

高速プライマリレイ走査のための アフィン変換レイアライメント

西舘 祐樹 藤代 一成
慶應義塾大学 理工学部情報工学科

1 背景と目的

近年、専用ハードウェアの普及に伴い、リアルタイムレンダリングにおいてもレイトレーシングが使われ始めている。しかし、依然としてレイ走査の時間計算量が高く、リアルタイムに出射できるレイの本数は限られている。これまでもレイの本数を削減する手法は数多く提案されているが、それらの手法には、取得できる衝突情報も減少してしまうという本質的なデメリットが存在する。そこで、本稿では情報を損なうことなくレイの本数を削減する手法として、アフィン変換レイアライメント (affine-transformed ray alignment) を提案する。加えて、本手法をプライマリレイ走査に適用する実験を行い、その高速化効果について報告する。

2 概要

本手法では、図1に示すように、アフィン変換を用いて複数本のレイを一本にまとめることでレイの本数を削減する。また、レイの削減と同時にオブジェクトのインスタンスングを行うことで、情報損失を避けながらレイ走査を高速化する。

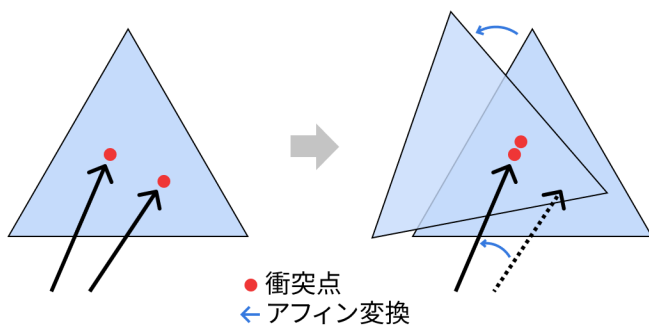


図1: アフィン変換レイアライメントの概要。レイとオブジェクトの両方に同一のアフィン変換を適用することで、レイの本数を減らしながら、変換前と同一の衝突情報を取得する。

3 手法

アフィン変換レイアライメントを導入したあと、プライマリレイ走査への適用について詳説する。

3.1 アフィン変換レイアライメント

アフィン変換レイアライメントは、アフィン変換を用いて複数本のレイを一本にまとめることでレイの本数を削減する手法である。通常はレイ削減に伴って衝突情報も失われてしまうが、本手法では一本のレイから複数本分の衝突情報を取得することで、損失した情報を補填する。その際、アフィン変換前のレイと同じ衝突情報を取得する必要があるため、レイに適用したものと同一のアフィン変換をオブジェクトにも適用し、インスタンス作成しておく。その後、インスタンスを含めた bounding volume hierarchy (BVH) [1] を構築することで、インスタンスングによる時間計算量の増加を対数時間に抑えている。

3.2 プライマリレイへの適用

プライマリレイとは、視点から出射される第一のレイである。全てのレイの方向がレンダリング前に既知であるため、図2に示すように、画面の大部分のレイを単一のアフィン変換で重ねることができる。例として、Y軸に対してレイを反転すると、本数を半減することができる。続いてX軸に対して反転を重ねることでレイの本数をさらに削減できるが、インスタンスの重なりが増えるほどBVHの品質も低下するため、適切な削減数はシーンによって異なる。

3.3 二段階BVH

レイトレーシングでは、レンダリング時間だけでなくBVHの構築時間についても考慮する必要がある。今回の実験では、標準的なBVHと比較して、レンダリング速度が遅い代わりに動的なシーンにおいて高速に更新可能な二段階BVH[2]を実装した。二段階BVHは、オブジェクト単位とシーン全体の二種類の木構造からなる。シーン初期化時にオブジェクト単位で木構造を事前構築し、オブジェクトの移動や回転に合わせてシーン全体の木構造を更新する。また、BVH構築前にインスタンスのカリングを行うことで、さらなる高速化とデータ量の削減を実現する。

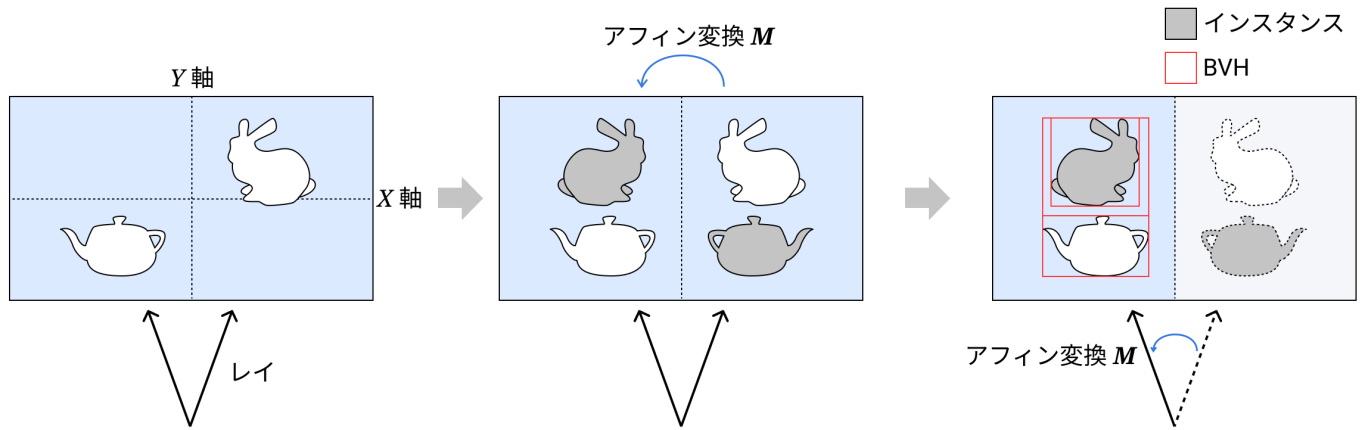


図 2: アフィン変換を用いて複数本のレイを一本にまとめることで、プライマリレイの本数を削減する手法の例。レンダリング前に、オブジェクトを Y 軸に対して反転したインスタンスを作成したのち、BVH を構築しておく。レンダリング時には、画面左側半分にはのみレイを出射し、一本のレイで二本分の衝突情報を取得することで、元のレイの本数と同等の情報が得られる。

表 1: レンダリング時間 (T) とメモリ使用量 (M)

手法 (反転軸)	T	M
従来手法	1.96×10^3 ms	9.2203 MB
提案手法 (Y 軸)	1.42×10^3 ms	9.2221 MB
提案手法 (X 軸)	1.52×10^3 ms	9.2220 MB
提案手法 (Y&X 軸)	1.12×10^3 ms	9.2221 MB

表 2: BVH の事前構築時間と更新時間

手法 (反転軸)	事前構築時間	更新時間
従来手法	7.01×10^3 ms	4.18×10^1 ms
提案手法 (Y 軸)	7.01×10^3 ms	6.12×10^1 ms
提案手法 (X 軸)	7.01×10^3 ms	5.75×10^1 ms
提案手法 (Y&X 軸)	7.01×10^3 ms	8.77×10^1 ms

4 結果

実験環境として、CPU: Intel Core i7-11700 2.50 GHz, RAM: 32.0 GB を用いて、262,267 枚のポリゴンのシーンを、 $1,280 \times 720$ 画素でレンダリングした。また、レイを X 軸反転する手法、Y 軸反転する手法、X 軸と Y 軸の両軸反転する手法の三つを実装した。なお、本手法の効果は軸反転の順序には依存しない。

レイを画面全体に出射する従来手法と、一部に出射する提案手法との間で、レンダリング時間とメモリ使用量について比較した結果を表 1 に示す。レンダリング速度は、レイの本数を削減するほど高速化され、最大で約 1.74 倍まで高速化されることが確認できた。また、メモリ使用量は手法適用前とほぼ変わらず、インスタンスによるデータ量の増加はきわめて小さかった。

BVH の事前構築時間と更新時間を表 2 に示す。事前構築時間については本手法による低速化は一切ないが、更新時間の増加がみられた。しかし、レンダリング時間の短縮効果に比べると十分に小さく、処理全体としては高速化された。

5 結論と今後の課題

本稿では、情報損失のないレイ走査の高速化手法として、アフィン変換レイアライメントを提案した。さらに、プライマリレイ走査への適用実験によって、レンダリングの高

速化が確認され、空間計算量の増加も少なく、BVH の構築時間への影響も小さいことが示された。

アフィン変換レイアライメントは、本稿で紹介した応用以外にも、VR のためのステレオレンダリングや、複数フレームを同時にレンダリングするマルチフレームレンダリングにも応用可能である。

今後の課題として、本手法による高速化の効果を予測し、それによってシーンごとに最適なアフィン変換を推定することが考えられる。

謝辞

本研究の一部は、令和 3 年度科研費挑戦的研究 (開拓) 20K20481 の支援により実施された。

参考文献

- [1] James H. Clark. Hierarchical geometric models for visible surface algorithms. *Communications of the ACM*, Vol. 19, No. 10, pp. 547–554, 1976.
- [2] Ingo Wald, Carsten Benthin, and Philipp Slusallek. Distributed interactive ray tracing of dynamic scenes. In *Proceedings of the 2003 IEEE Symposium on Parallel and Large-Data Visualization and Graphics*, pp. 77–85. IEEE Computer Society, 2003.