空間分割型並列レイトレーシング法の実現と評価

吉谷 崇史　　　山崎 勝弘
立命館大学理工学部

1　はじめに

レイトレーシング法では、視線と物体との交差判定に最も時間を要する。空間を小さな空間に分割し、その空間内の物体のみについて交差判定を行うことで、時間を短縮することができる。本研究では分割された各空間をトーラス接続されたマルチトランスピュータT805(以下TP)64台に割り当て、高速化を図る。

2　空間分割型レイトレーシング

ある空間に物体を配置し、ボクセルと呼ばれる一定の大きさの空間に分割する。その各ボクセルに属する物体のみを登録する。次にレイトレーシング計算を行う。基本的な計算では、すべての物体について交差判定を行ってきた。空間分割の場合、レイがあるボクセルに入ると、そのボクセルに属する物体に対してのみ交差判定を行う。このアルゴリズムをマルチプロセッサに適用し並列処理を行う。それぞれのボクセルを各プロセッサに割り当て、プロセッサ間で光線情報パケットをやり取りすることで画像を生成する。図1のように$4 \times 4 \times 4$の分割を行い、各ボクセルには、ボクセル番号(ijk)を割り当てる。i、j、kの値によって、トーラス接続された各TPには番号が付けられる。

各プロセッサは、メインタスクと通信タスクからなり、さらに通信タスクには各ポートからの入出力スレッドが10個存在する。メインタスクではレイトレーシングの計算を行い、通信タスクでは送られたレイパケットが自分の割り当てであればメインタスクへ、そうでなければ該当するプロセッサへ送る。

3　3DDDAによる光線移動

空間分割ではボクセルで空間を管理しているため、光線がどのボクセルを通過するかを知る必要がある。等分割による空間分割を行うため3DDDAを用いる。DDA(Digital Differential Analyzer)とは、CRTのディスプレイの解像度による線分断裂を防ぐための、線分発生アルゴリズムである。2次元の場合を2DDDA、3次元の場合を3DDDAと呼ぶ。図2は2DDDAを表し、3DDDAはz軸方向に拡張したものである。点1で光線が(i, j)=(0,0)のボクセルに入ったとする。次にどのボクセルを通るか決めるためtxとtyを調べる。$tx < ty$より、(i+1, j)=(1,0)を通ることになる。

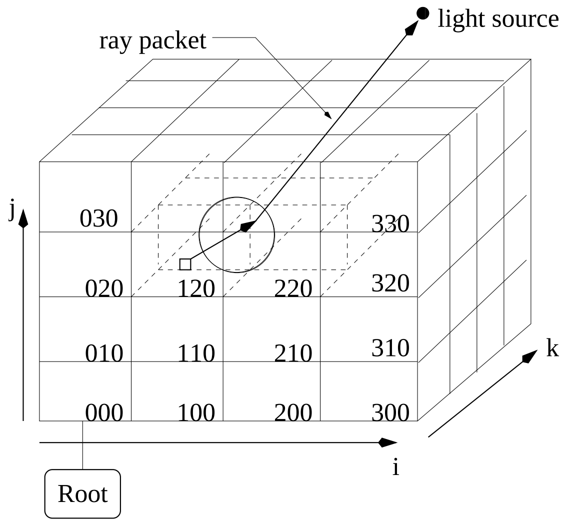

図1: 空間分割 ($4 \times 4 \times 4$)

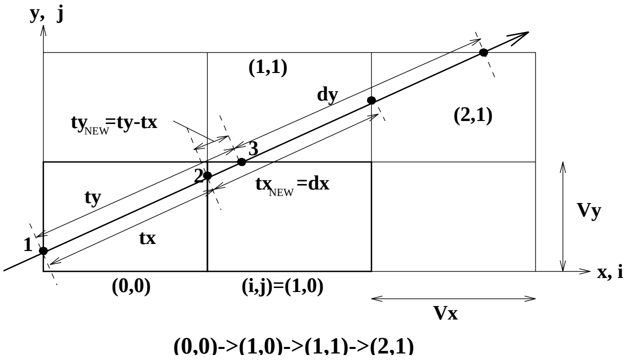

図2: 2DDDA

4　レイパケットとリターンパケット

レイパケットとはその光線に関するデータを含んだパケットである。このパケットがボクセルに入ったときに、参照され計算される。またリターンパケットとは、計算された結果データを含んだパケットである。表1にレイパケットの定義を示す。リターンパケット

Realization of Parallel RayTracing using Spatial Subdivision
Takashi Yoshitani and Katsuhiro Yamazaki
Department of Computer Science, Ritsumeikan University.

は、画面上での座標と計算された色 (r,g,b) を持ったパケットである。

ルート TP からレイパケットを必要なボクセルに送り、物体との交差判定を行う。交差物体がなければ次のボクセルにレイパケットを送り、それを繰り返す。交差物体が見つかると、影の計算をするため、そのボクセルから光源の方向にレイパケットを送る。通過するボクセル内で交差判定をし、交差する物体があれば、影の計算をしてリターンパケットとしてルートに返す。レイパケットが物体と交差しないときは、空間の外に出る前のボクセルからリターンパケットを返す。

表 1: レイパケット

	変数名	内容
	Num(x,y,z)	現在のボクセル番号
	GoNum(x,y,z)	次のボクセル番号
*	Layx,Layy	計算中の画面上での座標
*	Kind	レイの種類
	Weight	レイの重み
	E(x,y,z)	光線の方向ベクトル
*	I(r,g,b)	計算された交点の輝度
	x,y,z	視点と物体の交点座標
	n(x,y,z)	物体上の交点の法線ベクトル
次のボクセル検索用データ		
交差する物体のデータ		

* リターンパケットの情報

5　実験と考察

実験では、鏡面などの反射を繰り返す物体を除いた通常の球のみについて実行した。図 3 は、UNIX1 台による空間分割を評価したものである。用いた画像は 64,256,512,1024 物体を各ボクセルに均等に配置したものである。図から、各物体数までの分割では性能は上がっているが、それを越えると逆に悪くなることが分かる。また、物体数が増えるほど実行時間の減少率も大きい。1024 物体のとき 32 分割で 313 秒、1024 分割で 60 秒と交差判定の回数が約$\frac{1}{5}$に減少している。図 4 では、各ボクセルを TP に割り当てて、実行した結果である。物体数を 64,192,256,512,1024,8320 に変化させ、それらを均等に配置したものと、ランダムに配置したものを画像として用いている。ランダムな画像が多く時間がかかっているのは、物体がいくつかのボクセルにまたがり、実際に全プロセッサに配置される物体数が増えているためである。実行時間がリニアに増加しないのは、用いた画像によってレイが最も手前の空間で終了する場合と、最後の空間まで通過する場合の比率が異なるからであると考えられる。

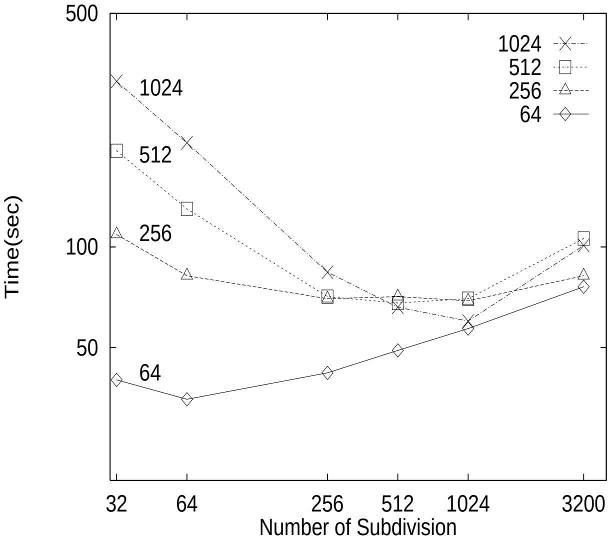

図 3: UNIX1 台での空間分割の評価

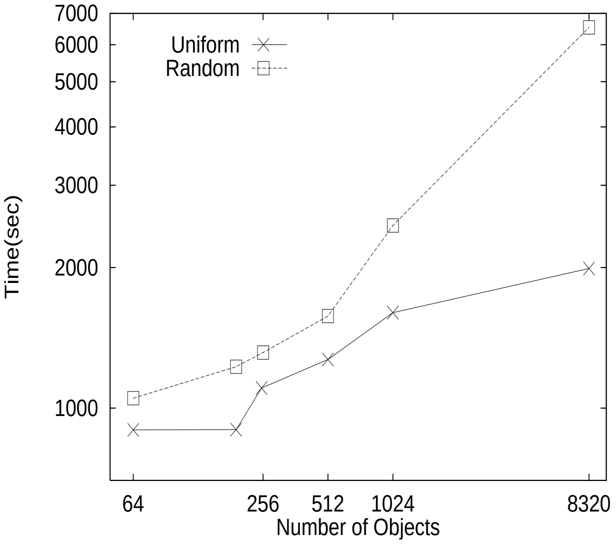

図 4: TP64 台、64 分割による実行結果

6　おわりに

空間分割では物体数に等しい分割数で、交差判定が最も減少することを示した。物体数、分割数、TP 台数を変化させた評価が今後の課題である。

参考文献

[1] 原田 他 ”画面分割と空間分割によるレイトレーシング法の高速化”，情処学 50 回全大，1C-6，1995.