

スクリーン空間における大域照明に関する研究

Yang Wenxiang 齋藤豪
東京工業大学 情報理工学院

1 はじめに

大域照明は、コンピュータグラフィックスの分野で重要な概念であり、仮想の3D空間においてより現実感がある照明効果を実現するための一連の手法を指す。これらの手法は、3D空間内のすべての光源の相互作用を考慮し、光が空間中でどのように伝播し反射するかを模倣する。大域照明の主な目標は、コンピュータがレンダリングする画像のリアリティを向上させ、それを現実世界の照明条件により一致させることである。

本稿では、画像転送ベースの分散レンダリング手法への大域照明の導入を目指し、スクリーン空間の情報を利用して遅延シェーディング (Deferred shading) のパイプラインで大域照明を計算する手法を提案する。さらにレンダリング結果と計測実験の結果を報告する。

2 提案手法

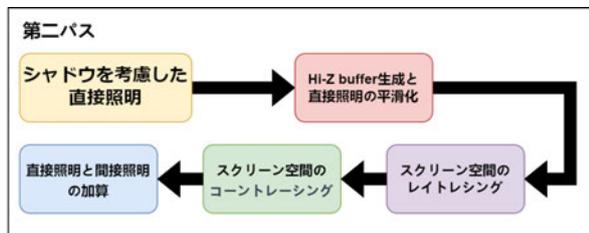


図 1: シェーディング計算のパイプライン

画像転送ベースの分散レンダリング手法への導入を目指しているため、本稿で提案するレンダリング手法は遅延シェーディングのパイプラインを利用している。遅延シェーディング [1] はレンダリングを二つのパスに分ける。第一のパスはジオメトリパスと呼ばれ、このパスでシーンを一度レンダリングして、ジオメトリ情報を G-buffer [2] に保存する。第二のパスはライトニングパスと呼ばれ、G-buffer を利用してスクリーンスペースでシェーディングを行う。そのため、大域照明のシェーディング計算は第二のパスで行い、G-buffer に必要な情報を記録しておく必要がある。本研究の大域照明は Uludag ら [3] によるスクリーン空間のレイト

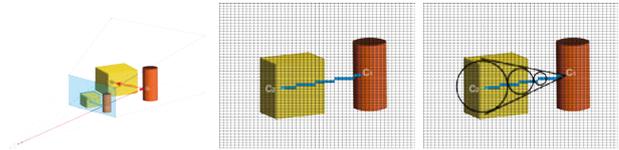


図 2: 間接照明の生成法

アルbedo (RGB 24-bit)	金属度 (8-bit)
法線情報 (XYZ 24-bit)	表面粗さ (8-bit)
観察視点の深度情報 (32-bit)	
シャドウマップ (32-bit)	

図 3: G-buffer の設計

レーシング (Screen-space ray tracing, SSR) を基礎にして、遅延シェーディングの第二パスのシェーディング計算に導入する。Uludag らの SSR は間接光源を探す方法であり、直接照明の計算部分でのシャドウを考慮していない。本研究での直接照明の計算に SSABSS [4] によるシャドウ計算を導入したの第二パスのパイプラインを図 1 に示す。

第二パスでシェーディング計算を行うため、第一パスでは次の要素を含む G-buffer を生成する。物体表面の属性として、アルbedo、法線、素材特性、光源、視点との相対位置を計算するための深度値、そして一次光源からの可視性判定のためのシャドウマップである。素材特性としては、金属度と表面粗さを今回は採用している。各要素は図 3 に示すように G-buffer 内に割当てを行った。

SSABSS を用いた影のある一次光源による反射画像 (直接照明) を計算した後、その画像上で SSR を実行する。SSR では、図 2 左のような三次元での鏡面反射方向へのパストレーシングを行う代わりに、同図中央のようにスクリーン上でトレースを行って光線の衝突位置までの長さを求めた後、同図右のようにそれを高さとする二等辺三角形を用いたコーントレーシングを行う。このときの二等辺三角形の頂点の角は表面粗さにより制御される。SSR によって二次光源による反射 (間接照明) を一次光源による反射画像に加算して最

Research of screen-space global illumination
Wenxiang Yang
Suguru Saito
School of Computing, Tokyo Institute of Technology

終画像を生成する。式で表すと式 (1) となる。

$$\begin{aligned} \text{Buffer}[\text{direct}] &= \text{Reflection}(\text{Buffer}[\text{shadowmap}, \\ &\text{albedo}, \text{normal}, \text{metaric}, \text{roughness}], \text{Light}) \\ \text{Buffer}[\text{indirect}] &= \text{SSR}(\text{Buffer}[\text{albedo}, \text{normal}, \\ &\text{metaric}, \text{roughness}, \text{direct}]) \\ \text{Buffer}[\text{result}] &= \text{Buffer}[\text{direct}] + \text{Buffer}[\text{indirect}] \end{aligned} \quad (1)$$

3 結果と評価



図 4: アルベド



図 5: 単純一次反射計算

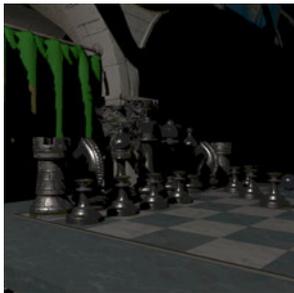


図 6: +影効果追加

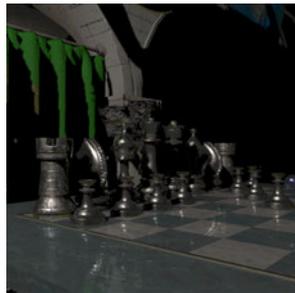


図 7: +影大域照明効果追加
(表面粗さ 0.0)



図 8: +影大域照明効果追加
(表面粗さ 0.3)

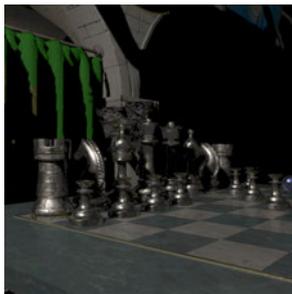


図 9: +影大域照明効果追加
(表面粗さ 0.9)

実験環境として、CPU : Intel Core i9-10900 2.80GHz、GPU : NVIDIA GeForce RTX 2080 Ti、RAM : 32GB を用いて、1024 × 1024 の解像度でレンダリングを行った。レンダリング結果を図 4 から図 9 に示す。

図 7 から図 9 を見ると表面が粗くなるにつれて盤面への映り込みがぼやけてくるのがわかる。また、直接反射の影内でも間接照明の照射が計算されていること

で完全な黒ではなく間接照明の結果が反映されていることがわかる。

1 フレームの生成時間を表 1 に示す。表より、レンダリング処理のうち、影と大域照明効果のための演算は比較的短いことが分かる。なおこれら効果のための演算時間はスクリーンスペースでの演算であることからシーンの複雑性にはほぼよらず、解像度に依存する。

表 1: 1 フレームの生成時間

単純一次反射計算	+影効果追加	+影大域照明効果追加
15.08ms	15.79ms	16.69ms

4 結論と今後の課題

本稿では、分散レンダリング手法 [5] への大域照明の導入を目指した遅延シェーディングによるスクリーンスペースの影と大域照明の演算の実装についての我々の研究を述べた。この大域照明の演算法では全方向の拡散反射のための入射光についての考慮はされていないが、生成画像からは影と大域照明の描画を確認することができた。ただしエイリアシングへの対応が画質の面での今後の課題である。また分散レンダリング手法へ本稿で述べた実装を組み込んだ場合の実験計測を行うことも今後の課題である。

参考文献

- [1] Shawn Hargreaves and Mark Harris. Deferred shading. In *Game Developers Conference*, Vol. 2, p. 31, 2004.
- [2] Takafumi Saito and Tokiichiro Takahashi. Comprehensible rendering of 3-d shapes. *ACM SIGGRAPH Computer Graphics*, Vol. 24, No. 4, pp. 197–206, 1990.
- [3] Yasin Uludag. Hi-z screen-space cone-traced reflections. In *GPU Pro 360 Guide to Lighting*, pp. 237–280. AK Peters/CRC Press, 2018.
- [4] Zhongxiang Zheng and Suguru Saito. Efficient screen space anisotropic blurred soft shadows. *IEICE TRANSACTIONS on Information and Systems*, Vol. 97, No. 8, pp. 2038–2045, 2014.
- [5] Hao Fang, Naohito Okumura, Kakeru Ishii, and Suguru Saito. Distributed rendering on grid computers for multiple users in shared virtual space. In *2022 International Conference on Cyberworlds (CW)*, pp. 47–54, 2022.