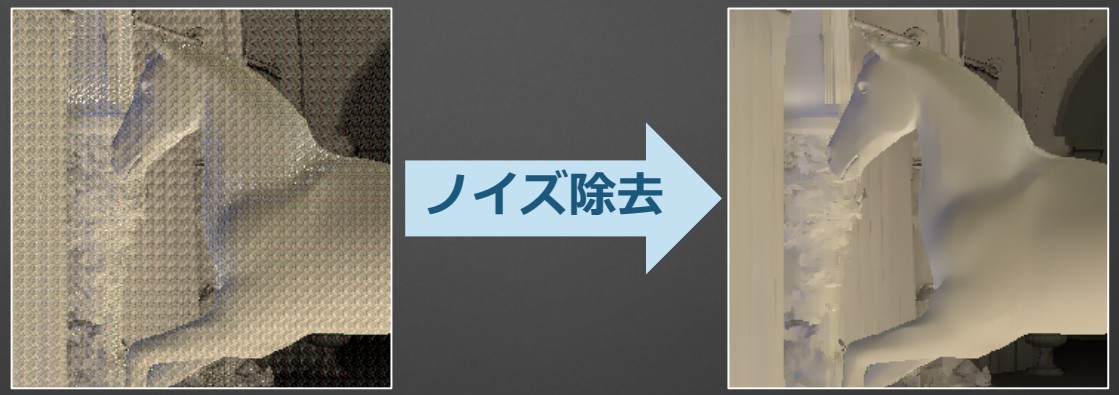
VPLの位置

VPLの放射強度の計算に使用

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

Interleaved Sampling

- ピクセル毎に異なるVPLを使ってシェーディング [Wald02]
  - 8×8ピクセルのサンプリングパターン → ピクセルあたりのVPL数は1/64に
- 誤差（分散）は高周波ノイズとして画面に現れる
- 後処理でノイズ除去フィルタリング

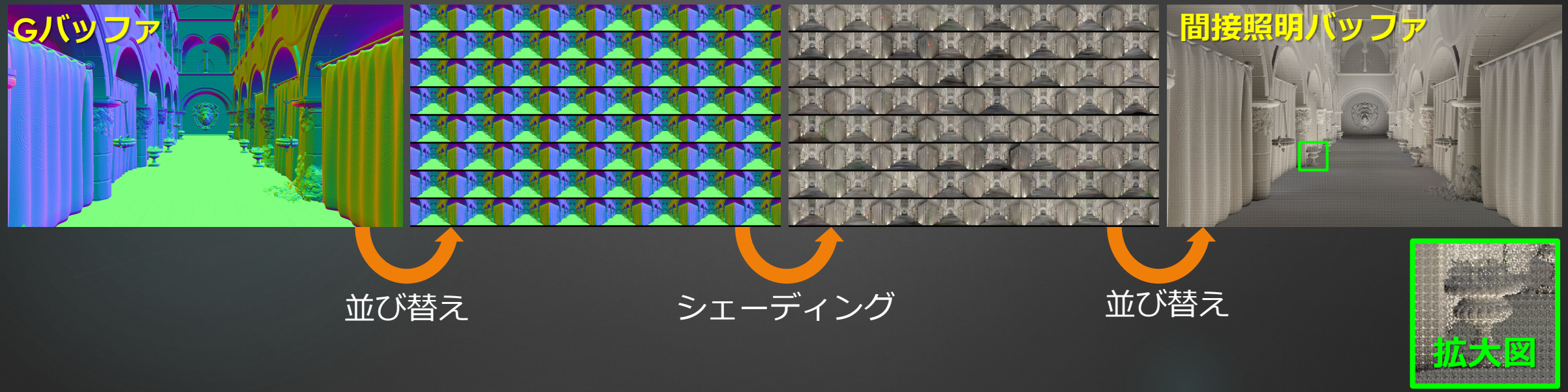


8×8 interleaved sampling

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

Interleaved SamplingのGPU実装

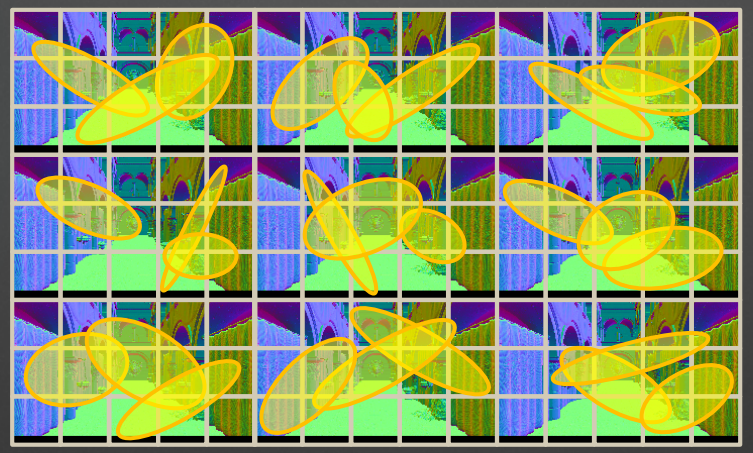
- ピクセルの並び順でスレッドを割り当てるとdivergenceが発生☹
- 画面を8×8の部分領域にピクセルを並び替えて回避 [Segovia06]
  - 部分領域内では同じ光源が使用される☺



© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

ライトカリングを適用

- Interleaved samplingと確率的ライトカリングの組合わせ
- 並び替えた部分領域毎に楕円体のカリングを実行
  - 部分領域あたりのVPL数は1/64（65536個 → 1024個）に減っている
  - さらに確率的にカリングしてピクセルあたりのVPL数を減らす



© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

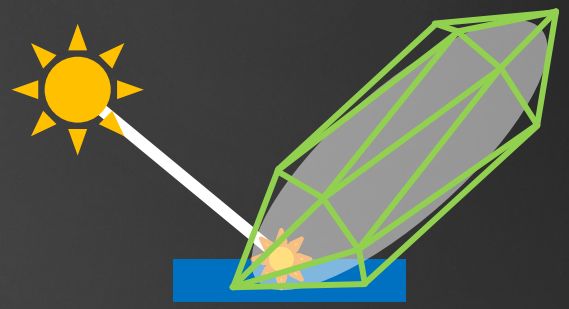
楕円体のカリング

CULLING METHODS FOR ELLIPSOIDS

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

Splatting [Dachsbacher06]

- Bounding volumeを囲むポリゴンをラスタライズ
- Pixel shaderで遅延シェーディング



☺ 任意の凸形状bounding volumeを扱える

☹ 最近のカリング手法と比べると遅い

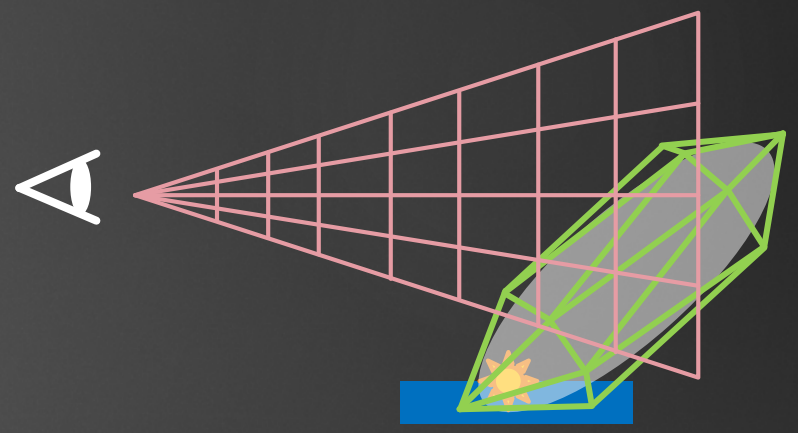
☹ Interleaved samplingによる高速化はpixel shaderのみ

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

Conservative Rasterization-based Clustered Shading

[Ortegren16]

- 各cluster (= froxel)と光の範囲との交差判定にconservative rasterizationを利用
  - Bounding volumeを囲むポリゴンをラスタライズ



☺ 任意の凸形状bounding volumeを扱える

☺ Splattingより高速

☹ Interleaved samplingによる高速化はpixel shaderのみ

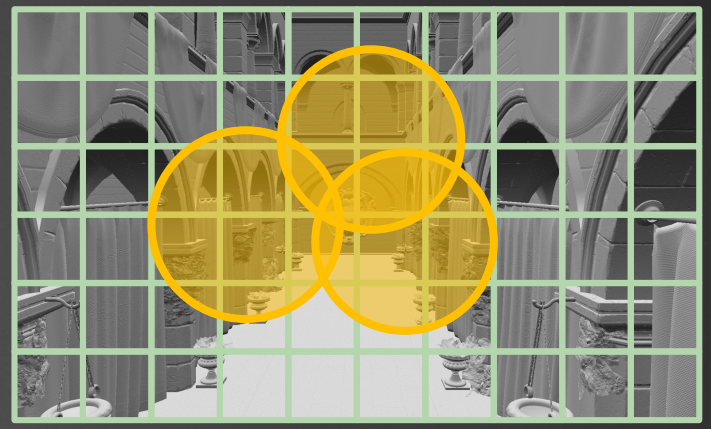
☹ VPLsではメモリー使用量大（最悪の場合: froxel数×VPL数）

  - Interleaved samplingとの組合わせだとfroxel数は多めになる

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

Tiled Deferred Culling

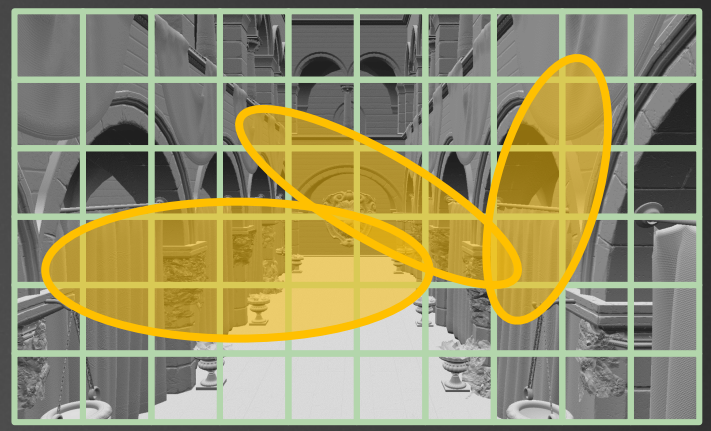
- ☺ 追加のグローバルメモリー不要
- ☺ Interleaved samplingによる高速化大
- ☹ 楕円体に対応してない



© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

Tiled Deferred Culling

- ☺ 追加のグローバルメモリー不要
- ☺ Interleaved samplingによる高速化大
- ~~☹ 楕円体に対応してない~~

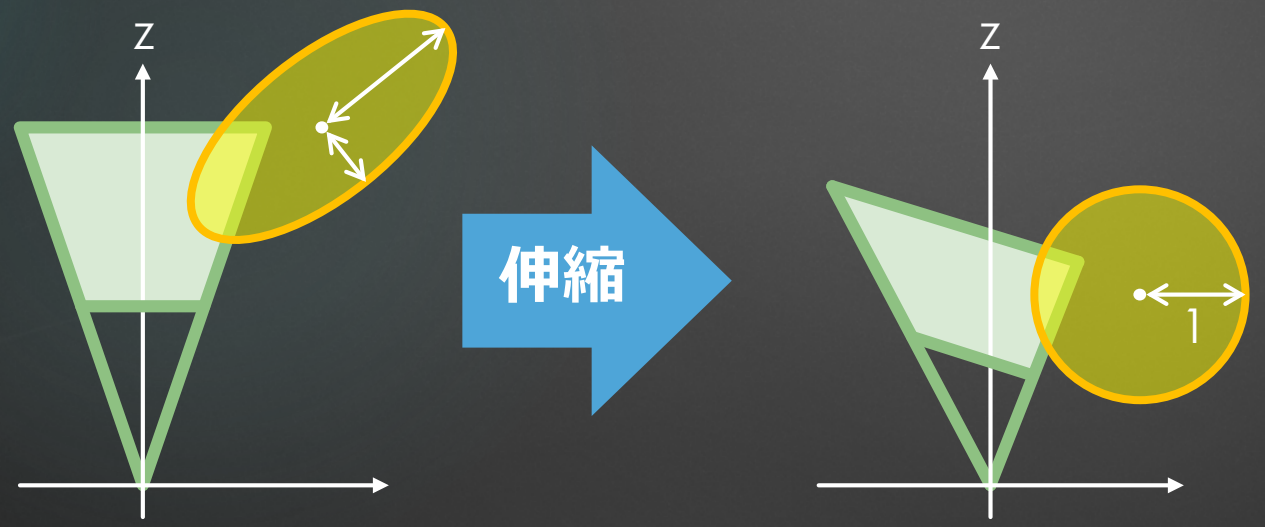


カリング対象を楕円体に拡張

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

楕円体とFrustumとの交差判定

- タイル毎のfrustumと各光源の境界楕円体との交差判定
- 楕円を伸縮して単位球に変換。Frustumも同じように伸縮
- 球とfrustumとの交差判定になる

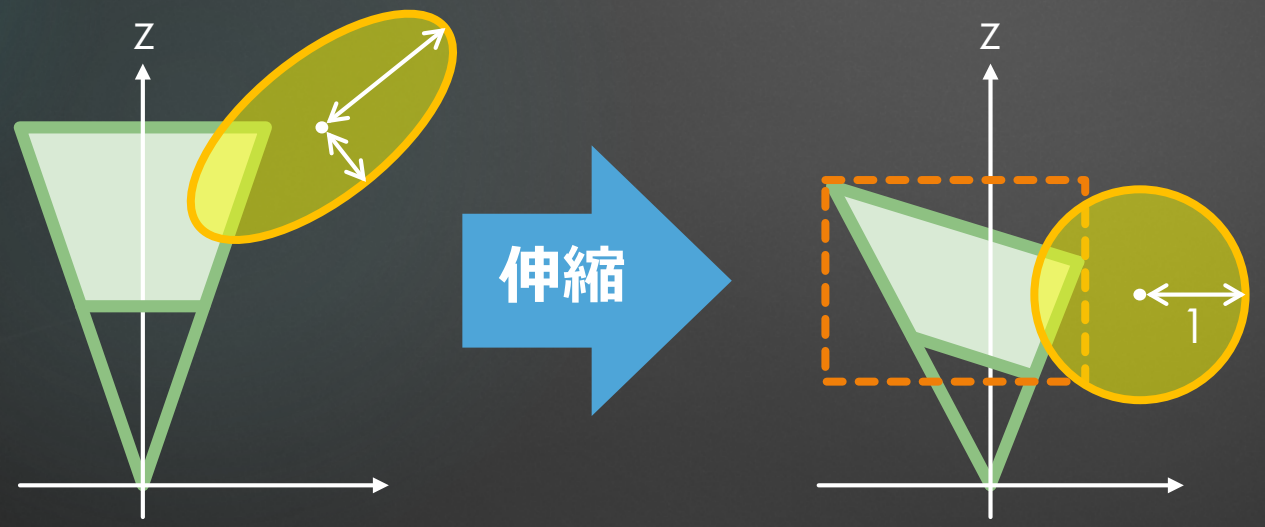


変換行列と単位球の位置は
カリング処理の前にVPL毎に計算しておく

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

粗い交差判定を使った高速化

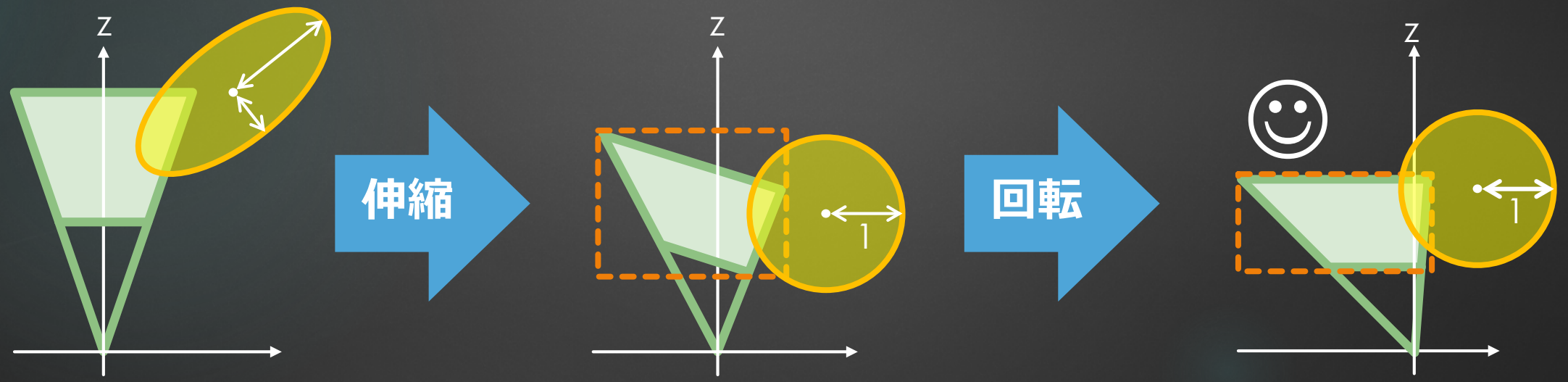
- カリングでは粗い交差判定で良い
- Frustumの代わりにAABB（軸並行境界ボックス）を用いる
- 球とAABBとの高速な交差判定[Arvo90b]が使える



© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

最適化

- 伸縮によって傾いたfrustumだとAABBに隙間が多い ☹
- Frustumの底面がz軸に垂直ではない → AABBの計算量が比較的大きい ☹
  - 既存の高速なカリング実装の多くは垂直を仮定
- 解決方法: **低面がz軸と垂直になるように事前に判定空間を回転しておく**



© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

結果（光沢反射VPLのみ）

Splatting [Dachsbacher06]	10.12 ms	8.91 ms	17.67 ms	5.93 ms	58.81 ms	14.68 ms	13.02 ms
Clustered shading [Ortegren16]	5.92 ms	4.99 ms	8.13 ms	3.92 ms	15.66 ms	10.99 ms	5.40 ms
本手法 (compute-based tiled)	**3.73 ms**	**2.87 ms**	**3.94 ms**	**2.15 ms**	**7.28 ms**	**5.14 ms**	**2.47 ms**

解像度: 1920×1080, 65536 VPLs, GPU: NVIDIA® GeForce® GTX 970

Interleaved samplingと組み合わせる場合は
rasterization-basedよりcompute-basedの方が効率的

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

Tips

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

サンプルコード

- 論文の補足資料にHLSLコードが載っています
  - http://jp.square-enix.com/tech/publications.html
- Modified Half-Z Tiled Culling [Stewart15]をベースに実装
  - 各tile内をふたつの深度領域に分けて堅牢性を高めたもの
  - https://github.com/wolfgangfengel/GPU-Pro-6/tree/master/06_Compute/01_Compute-Based%20Tiled%20Culling
- この実装には改善の余地がある

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

伸縮・回転の簡単化

- 伸縮 ＝ 回転→拡大縮小→ 元に戻す回転
- Frustum低面がz軸と垂直になるように結局回転するので元に戻す回転は不要

$$\text{伸縮行列: } \mathbf{S} = \cancel{\acute{\mathbf{R}}} \begin{bmatrix} 1/r_u & 0 & 0 \\ 0 & 1/r_v & 0 \\ 0 & 0 & 1/r_w \end{bmatrix} \acute{\mathbf{R}}^{\mathrm{T}}$$

Listing 1: *Our bounding ellipsoid generation.*

```
struct ELLIPSOID
{
  float3 transformedCenter;
  float3x3 transformation;
};

float3x3 CalculateStretchingMatrix( const float3 eu, const float3 ev, const float3 ew, const float u, const float v, const float w )
{
  return mul( transpose( float3x3( eu * u, ev * v, ew * w ) ), float3x3( eu, ev, ew ) );
}

void GenerateBoundingEllipsoid( const uint lightIndex, const float r, const float3 position,
                                const float3 direction, const float3 normal, const float roughness )
{
  const float squaredRoughness = roughness * roughness;
```

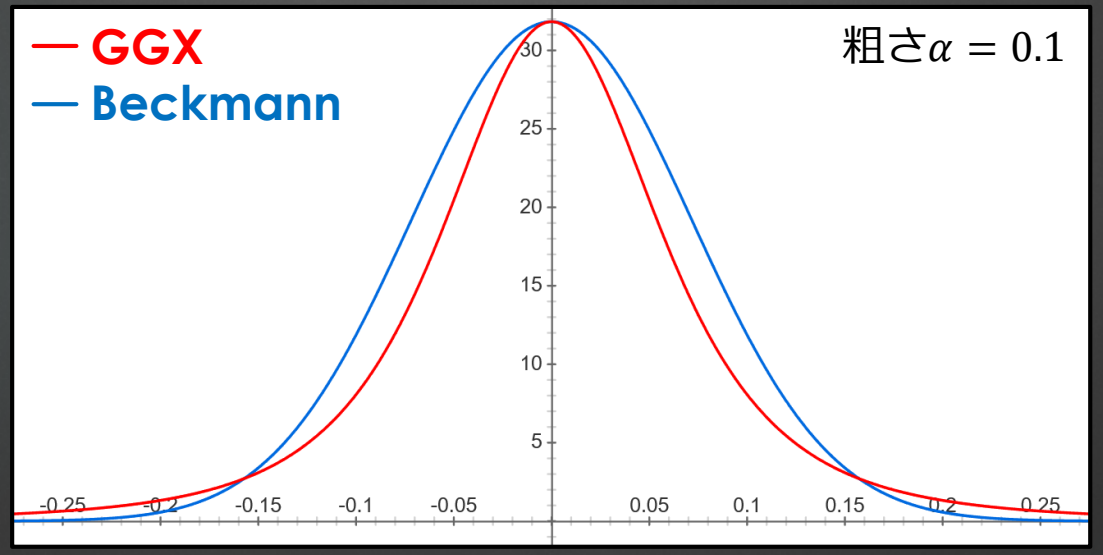
mul()とtranspose()は削除可能

論文の補足資料に載っているサンプルコード[Tokuyoshi17]

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

GGX以外のBRDF

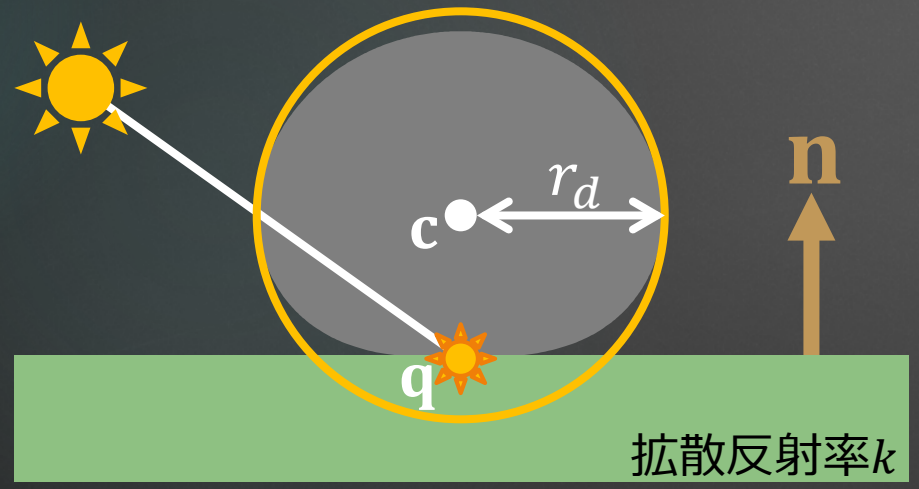
- ロシアンルーレットの確率に用いる反射ローブは近似モデルでも良い
- GGX以外でもGGX近似した確率を使用（非積分関数の方は近似しない）
- 分散は増えるがunbiased



© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

拡散反射VPLs

- Lambert反射でも境界楕円体は解析的に求まるが、ほとんど球状 [Dachsbacher06]
- 楕円体を囲む境界球を使った方が単純で効率的
- 境界球の中心$\mathbf{c}$はVPLの位置$\mathbf{q}$より上になる



境界球の中心：$\mathbf{c} = \mathbf{q} + \left(\frac{1}{27}\right)^{\frac{1}{4}} \sqrt{\frac{\Phi k}{\pi \delta \xi}}\,\mathbf{n}$

境界球の半径：$r_d = \left(\frac{4}{27}\right)^{\frac{1}{4}} \sqrt{\frac{\Phi k}{\pi \delta \xi}}$

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

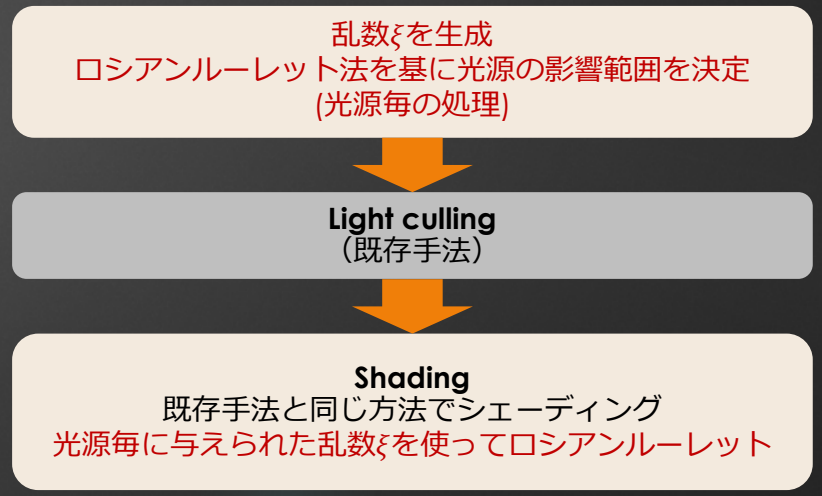
まとめ

CONCLUSIONS

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

確率的ライトカリングとは

- 誤差を小さく抑えることができるカリング手法
- 光の影響範囲をランダムに決定
- 誤差をBiasから分散に変えて、誤差の累積を防ぐ
- 既存のカリング実装自体は変更不要



© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

光沢反射VPLsへの拡張

- 反射光の境界楕円体を使って効率化
  - $D(\cos\theta)$がGGX分布のとき、$\sqrt{D\left(\cos\frac{\theta}{2}\right)}$は回転楕円体となる性質を利用
- 楕円体に対応したカリングはrasterization-basedの手法がある
  - Interleaved samplingとの相性は悪い
- Compute-based tiled cullingの対象を楕円体に拡張して高速化

動的な光沢反射コースティクスをリアルタイムレンダリング

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

制限事項

- シェーディング点側のBRDFは依然として無視
  - 分散大☹
- 分散がバンディングアーティファクトとして画面に現れる
  - 光源に対してひとつしか乱数を用いていないため
- 完全鏡面反射は不可能
- 屈折と異方性反射（今後の課題）

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

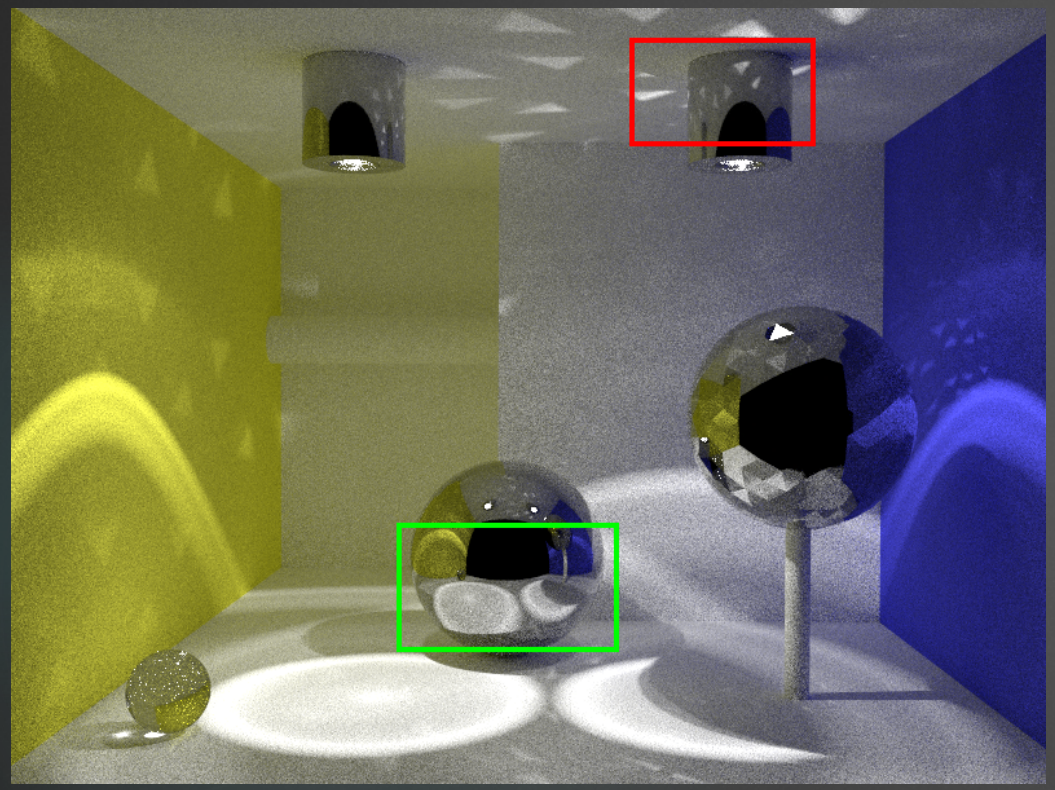
制限事項

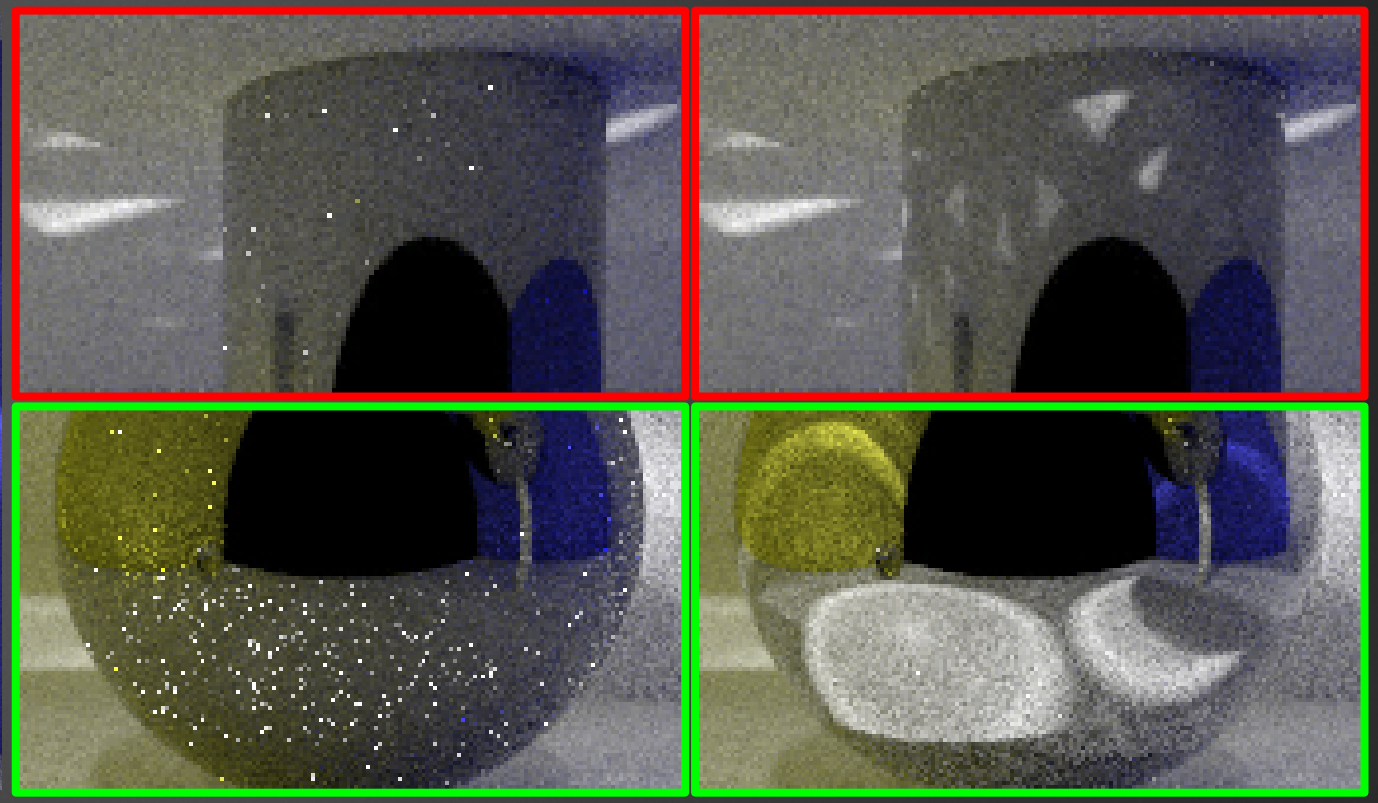
- シェーディング点側のBRDFは依然として無視
  - 分散大☹
- 分散がバンディングアーティファクトとして画面に現れる
  - 光源に対してひとつしか乱数を用いていないため
- 完全鏡面反射は不可能
- 屈折と異方性反射（今後の課題）

オフラインレンダリングについては解決可能

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

双方向パストレーシングへの拡張





Bidirectional Path Tracing Using Backward Stochastic Light Culling

Y. Tokuyoshi and T. Harada. SIGGRAPH 2018 Talks

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

参考文献

- Arvo, J. 1990. A simple method for box-sphere intersection testing. In Graphics gems, Academic Press Professional, Inc., 335-339.
- ARVO J., AND KIRK D. 1990. Particle transport and image synthesis. SIGGRAPH Comput. Graph. 24, 4, 63–66
- BECKMANN P., AND SPIZZICHINO A. 1963. Scattering of Electromagnetic Waves from Rough Surfaces. MacMillan
- COOK R. L. AND TORRANCE K. E. 1982. A reflectance model for computer graphics. ACM Trans. Graph. 1, 1, 7–24
- DACHSBACHER C., AND STAMMINGER M. 2005. Reflective shadow maps. In I3D '05, 203–231
- DACHSBACHER C. AND STAMMINGER M. 2006, Splatting indirect illumination. In I3D'06, pp. 93–100
- HARADA T., MCKEE J., AND YANG J. C. 2012. Forward+: Bringing deferred lighting to the next level. In Eurographics '12 Short Papers
- Heitz, E., and D'Eon, E. 2014. Importance sampling microfacet-based BSDFs using the distribution of visible normals. Comput. Graph. Forum 33, 103-112.
- JAKOB W., HAŠAN M., YAN L.-Q., LAWRENCE J., RAMAMOORTHI R., MARSCHNER S. 2014 Discrete stochastic microfacet models. ACM Trans. Graph. 33, 4, 115:1–115:10
- KELLER A. 1997. Instant radiosity. In SIGGRAPH'97, pp. 49–56
- OLSSON O. AND ASSARSSON U. 2011. Tiled shading. J. Graph. GPU, and Game Tools, 235–251
- OLSSON O., BILLETER M., ASSARSSON U. 2012. Clustered deferred and forward shading. In HPG '12, pp. 87–96
- ÖRTEGREN K., PERSSON E. 2016. Clustered shading: Assigning lights using conservative rasterization in DirectX 12. In GPU Pro 7: Advanced Rendering Techniques. A K Peters/CRC Press, pp. 43–81
- PHONG B. T. 1975. Illumination for computer generated pictures. Commun. ACM 18, 6, 311–317
- SEGOVIA B., IEHL J.-C., MITANCHEY R., PÉROCHE B. 2006. Non-interleaved deferred shading of interleaved sample patterns. In Graph. Hardw. '06, pp. 53–60
- SMITH B. G.: Geometrical shadowing of a random rough surface. IEEE Trans. Antennas and Propagation 15, 5 (1967), 668–671
- STEWART J.: Compute-based tiled culling. In GPU Pro 6: Advanced Rendering Techniques. A K Peters/CRC Press, 2015, pp. 435–458
- WALD I., KOLLIG T., BENTHIN C., KELLER A., SLUSALLEK P. 2002. Interactive global illumination using fast ray tracing. In EGWR'02, pp. 15–24. 2
- Walter B., Dong Z., Marschner S. and Greenberg D. P. 2014. The Ellipsoid Normal Distribution Function.
- WALTER B., MARSCHNER S. R., LI H., TORRANCE K. E. 2007. Microfacet models for refraction through rough surfaces. In EGSR'07, pp. 195–206

"NVIDIA® GeForce®" is a trademark or registered trademark of NVIDIA Corpration.
"Radeon™" is a trademark or registered trademark of Advanced Micro Devices., Inc.