

高位合成によるラジオシティ法のソフトウェア /ハードウェア協調システムの開発

学籍番号:1731103 成見研究室 田村昂太郎

1 はじめに

コンピュータグラフィックスにおいて現実感を与える最も重要な要因の1つとして、三次元空間内のオブジェクトの陰影計算がある。ラジオシティ法 [1],[2] は、多大な計算コストと引き換えに光の相互反射を考慮した柔らかな光の映写効果を現実に忠実かつ精巧に描写できる手法である。FPGA (Field Programmable Gate Array) は、再構成可能な論理デバイスであり、計算を高速に行う数値アクセラレータとして使用することも出来る。FPGA の回路設計は従来多大な開発コストを要するものであったが、近年高級言語から回路を生成する高位合成と呼ばれる技術により開発工数が縮小され設計抽象度も向上した。しかし、FPGA で構成される算術演算回路の解析的な最適化は困難であり、高位合成における最適化手法の確立が課題となっている。以上の背景を踏まえ、本稿ではラジオシティ法を適用例として高位合成ツールである SDSoC の最適化手法の評価を行った。

2 ラジオシティ法

ラジオシティとは単位面積あたりの光放射束を意味する。ラジオシティ法は熱力学的エネルギー遷移を背景に持ち、3D シーンを閉じた系とみなしてその平衡状態の光エネルギーを得る。

ラジオシティの算出はシーン内オブジェクトの表面をパッチと呼ばれる小領域の単位に分割する。このときのパッチ間のエネルギー伝達割合 (フォームファクター) の計算がすべての組み合わせの算出に $O(N^3)$ 要する。

2.1 Unshot 法

ラジオシティ計算において高速に収束解を得ることができる Unshot 法は、パッチ i が有する未放射ラジオシティを想定する。未放射ラジオシティの総和が程度収束するまで未放射エネルギーが最大のパッチを選択し、それが全てのパッチにエネルギーを輻射する、というステップ (これを本稿では1イテレーションと呼ぶ) を反復する。これによって全てのパッチに対するフォームファクタを計算する必要がなくなる。

2.2 Möller-Trumbore 交差判定法

フォームファクタを得るため、パッチ同士の可視性を算出する必要がある。そのために、パッチ i, j 間の線分 (これをレイと呼ぶ) と他なるパッチが交差するか判定する必要がある。その方法は種々考案されているが、直線と三角形の交差判定を高速にかつ省メモリで計算できる Möller と Trumbore により考案されたアルゴリズムでは次の式 (1) を u, v, t に関して解くことで交差判定の可否を得る。

$$O + tR = (1 - u - v)V_0 + uV_1 + vV_2 \quad (1)$$

レイの開始点 O と、そこから終点へと向かう相対ベクトル R で表現されるレイ $O+tR$ が座標 V_0, V_1, V_2 において、 $0 < t, u, v, u+v < 1$

を満たせばレイと三角形は交差する。 u, v, t を得るには式 (1) にクラメルの式を適応して式 (2) を解く。

$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \frac{1}{(R \times E_2) \cdot E_1} \begin{bmatrix} (T \times E_1) \cdot E_2 \\ (R \times E_2) \cdot T \\ (T \times E_1) \cdot R \end{bmatrix} \quad (2)$$

ただし、このとき $\det = (R \times E_2) \cdot E_1 \neq 0$ であり、 $E_1 = (V_1 - V_0), E_2 = (V_2 - V_0), T = S - V_0$ とする。 \det はパッチとレイの並行性の評価であるが、数値安定性のために、計算機では0そのものではなく誤差項 ϵ との比較とする。

3 システム概要

Unshot 法を適応したラジオシティ法の計算を行う CPU と FPGA による専用計算機を Zynq7000 ファミリの SoC チップ XC7Z020-CLG484-1 で構成する。FPGA へ5つの座標情報を与え、FPGA が式 (2) を解き、交差判定の結果とする。本研究では、次の4つの最適化に関して評価を行う。

3.1 アルゴリズム変更

従来の CPU の処理では u, v, t の逐次計算の度に交差可否の分岐判定しているが、それを高位合成すると分岐回路が演算回路に複数挿入されて回路デザインが複雑になる可能性がある。そこで、 u, v, t に関して全て計算し終えたのちに交差の可否判定を1度だけ行うことで、回路資源の削減と動作可能周波数の向上を狙った。また、交差判定の計算に除算が存在するが、除算回路を構成した場合演算コストが大きい。しかし、最終的な交差成立の可否を得るには除算は必須ではない。(4)式において除数となる行列式 \det で除算の代替として u, v, t に \det の正負符号の積をそれぞれ代入し、 $t, u+v < 1$ の比較条件を $t, u+v < |\det|$ として同等の交差判定結果を得る。

3.2 転送に伴うオーバーヘッドの削減

FPGA と CPU のデータ転送によるオーバーヘッドの削減のために、レイとパッチを任意の M 個分まとめて転送を行い、一度の FPGA 関数の呼び出しで M 個のパッチの交差判定結果を得るよう変更を行った。このときのパッチ数 N に対して1イテレーションでの転送回数と転送量に関して表1に示す。

3.3 並列化

ループ展開による並列化によって、ループ内の処理を複製しループの総反復数を削減することで、スループットの向上が期待される。本研究ではレイ単位の計算をループ展開する。

表1: 1イテレーションあたりの転送回数と転送量

	非最適化	最適化
転送回数	N^2	$\lceil \frac{N}{M} \rceil^2$
転送量	$O(N^2)$	$O(M \lceil \frac{N}{M} \rceil^2)$

表 2: 1 組の交差判定回路の性能見積もりレポート

	非最適化	A	B	C	D
最大レイテンシ	253	36	50	36	36
最小レイテンシ	53	36	50	36	36
BRAM	0	0	0	0	0
DSP	30	48	48	48	126
FF	5,851	5,400	7,691	5,366	13,069
LUT	8,661	10,961	13,661	10,824	22,168

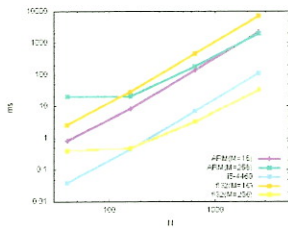


図 1: 1 イテレーションの平均実測値

3.4 計算精度と回路規模のトレードオフ

従来単精度浮動小数点数で計算を行っていた数値を任意 bit 幅固定小数点数で計算した際に、どの程度交差判定の計算を誤るか調査する。その誤差が最終的なレンダー結果に影響しなければ、その限りで回路規模を縮小できる。

4 評価

4.1 アルゴリズムの変更

1 組のレイとパッチの交差判定回路に関してデザインの見積もり評価を行った (表 2)。ここでは 100MHz の浮動小数点計算を行い、レイテンシはそのときの要するクロックサイクルとする。非最適化は CPU 向けコードをそのまま合成し、A はそこから除算除去のみを行う場合、B は分岐を減らし最後に一度のみ交差比較を行う場合、C は A,B を適応した場合、D は C をディレクティブによってパイプライン化指定を行ったものである。非最適化時と比較して、最大レイテンシは A,C,D において約 75%改善したことが分かる。C から D のパイプライン化によって回路リソースの利用量が增大しているが、レイテンシが増加していないことから演算モジュールが並列化されたと考えられる。

4.2 転送に伴うオーバーヘッドの削減

$M = 16,256$ のときパッチ数 N を変化させて障害物のないシーンに対して計算を行った際の実測値の際を図 1 に示す。また、図 1 でのみ FPGA を使った計算は Zynq Ultrascale+ の U96 SoC を利用した。M の大きさによって ARM の CPU, FPGA とともに計算時間が変化したことが分かる。M=256 のときかつ N が大きなきには intel Core i5-4460 よりも高速になった。

4.3 並列化

表 3 は $M = 256, N = 2640$ の room1 のシーンの 1 イテレーションにかかる時間を示している。演算に必要なモジュールは並列数に比例して利用量が增加しているが、実行時間はほぼ変化がなかったため、回路自体は並列化されているものの、メモリアクセスがボトルネックとなっていると推測される。

4.4 計算精度と回路規模のトレードオフ

固定小数点数は正の無限大への丸め、オーバーフローは正負にそれぞれ飽和するモードとする。入力シーンの世界座標は x,y,z 座標

表 3: ループ展開の並列数と回路性能見積もり

並列数	1	2	4	8
実行時間 (ms)	32.2	30.2	31.1	-
予想タイミング (ns)	8.37	8.37	8.37	8.37
最小レイテンシ	110594	111617	55809	27905
最大レイテンシ	52488194	52489217	26244609	13122305
BRAM	0	29	0	0
DSP	18	36	72	144
FF	4006	8191	16369	327709
LUT	6566	13415	26976	53856

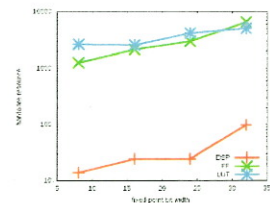
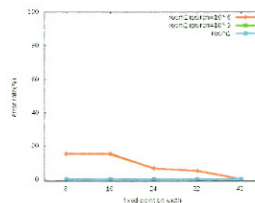


図 2: bit 幅と交差判定誤り率 図 3: bit 幅と資源利用量

ともに 1 から -1 までの値に事前にスケーリングし、オーバーフロー対策を行った。このとき符号ありの整数部分を 1bit のみで固定として残りを小数点以下の表現に割り当てを行ったため、整数部分は常に 0 で符号 bit に相当する。図 2 の x 座標値 40 は便宜的に単精度浮動小数点による結果を示す。

誤差は障害物のない部屋 (room1) に関しては全く生じず、一枚の板が障害物 (room2) としてある部屋は許容する誤差項 ϵ の値に応じて誤差が生じた結果となった。room1 と room2 の $\epsilon = 10^{-3}$ のときの誤差は常に 0 で一致しているため、グラフは完全に重なってしまっている。room1 は障害物がないため交差判定には影響がないが、障害物のある部屋では ϵ の値に応じて精度が変化することがわかった。また、bit 幅に応じて演算モジュールに必要な資源量が変化することを確認した (図 3)。

5 おわりに

本稿では FPGA と CPU が一体になった Zynq-7000 SoC を利用し、ラジオシティ法を適用例として高位合成ツールである SDSoc の最適化手法の評価を行った。ラジオシティ法のプロセスのうち大きなウェイトを占める交差判定アルゴリズムのハードウェア向けの最適化により、最大レイテンシを見積もり上で約 75%ほど改善し、一本の交差判定回路でインテルマシンより高速になった。また、固定小数点表現を用いることで、計算精度とハードウェアのリソース使用量のトレードオフを確認した。正常な並列化ができれば、回路資源が豊富な ZUC15FG SoC を利用して 115 本の交差判定回路を並列に実装可能であり、その場合現状から 115 倍高速になると見積もることができる。そして交差判定を RTL 実装時との比較を行うことで、高位合成技術の優位性が明らかになるだろう。

参考文献

- [1] C.M. Goral, K.E. Torrance, D.P. Greenberg and B. Battaile, Modeling the Interaction of Light Between Diffuse Surfaces, SIGGRAPH'84, 213-222, 1984
- [2] T. Nishita and E. Nakamae, Continuous Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interrection, SIGGRAPH'85, 23-30, 1985