

8K-1

光線追跡法における再帰空間分割法のための

データ構造についての考察

本谷 敏康 森田 啓義 武尾 哲也
電気通信大学

1 概要

光線追跡法の高高速化手法のうち、交差判定回数の削減によるものの代表としては空間分割法がある [1, 2]. これは物体が定義されている空間を、適当な部分空間 (ボクセル) に分割し、分割した領域リストに物体を割り付けるものである. 空間分割法はその分割方式により、様々なヴァリエーションがあるが、そのなかで物体の分布に適応した空間分割が実現できるという利点を持つのがオクトツリー分割 (もしくは再帰空間分割, 適応型空間分割) であるが、ボクセルに関する情報を 8 分木で管理するため、隣接ボクセル間移動に時間がかかるという欠点が指摘されている [2].

ここでは、再帰空間分割法の利点をそのままに、8 分木よりも速く隣接ボクセル間移動が出来るようなデータ構造を提案し、また、8 分木との比較実験の結果を報告する.

2 再帰空間分割法と 8 分木構造について

再帰空間分割法での 8 分木は、分割前のボクセルを親ノード、分割後の 8 つの小ボクセルを子ノードとした構造となる.

2次元の場合の再帰空間分割の例を図 1 の上に、それに対応する多分木 (2次元の場合なので、8 分木ではなく 4 分木となっている) を図 1 の右に示す. 図 1 上図において、丸は物体、その中の数字は物体番号を表す. また、図 1 右図の正方形は各ボクセルに対応するノード、付随する数字は対応するボクセルに割り付けられた物体、矢印はポイントを表す.

8 分木を用いた場合の隣接ボクセル間移動とは、光線が

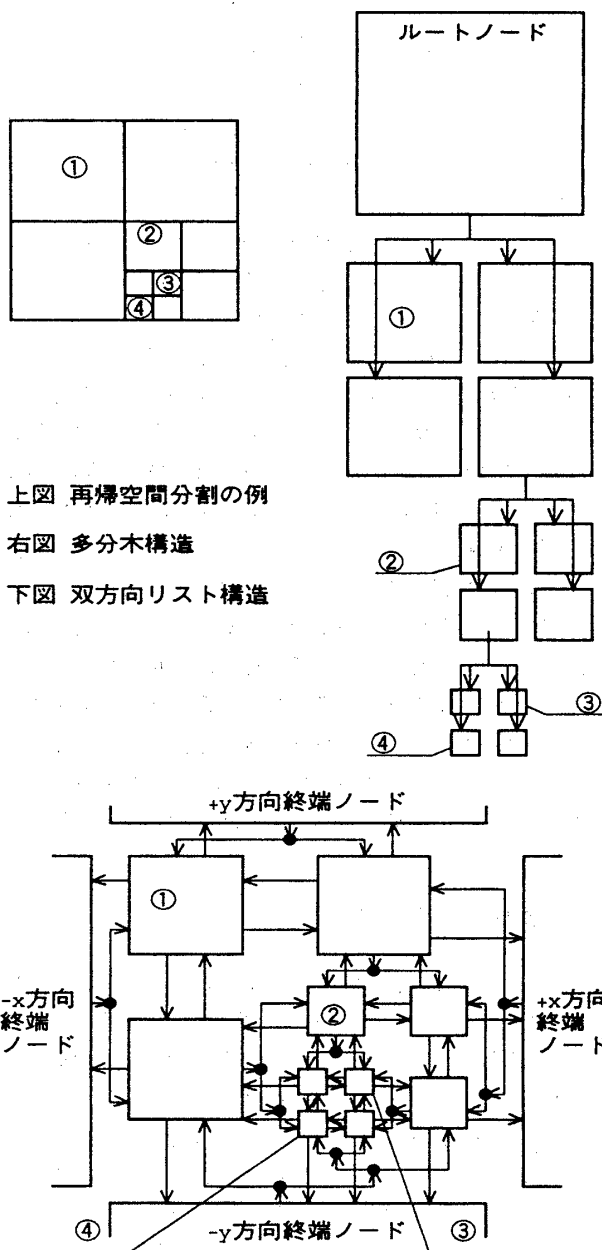


図 1: 再帰空間分割の例及びその多分木と双方向リスト構造

現在通過中のボクセルから外に出る点よりも微小距離だけボクセルの外側にある点を求め、その点を含むボクセルに対応するノードを8分木から探索することである。つまり、その処理時間は移動後のボクセルのノードの8分木における深さ（もしくは分割レベル）に依存してしまう。

8分木において実際に物体情報を持っているのはリーフノードだけであり、中間ノードの存在は冗長である。したがって、8分木は空間分割法（ボクセル間移動）に適したデータ構造ではない。

3 8分木構造に代わるデータ構造の提案

隣接ボクセル間移動の処理を重視するならば、幾何学的に隣接したボクセルに対応するノード同士が双方向ポインタで連結されたような、いわば、双方向リスト構造を3次元に拡張したようなデータ構造を用いるべきであろう。

つまり、ボクセルの各面からポインタがのびるような格好になる。但し、再帰空間分割法ではボクセルの大きさはまちまちで、ある面を境界として隣接するボクセルが複数になる場合があるので、この場合は4分木で連結する。例えば、図1の上のような再帰空間分割には、図1の下のような2次元双方向リスト構造が対応する。図1下図において、正方形は各ボクセルに対応するノード、付随する数字は対応するボクセルに割り付けられた物体、矢印はポインタ、黒丸は2分木のノード（3次元の場合はこれが4分木となる）を表す。

このようなデータ構造を用いた場合の隣接ボクセル間移動とは、光線が現在通過中のボクセルから外に出る面と点を求め、その点をキーとして、その面に対応する4分木を探索することである。この場合、探索する4分木の深さはただか移動前と後のボクセルの分割レベルの差にしかない。従って、明らかに8分木を用いた場合よりも、隣接ボクセル間移動の処理時間は減少する筈である。

4 実験とその結果

8分木、3次元双方向リストを用いて空間分割法を実現した2種類のレイトレーシングプログラムを作成し、物体の分布や視点など、同一の条件のもとで画像を生成させたときの処理時間、使用メモリ量等を比較した。

物体数が820個のときの処理時間を図2に示す。当初の期待通り隣接ボクセル間移動の処理時間の削減に成功している。今回の実験では物体数が10,91,820,7381の場合のデータをとったが、いずれの場合でも同様に、ボクセル間移動処理時間の増加が緩やかになる傾向がみられ、約60から70パーセントの処理時間で同一の画像の生成を実現することができた。

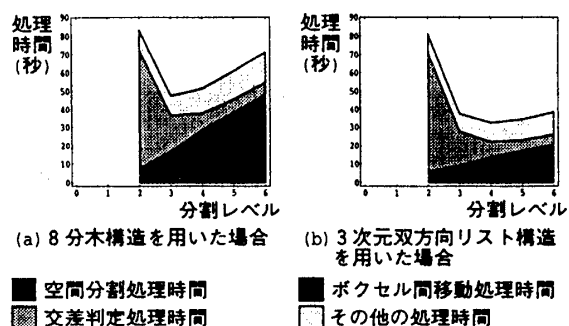


図2: 実験結果 (処理時間)

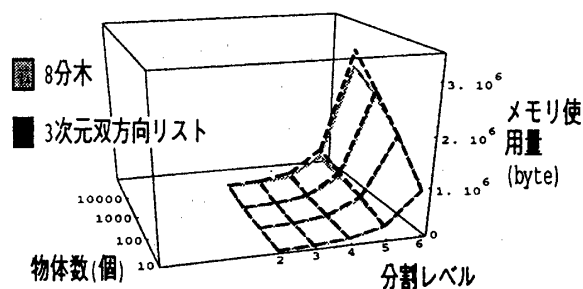


図3: 実験結果 (メモリ使用量)

また、分割の際にボクセルのために使用されたメモリ量を図3に示す。3次元双方向リストで使用されるメモリは、8分木の場合のたかだか4割増し程度に抑えられている。

5 まとめ

光線追跡法に適用する再帰空間分割法に適した、8分木に代わるデータ構造を提案し、その有用性を実験により確認した。同じ程度のメモリ使用量で、かつ、処理時間の削減を実現した3次元双方向リスト構造は8分木よりも優れていると言えることができる。

尚、本研究に関していろいろと助言して下さった、電気通信大学大学院情報システム学研究所の星守先生に深く感謝の意を表し、お礼の言葉といたします。

参考文献

- [1] Andrew S. Glassner. Space subdivision for fast ray tracing. *IEEE Computer Graphics and Applications*, Vol. 4, No. 10, pp. 15-22, October 1984.
- [2] Akira Fujimoto, Takayuki Tanaka, and Kansai Iwata. Arts: Accelerated ray-tracing system. *IEEE Computer Graphics and Applications*, Vol. 6, No. 4, pp. 16-26, April 1986.