



空からの光

NVIDIA

- 最も簡単なアンビエントオクルージョン (AO)
- 光源 = 空 (球体光源)
- AO - 二つの定義
 - AO = 空の照明による拡散光 [Landis 02] [Christensen 03]
 - AO = 空の照明による影 (sky illumination) [Pharr and Green 04] [Hegeman et al. 06]
- ただ、屋外のシーンに限られている

シーン

球体光源

アンビエントオクルージョン

NVIDIA

- 光源 = ローカルな半球
 - 現在の面上の点を中心に
 - 半径 = ユーザー設定
- レイトレーシングにより、描画できる [Gelato] [Mental Ray]

シーン

ローカルな球体光源

アンビエントオクルージョン

NVIDIA

- 深さ、曲面性、距離感をモデルに与えるのが特徴

AOなし

AOあり

スクリーン空間上のアンビエントオクルージョン

NVIDIA

- 下記の報告にはじめて紹介された [Shanmugam and Orkan 07] [Mittring 07] [Fox and Compton 08]
- 入力 = Z-バッファ + 法線
 - ダイナミックなシーンを対象に、前処理なしでAOの近似値が描画できる
- Z-Buffer = ハイトフィールド
 - $z = f(x, y)$

目

イメージ平面

Z-バッファ

地平のマッピング [Max 86]

NVIDIA

- 1次元のハイトフィールドだと

Z

地平角度

P

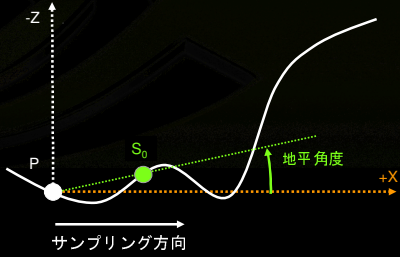
+X

サンプリング (標準化) の方向

地平を検出する



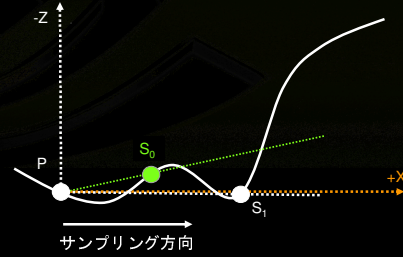
- ハイトフィールドのマーチング



地平を検出する



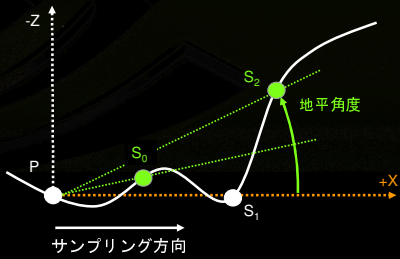
- ハイトフィールドのマーチング



地平を検出する



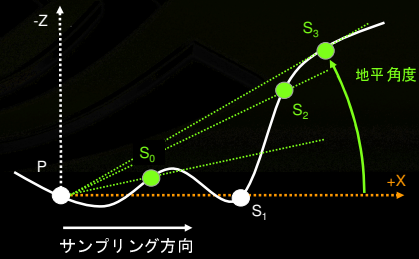
- ハイトフィールドのマーチング



地平を検出する



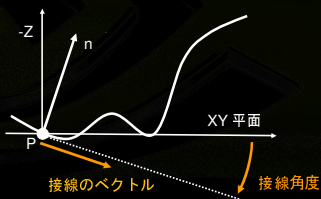
- ハイトフィールドのマーチング



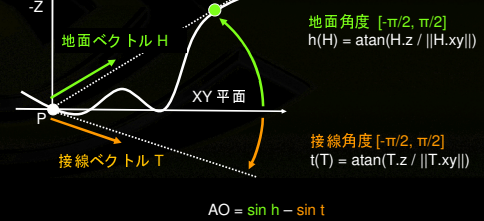
接線の平面



- 点 (P) と P の法線 (n) だと



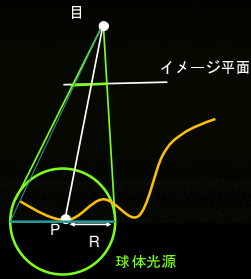
地面ベースのアンビエントオクルージョン



アンビエントオクルージョンの半径



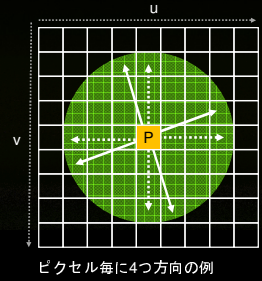
- 目の空間上で定義されたアンビエントオクルージョンの半径
 - シーン = 深さのイメージ
- 光源の球体をテクスチャー空間へと投射する
 - 球体の投射を円で近似化する
 - 円をUV空間に投射する



深さイメージのサンプリング



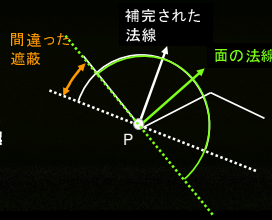
- ピクセル毎の方向が正常分布になっているとする
 - 方向毎のサンプルを固定
- ピクセル毎のランダム化
 - ピクセル毎、方向をランダムに回転させる
 - ランダムなオフセットでサンプルをジッターさせる (ランダム化)



法線



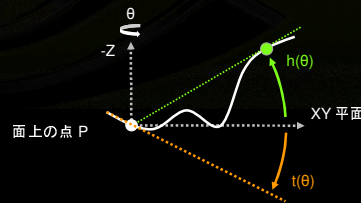
- ピクセル単位で法線を保存する
 - 補完された法線ではない
 - 間違った遮蔽が発生してしまう
 - 面の法線になる
 - ジオメトリのパスで目空間上の座標に対して、ddx/ddy命令を利用する



コアなアルゴリズム



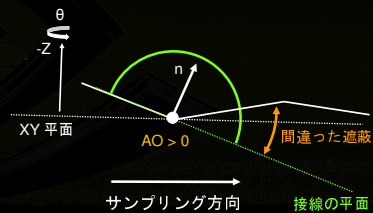
- 2次元上、AOの積分を取る
 - 複数の2次元方向でAOの平均値を取る
 - $AO(\theta) = \sin h(\theta) \cdot \sin t(\theta)$



折り目に見られるアンビエントオクルージョン



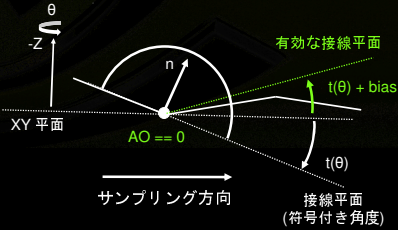
低三角化による問題



解: 角度のバイアス



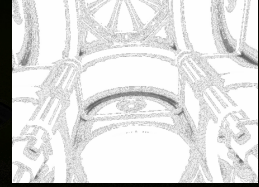
- Mental Rayでの"Spread"パラメータに似ている
- 接線平面周辺の遮蔽を無視する



角度バイアスの実例



角度バイアスなし



角度バイアスあり = 30度

スクリーン外のサンプリング



- 視錐台外では、シーン情報がない
- 辺固定と角度バイアスを用いて間違った遮蔽を外した

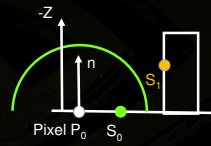


角度バイアス = 0

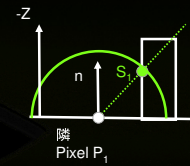


角度バイアス = 30度

非連続性の問題



$$AO(P_0) = \sin h - \sin t = \sin 0 - \sin 0 = 0$$



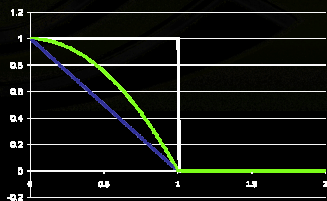
$$AO(P_1) = \sin h - \sin t = \sin(45\text{deg}) - \sin 0 = 0.7$$

→ P₀とP₁の間、非連続が発生する

減衰の関数



- 放射関数W(r)により、AOのウェイトを付ける
- "obscurances"に似ているアプローチ [Zhukov et al. 98]
- [Gelato] と[Mental Ray]では、"Falloff"のパラメータ



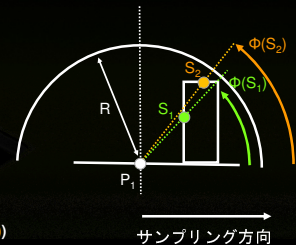
基準化した距離
 $r = \|S - P\| / R$

減衰関数
 $W(r) = 1 - r^2$

サンプル単位の減衰



- WAO = 0と、初期化
- サンプルS₁の後
 - $AO(S_1) = \sin \Phi(S_1) - \sin t$
 - $WAO += W(S_1) AO(S_1)$
- サンプルS₂の後
 - もし、 $\Phi(S_2) > \Phi(S_1)$
 - $AO(S_2) = \sin \Phi(S_2) - \sin t$
 - $WAO += W(S_2) (AO(S_2) - AO(S_1))$



減衰ありとなしの例



減衰あり
 $W(r) = 1 - r^2$

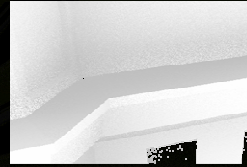


減衰なし
 $W(r) = 1$

ノイズ



- ・ピクセル毎のランダム化によるノイズ



AO: 6 方向x 方向毎に6ステップ

Cross Bilateral Filter



- ・AOをぼかす
- ・深さ依存性のガウスのブラ
 - ・ [Petschnigg et al. 04]
 - ・ [Eisemann and Durand 04]
 - ・ 辺のぼかしが防止できる
- ・ 厳密には、非分離型のフィルターだが、X方向とY方向にそれぞれ与える

Cross Bilateral Filter



- ・深さ依存のぼかし



ぼかしフィルターなし



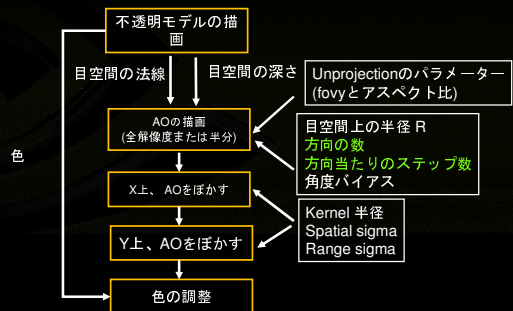
15x15のぼかしフィルター

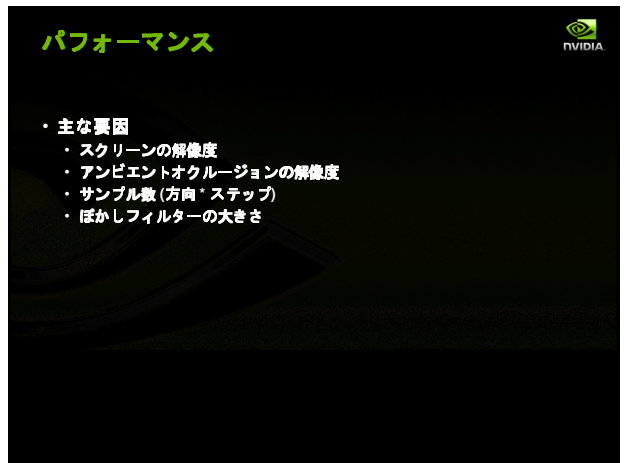
半解像度のAO



- ・ AOはほとんど低周波数
 - ・ 半解像度の描画も可能
 - ・ 半解像度の深さイメージを読み込む
- ・ ぼかし処理は、まだ全解像度を対象にする
 - ・ 辺での色滲みを防止する
 - ・ 目上の深さを全解像度で読み込む
 - ・ [Kopf et al. 07]

描画パイプライン





半解像度のAO

イメージサイズ 1600x1200
 AO解像度 800x600
 ぼかし解像度 1600x1200

半解像度 AO	GeForce GTX 280
ジオメトリ	1.0 ms
AO	3.5 ms
ぼかし	2.5 ms
合計	7.0 ms
	143 fps

6 directions per pixel
 6 steps per direction
 15x15 Blur Size

全解像度のAO

イメージサイズ 1600x1200
 AO 解像度 1600x1200
 Blur 解像度 1600x1200

Full-Res AO	GeForce GTX 280
Geometry	1.0 ms
AO	30.0 ms
Blur	2.5 ms
Total	33.5 ms
	30 fps

6 方向/ピクセル
 6 ステップ/方向
 15x15 ぼかしのサイズ

半解像度のAO

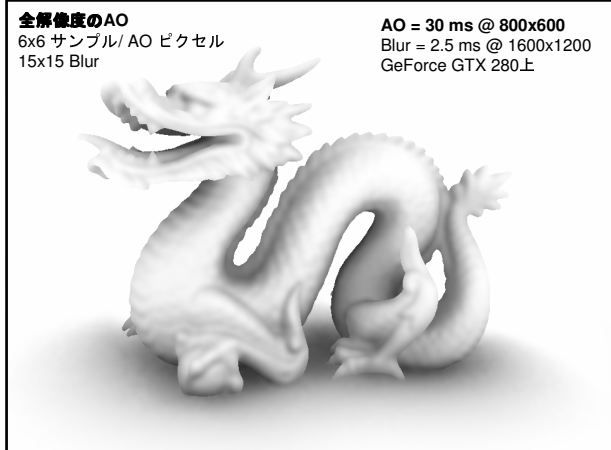
6x6 サンプル/ AOピクセル
 Blurなし

AO = 3.5 ms @ 800x600
 GeForce GTX 280上

半解像度のAO

6x6 サンプル/ AOピクセル
 15x15 Blur

AO = 3.5 ms @ 800x600
Blur = 2.5 ms @ 1600x1200
 GeForce GTX 280上



結論

GeForce 8
DIRECTX 10
Video
Whitepaper
Download

- DirectX10 SDK のサンプル
 - developer.nvidia.comにて配信中
 - ビデオと技術白書も
- ゲームエンジンに組み込みやすい！
 - 入力 = 目空間上の深さと法線
 - 後処理のパスで描画する
- 更に詳しくは、ShaderX7 (掲載予定)

Acknowledgments

- NVIDIA
 - Rouslan Dimitrov, Samuel Gateau, Michael Thompson, Ignacio Castano, the demo team
- Models
 - Dragon - Stanford 3D Scanning Repository
 - Sibenik Cathedral - Marko Dabrovic

Q & A

コードのサンプルの以下のアドレスまで
<http://developer.nvidia.com>

bdudash@nvidia.com
tkazama@nvidia.com

**ありがとう
ございました!**

10:40 - 12:00
Displacement Subdivision Surfaces in DX11
 Takayuki Kazama

13:00 - 14:20
GPU Physics and CUDA
 Kitty Vongsay

14:50 - 16:10
Real time Hair Rendering
 Bryan Dudash

16:40 - 18:00
Horizon Based Ambient Occlusion
 Bryan Dudash

developer.nvidia.com

References

- [Max 86] MAX, N. L. 1986. "Horizon mapping: Shadows for bump-mapped surfaces." In Proceedings of Computer Graphics Tokyo '86 on Advanced Computer Graphics.
- [Zhukov et al. 98] Sergej Zhukov, Andrej Inoes, Grigorij Kronin, 1998. "An ambient light illumination model." In Rendering Techniques '98, G. Drettakis and N. Max, Eds., Eurographics, 45-56.
- [Landis 02] Landis, 2002. "Production-Ready Global Illumination," In ACM SIGGRAPH Course #16.
- [Christensen 03] Christensen, P. H. 2003. "Global illumination and all that". In ACM SIGGRAPH Course 9.

References

- [Eisemann and Durand 04] Elmar Eisemann and Frédo Durand, "Flash Photography Enhancement via Intrinsic Relighting", ACM SIGGRAPH 2004.
- [Petschnigg et al. 04] Petschnigg, Szeliski, Agrawala, Cohen, Hoppe, Toyama, "Digital photography with flash and no-flash image pairs." ACM SIGGRAPH 2004.
- [Schüler 05] Christian Schüler, 2005. "Eliminate surface acne with gradient shadow mapping", ShaderX4: Advanced Rendering Techniques.
- [Gelato] Gritz, L. 2006. "Gelato 2.1 Technical Reference". NVIDIA.
- [Mental Ray] "mental ray Shader Reference", November 2007.

References

- [Mittring 07] MITTRING, M. 2007. "Finding next gen: Cry Engine 2". In SIGGRAPH '07: ACM SIGGRAPH 2007 courses.
- [Shanmugam and Orikan 07] SHANMUGAM, P., AND ARIKAN, O. 2007. "Hardware accelerated ambient occlusion techniques on GPUs". In I3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games.
- [Kopf et al. 07] Johannes Kopf, Michael F. Cohen, Dani Lischinski, Matt Uyttendaele, "Joint Bilateral Upsampling", SIGGRAPH '07: ACM SIGGRAPH 2007 papers.
- [Fox and Compton 08] Megan Fox and Stuart Compton, "Ambient Occlusive Crease Shading", Game Developer Magazine, March 2008.

目空間上の位置を計算する

• 座標 (u,v)にあるサンプルのS だと

— 再構築の (x,y)とサンプルのzのずれ違いを防止するため、uvを移動させる

接線角度

- 接線角度のTでは、
 - 平面上のT(θ)
 - $t(\theta) = \text{atan}(T.z / ||T.xy||)$
- 平面的基底を計算する
 - 基底 = (dP/du, dP/dv)
 - $T = dP/du \Delta u + dP/dv \Delta v$
 - 勾配シャドーマッピング [Schüler 05]