

2Y-1

主観品質を考慮したリアルタイムレイトレーシングの開発

石井 温† 志田晃一郎†

武蔵工業大学†

1 研究背景

リアルタイムレイトレーシングの研究が近年活発に行われている。それらにはCPUのSIMD(Single Instruction Multiple Data)拡張を用いた研究[1]やレイトレーシングをリアルタイム処理するチップアーキテクチャをFPGA(Field Programmable Gate Array)へ実装する研究[2]などが挙げられる。また、GPU(Graphics Processing Unit)と呼ばれるプログラム可能なグラフィックスハードウェア上での実行について、その有効性を検証する研究[3]など、リアルタイムレイトレーシングを実行する環境が整ってきている。

2 レイトレーシングの概要

レイトレーシングとは、3次元コンピュータグラフィックスにおける幾何光学に基づいた画像生成アルゴリズムである。図1のように空間内に視点を配置し、視点からの光線(視線)を描きたいスクリーンに向けて発射、追跡することにより各ピクセルの色を決める。レイトレーシングにおける画像品質のパラメータはサンプリング数(視線の数)、物体との反射・屈折回数である。

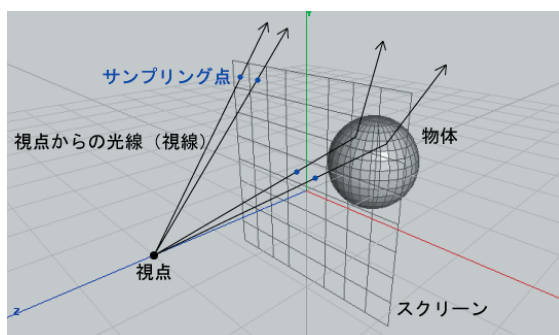


図1: レイトレーシングの概念図

3 動画における品質

動画の品質は、解像度や最大表示色数といった画像品質と1秒あたりの画像枚数(フレームレート)の2つの要素からなると考えられる。人間は、動画について約16 frames/sec以上のとき注目対象が運動していると

知覚できる。また、約33 frames/sec以上の場合は違和感のない滑らかな動画として認識できる[4]。テレビ電話の動画通信においては画像の動きの大きさに依存してフレームレートが変化する場合とフレームレート一定の場合の主観評価実験が行われており、平均フレームレートが同一の条件下ではフレームレート一定の場合の品質が最も良くなる[5]。

4 リアルタイムレイトレーシングの問題

リアルタイムレイトレーシングが出力する動画には以下の特性があると考えられる。

- (1) ハードウェアの処理能力が有限であるために画像品質とフレームレートはトレードオフの関係にある。
- (2) 物体形状を表すポリゴンの移動や数の増減、視点、スクリーンの移動といった描画対象の変化によって画像を生成するのに必要な計算量がフレームごとに変動する。

従来のリアルタイムレイトレーシングでは、画像品質が一定であるためフレームレートが変動するという主と従の関係になっている。

しかしながら、フレームレートが人間の知覚できる上限を超える場合やディスプレイのリフレッシュレートを超過する場合、逆に運動印象を知覚できないほどに低下する場合など、フレームレートは一定の範囲に保ち、画像品質を変動させた方が人間にとってより高い品質(主観品質)が得られる場合もありうると考えられる。

5 研究目的

本研究では、フレームレートを一定の範囲に保つことで主観品質を向上させるGPU上のリアルタイムレイトレーシングを開発する。

6 研究手法

6.1 画像品質及びフレームレートの制御

フレームレートを一定に保つために、以下の提案手法で画像品質及びフレームレートの制御を毎フレーム行う。

- (1) 前フレームの画像生成時間 T_p を計測する。
- (2) 基準時間範囲(上限を T_{max} , 下限を T_{min} とする)と前フレームの画像生成時間 T_p を比較する。

・ $T_p > T_{max}$

Constant Frame Rate Control for Real-time Ray tracing

† On Ishii, Koichiro Shida, Musashi Institute of Technology

画像品質を下げるように反射回数 R またはサンプリング数 S を下げる。

- $T_{min} \leq T_p \leq T_{max}$

反射回数 R またはサンプリング数 S は不変。

- $T_p < T_{min}$

画像品質を上げるように反射回数 R またはサンプリング数 S を増やす。

(3) これらのパラメータを用いて次フレームのレンダリングを行う。

6.2 GPU プログラミング

GPU とはプログラム可能なグラフィックスハードウェアであり、数十個の演算器を並列に動作させてレイトレーシングのような並列性のあるアルゴリズムを高速に処理できる。このため、汎用の CPU で実行するよりも効率的である [3]。本研究では GPU で実行されるプログラムの 1 つであるフラグメントシェーダにレイトレーシングアルゴリズムを記述し、画像品質及びフレームレートの制御をアプリケーション中に記述する。開発環境は GPU に nVIDIA 社の GeForce 6800GT を使い、API は OpenGL、補助ライブラリに GLUT (OpenGL Utility Toolkit)、GPU 用のプログラムを記述するための高級シェーディング言語には、OpenGL Shading Language を用いる [6]。

7 検証方法及び結果

今回の検証で描画対象となるシーンの俯瞰図を図 2 に、また、レイトレーシングの描画結果を図 3 に示す。このシーンでは左右の面が鏡面反射するためウィンドウに占める鏡面反射面が最も多いとき画像生成時間が最大になり、逆にウィンドウに物体が描画されないとき最小となる。描画区間は $0 < \text{rad} < 2\pi$ の間である。

画像生成時間が $T_{min} = 15 \text{ msec}$ 以上 $T_{max} = 20 \text{ msec}$ 以下になるように光線の反射回数を制御した場合と反射回数が一定の場合の画像生成時間の推移を図 4 に示す。検証では制御しないとき全フレームの 8.7% だけが基準時間範囲に入ったものが、制御すると 93.9% に上昇した。レイトレーシングのパラメータであるサンプリング数や反射・屈折回数は整数値となり、フレームレートも画像生成時間の計測が msec 単位であるため離散値となる。このため、パラメータが小さい整数のとき画像生成時間が基準時間範囲をまたいで振動する現象がみられた。

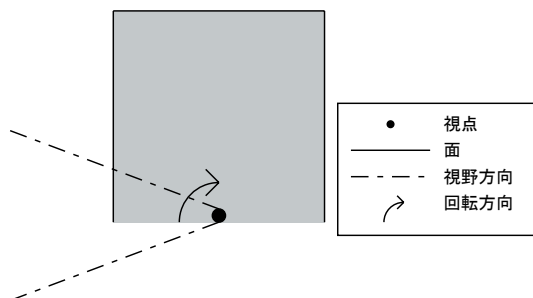


図 2 : シーンの俯瞰図

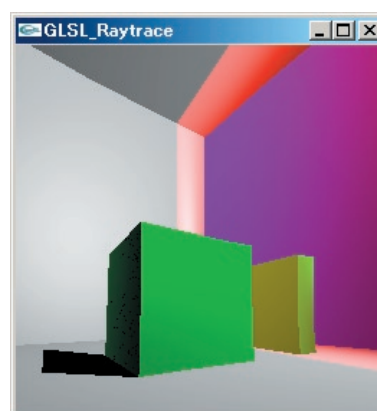


図 3 : レイトレーシングの描画結果

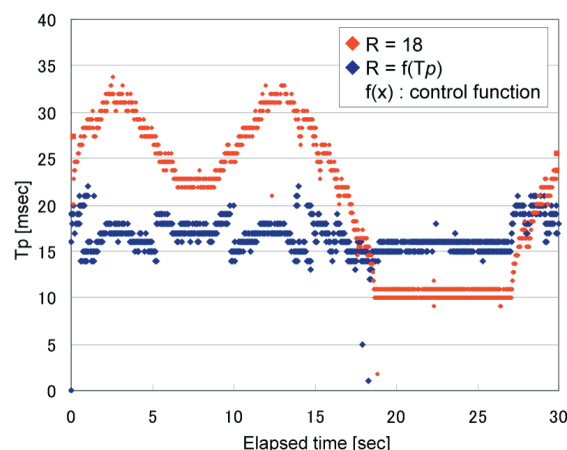


図 4 : 画像生成時間の推移

8 まとめ

本研究では提案手法の検証を行い、画像生成時間に応じてレイトレーシングのパラメータを変動させフレームレートが一定範囲に収まることを確認した。

今後の課題として、画像生成時間が基準時間範囲をまたぐように振動する場合の対策や複数のパラメータを用いた制御方法、また、従来手法と提案手法の主観品質評価を検討している。

参考文献

- [1] Wald, I., Slusallek, P., Benthin, C., Wagner, M., Interactive rendering with coherent ray tracing, *Computer Graphics Forum* **20**, 3, 153-164, 2001.
- [2] Schmittler, J., Woop, S., Wagner, D., Paul, W., Slusallek, P., Realtime Ray Tracing of Dynamic Scenes on an FPGA Chip. <http://graphics.cs.uni-sb.de/~jofis/SaarCOR/DynRT/DynRT.html>, 2004.
- [3] Purcell, T., Buck, I., Mark, W., Hanrahan, P., Ray Tracing on Programmable Graphics Hardware, *ACM Transactions on Graphics*. **21**, 3, 703-712, 2002.
- [4] 中島義明, 映像の心理学, サイエンス社, 1996.
- [5] 林孝典, 井合知, フレームレートの変化が動画像主観品質に及ぼす影響, *NTT R&D Vol. 46*, 793-800, 1997.
- [6] Rost, R., *OpenGL Shading Language*, Addison Wesley, 2004.