

ボクセル分割の最適化によるレイトレーシングの高速化手法

坂井俊雄 渡辺範人 三好雅則 福田善文
(株)日立製作所日立研究所

本稿では、レイトレーシングにおける空間分割法を最適化する方式について述べる。空間分割法にはオクトリー分割法とボクセル分割法がある。それぞれの方法を分析して、各セル（空間分割によって作成された部分空間）の交点探索における追跡処理の高速性を重視し、ボクセル分割法を採用した。さらに、ボクセル分割法の分割数をシーンを構成する形状の大きさの平均値から求めるようにした。本方式を適用して性能評価した結果、各種のシーンにおいてレイトレーシングによるレンダリングを高速化することを実証できた。

A Method for Accelerating Ray Tracing by Optimizing Regular Spatial Subdivision

Toshio Sakai Norito Watanabe Masanori Miyoshi Yoshihumi Fukuda
Hitachi Research Laboratory, Hitachi Ltd.

This paper presents a method that optimizes the spatial subdivision method for accelerating ray tracing. The spatial subdivision methods can be categorized as octree spatial subdivision methods, or regular spatial subdivision methods. The method which we proposed employs the regular subdivision method, which is faster than the octree spatial subdivision method in cell traversals. The regular grid size in the method is the same as the average size of all objects in a scene. We have found that the method realize to reduce ray tracing process by performing various experiments.

1. はじめに

近年、ワークステーションの機能向上に伴い、フォトリアリスティックな映像が高速に制作できるようになりつつあるが、更に高画質な映像を高速に作成することが要求されている。高画質な映像は物体表面のハイライト・ラインや周りの物体の写り込み、影などによって曲面の滑らかさや凹みなどを確認することができ、また、物体のきめ細かな模様やラベルを表現することにより実在感を提供することができるような映像である。また、デザインワークでは、デザイナーの創造的思考を阻害しないように、対話性を満足する高速な映像生成が必要となる。

レイトレーシング法はフォトリアリスティックな映像を作成するのに適した方法である。レイトレーシング法はハイライト・ラインや周りの物体の映り込みや影を単純なアルゴリズムで正確にシミュレーションできるので、プレゼンテーション用の製品の最終的な出力として使用されている。しかし、レイトレーシング法は多くの時間を必要とするため、現状ではデザインワークとして使用できない。一般に、レイトレーシング法の画像生成処理時間の大半は交点探索処理であり、この高速化が重要である。

交点探索処理を高速化する手法の中で、空間分割法はもっとも有効な手段だと評価されている。空間分割法は、形状が存在する空間をセルと呼ばれる部分空間に分割し、形状が存在するセルに形状を登録しておく。交点探索処理では、光線が通るセルに登録された形状との交点計算を行う。この方法にはオクトリー(Octree)分割法[1][2]、ボクセル分割法[3]、階層型ボクセル分割法[4]がある。しかし、空間分割法のそれぞれの手法は特質が異なり、レンダリングするシーンの状態によって処理速度が異なっている。

現在までに報告されたオクトリー分割法とボクセル分割法の比較ではシーンの状態に対する定量的な評価がなされていない。オクトリー分割法は、形状が一部に密集した状態ではボクセ

ル分割法より早くなる。しかし、どのくらいの密集度でオクトリー分割法がボクセル分割法より早くなるのか不明である。そのため、ボクセル分割法とオクトリー分割法のどちらがより高速な分割法なのか、結論を得ていない。

本報告ではまずそれぞれの空間分割法の問題点を分析し、その空間分割法の特質から最適な空間分割条件を導きだし、最適な空間分割法について実証も含めて検討する。

2. 空間分割法の問題点

2.1 オクトリー分割法

オクトリー分割法は図1(a)に示すように形状が存在する空間を8分割していく方法である。この分割は形状が1つだけ存在するようなセルが得られるまで延々と続く。このようにして得られたセルは8分木の階層を持つ(一般にこの分木の構造をオクトリーと呼んでいる)。実際のインプリメントでは、形状がただ1つだけ存在するようなセルに分割することはメモリと処理時間の問題から不可能であるので、セルに登録される形状数の制限を多くし、階層の深さを制限している。

一般にオクトリー分割法の処理速度は形状の分布によらず安定している[5]。処理速度が安定しているのは、空間がオクトリー分割によってセルに登録される形状数が一定になるように分割されるからである。

しかし、オクトリー分割法の問題は光線がオクトリー分割法によって作成されたセルを追跡しなければならないことである。セルの追跡は投影計算によって次のセルを求めることによって実行される[4]。投影計算は光線がどのセルを通るかを計算する処理である。従って、オクトリー分割法の次のセルの探索時間はボクセル分割法の探索時間より時間を要する。まして、隣接したセル内を探索する際にも階層構造にしたがって追跡しなければならないので、オクトリー分割法では1本の光線に対するセル追跡に時間を費やすことになる。特に、交点を持たない

光線がシーンを貫くときにはすべてのオクツリーを検索しなければならないので、この現象が著しく現われる。

2. 2 ボクセル分割法

ボクセル分割法は図1(b)に示すように全ての形状が存在する空間を等分割する方法である。等分割されているので、オクツリー分割法に比べセルを追跡する処理のスピードが早まる[5]。すなわち、3DDDA法(1回の整数のインクリメントと1回の浮動小数点の足し算)によって、次のセルを算出することができる。また、ボクセル分割法は、光線が交点を持たない場合でも、オクツリー分割法に比べ、追跡しなければならないセル数が少ない。例えば、空間を $128 \times 128 \times 128$ に分割した場合においても、1つの光線に対するセル追跡数は高々 128×2 位である。

しかし、この方法は形状が密集している場合、オクツリー分割法に比べ処理速度が落ちる。セル内の交点探索はセルに登録されている全ての形状との交点計算を必要とするからである。形状との交点計算はセルを検索する処理に比べ非常に遅い[6]ので、1つのセルに大量の形状が集中した場合、そのセル内の交点探索に時間がかかる。従って、形状が密集している場合、ボクセル分割法の方がオクツリー分割法より時間がかかることになる。逆に、密集を避けるためには分割数を増大すればよいが、大量のメモリが

必要となる。

3. 最適な空間分割方法

3. 1 最適空間分割条件

オクツリー分割法とボクセル分割法から最適な空間分割の条件は以下のようにまとめられる。

(1) セルに登録される形状数は極力少なくする。

この条件は、ボクセル分割法で形状が密集している場合にオクツリー分割法に比べ処理速度が落ちる事実から得られる条件である。形状との交点計算はセルを検索する処理に比べ、非常に遅い。セルの交点探索はセル内に登録されている全ての形状との交点計算を必要とする。大量の形状が集中したセルは大量の交点計算をしなければならないので、形状数が少ないセルに比べ時間がかかるからである。

(2) ボクセル分割を多用してセルを極力小さくする。

この条件はボクセル分割法はオクツリー分割法に比べセルの検索数が少なく、次のセルを算出する演算が速いという事実から得られる条件である。ボクセル分割法はメモリの問題を考えなければ、分割数を増大させても交点探索に影響しない。だから、セルを細かくするにはボクセル分割法で行なったほうが有効である。

(3) 階層は浅くする。

この条件はオクツリー分割法が階層が深い

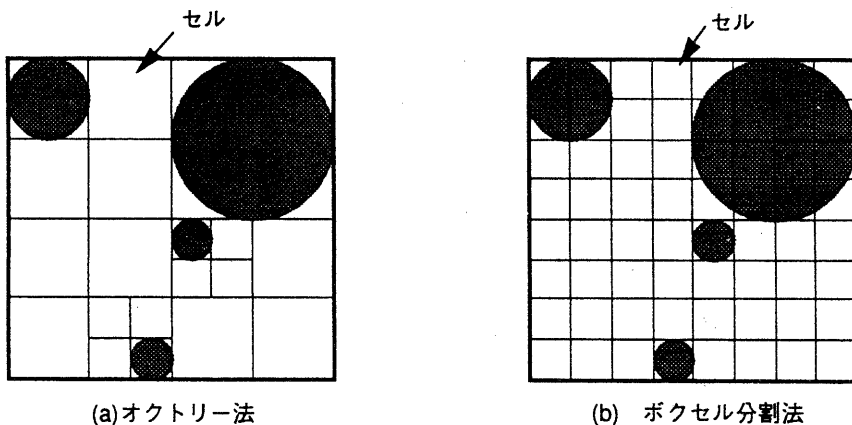


図1 空間分割法の比較

め、ボクセル分割法に比べセル検索に費やす時間がかることから得られる条件である。

3. 2 ボクセル分割法の最適化

3. 1の最適空間分割条件から考えると、単純なボクセル分割法が最も高速な空間分割方法となる。オクトリー分割法は、セル内に登録される形状数は少なくなるようにすることはできる。しかし、オクトリー分割法によって作成されたセルの追跡には、ボクセル分割法より多くの投影計算を必要とする。ボクセル分割法での投影計算は1光線当たり1回であるのに対して、オクトリー分割法の投影計算は階層が1つ深くなると8の倍数で増えていく。空間分割法としては、交点探索処理の際のセル追跡には極力時間を費やしたくないので、ボクセル分割法の方

が優れている。従って、セルに登録される形状数を少なく押さえるような分割数をもったボクセル分割法がもっとも高速な方法となる。

では、何が分割数を決定しているのか。図2に最適なボクセル分割の存在を裏付ける処理結果を示す。図2はレンダリングのテストとしてボクセル分割の分割数を4、8、16、32、64、128と変化させたときのレンダリング時間をプロットしたグラフである。図3に同時に作成された画像を示す。また、表1に各作成した画像の特徴を示す。画像のサイズはすべて320×240である。アンチエイシングは行っていない。図2の結果からシーンによって、最適分割数、すなわち計算時間が最小になる分割数が異なっていることがわかる。分割数が低いと

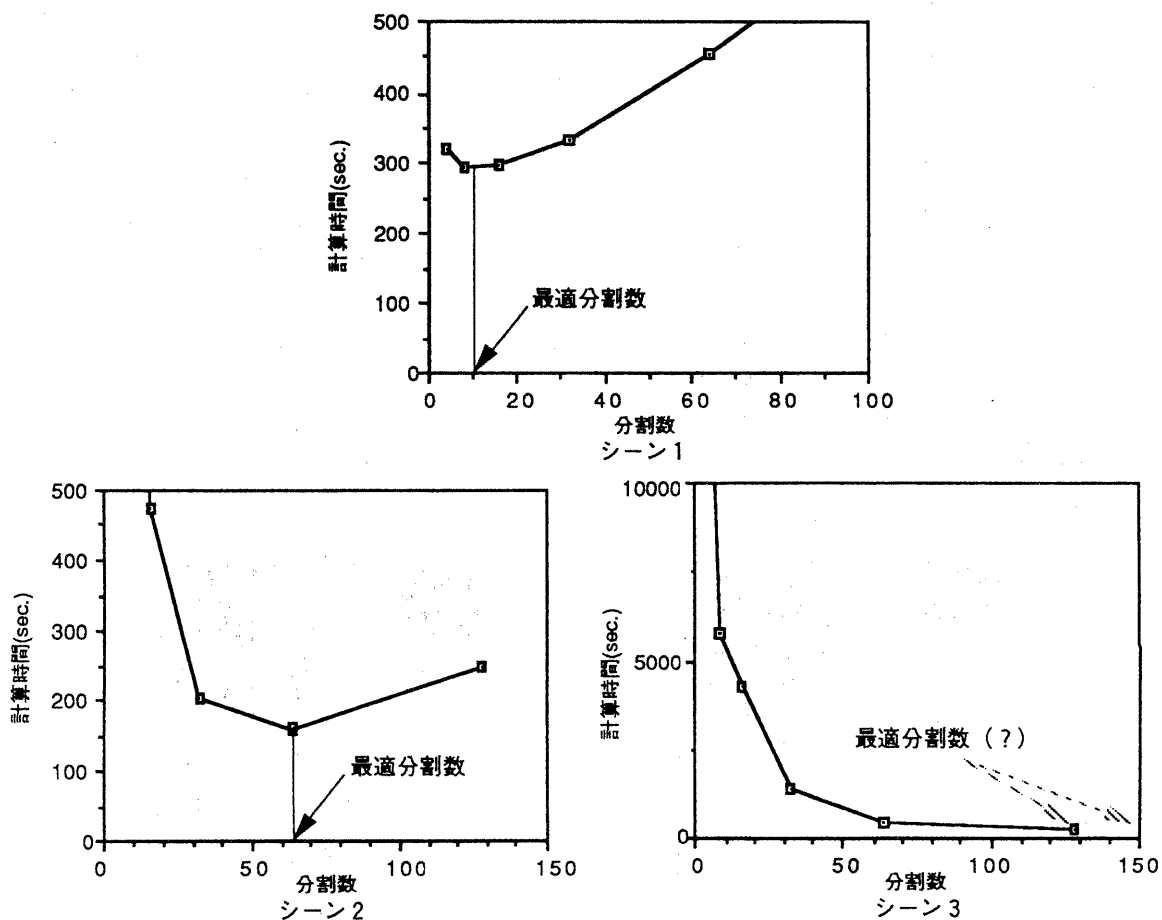
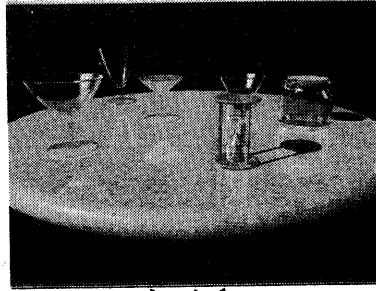
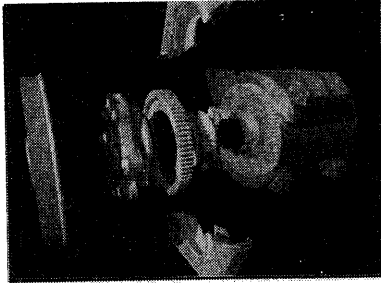


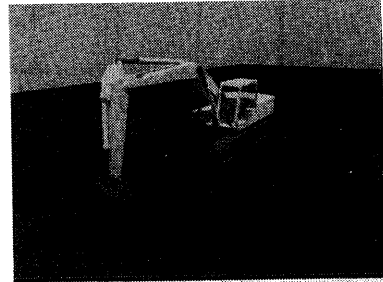
図2 シーンによる最適ボクセル分割数の変化



シーン1



シーン2



シーン3

図3 レンダリング画像

表1 各シーンの特徴

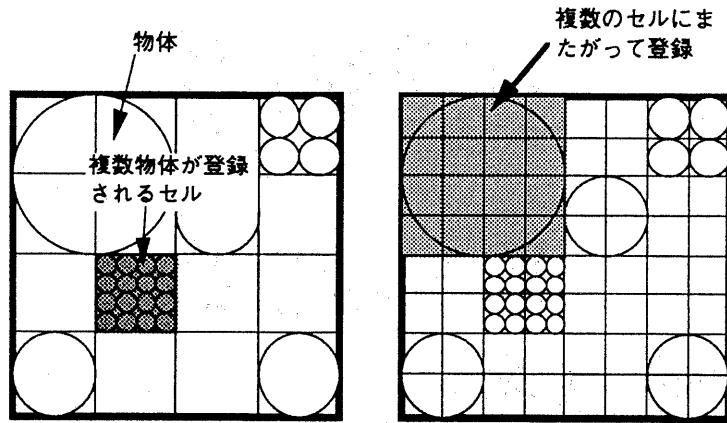
シーン名	曲面数	追跡の深さ	使用した光線数	備考
シーン1	70	4	274255	2次曲面のみ
シーン2	58549	1	137353	三角形のみ
シーン3	13856	2	157928	三角形のみ

ところで、最小になる場合もあれば、128を越えても計算時間が減少している場合もある。

この現象を図4を用いて説明する。まず、最適な分割数より小さい分割数において計算時間がかかっているのは、図4(a)に示すように、セルの大きさが形状の大きさより大きく、セルに登録される形状数が増えたからである。セルに登録される形状数の増加は形状との交点計算時間の増加を生み、計算時間がかかる。逆に、最適な分割数より大きい分割数において計算時間がかかっているのは、図4(b)に示すように、形

状の大きさに比べセルの大きさが小さくなり、形状は複数のセルにまたがって登録される。複数のセルに登録された形状と光線との交点計算はその形状に登録しているセル毎に行なわれる。これは空間分割法の欠点である。光線から見れば、交点計算する形状が増えたことになる。分割数を増やせばこの現象はさらに悪化するので、計算時間がかかるようになる。以上のことから、ボクセル分割の最適な分割数は形状の大きさに関与しているといえる。

そこで、形状の大きさから分割数を決定する



(a) 分割数小く 最適な分割数 < (b) 分割数大

図4 分割数と形状の関係

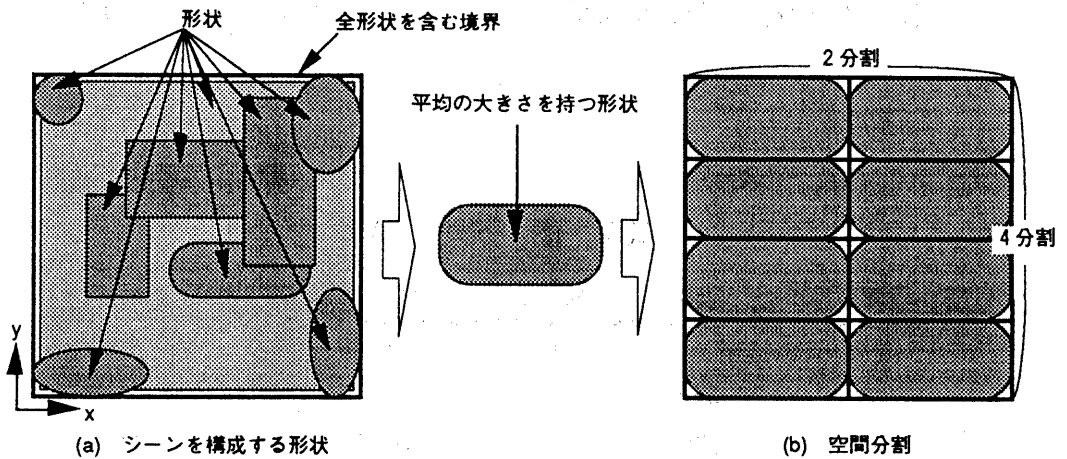


図5 形状の大きさによるボクセル分割法

方法として各形状が持っている形状境界の x 、 y 、 z 方向の大きさの平均を元に分割数を決定する方法を検討した。この方法は図5に示すように、空間に存在する形状は全て平均的な大きさを持ち、しかも一様に分布していると仮定して、空間を平均の大きさで分割する方法である。分割は同図に示すように x 方向と y 方向で独立に分割する。同図の例では x 方向を2分割し、 y 方向を4分割している。この方法は各形状の形状境界に注目するだけなので簡単に求めることができ、コストも少ない。この方法によって空間が分割された場合、平均的な大きさを持つ

た形状は1つのセルに包含される可能性が高くなる。

しかし、ボクセル分割法には分割数が増えると無駄なセルが増し、必要以上に多くのメモリを使用するという問題がある。この問題は形状が密集しているシーンで顕著になる。形状が密集する状態は、シーンを構成している物体が複数の小さい形状からなるときに発生する。物体が直接交点計算できないようなとき、物体を三角形に分割して計算する場合などがそれに当たる。このとき、形状の平均の大きさは非常に小さくなるので、ボクセル分割数は大きくなる。

密集している状態以外の空間は空き領域が多いので、無駄なセルが多くなる。

4. 結果と検討

図6に形状の大きさの平均を基準とした分割量に対する処理時間の変化を示す。各グラフは図3に示したシーンをレンダリングしたときの処理結果である。相対分割量1は形状の大きさの平均で空間を分割したことを意味している。具体的には図5において相対分割量1の分割とは空間がx方向に2、y方向4に分割されることである。相対分割量が下がると分割は荒くなる

り、上がると、分割は密になる。例えば、図5において相対分割量2の分割とは空間がx方向に4、y方向に8に分割されることである。

各シーンとも0.5から1.0の間で最小になっているのが分かる。形状数が多いシーンの処理速度は相対分割量の変化に対して激しく変化し、相対分割量1が最小になっている。しかし、形状数が少ないシーンの処理速度は相対分割量を変化しても影響は少ない。形状数が少ないシーンでは、空間の分割が荒い場合でも、セルに大量に形状が登録されることはないからである。

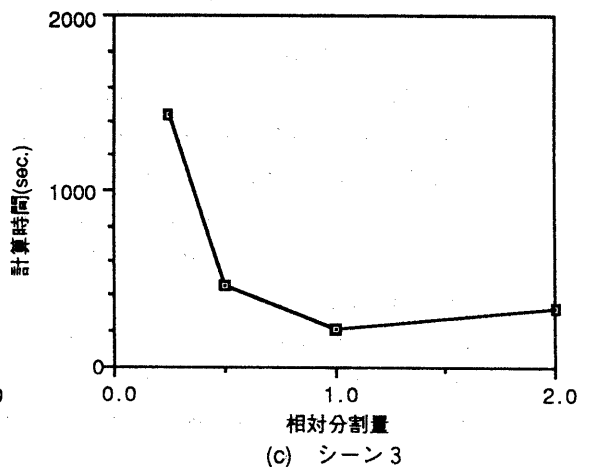
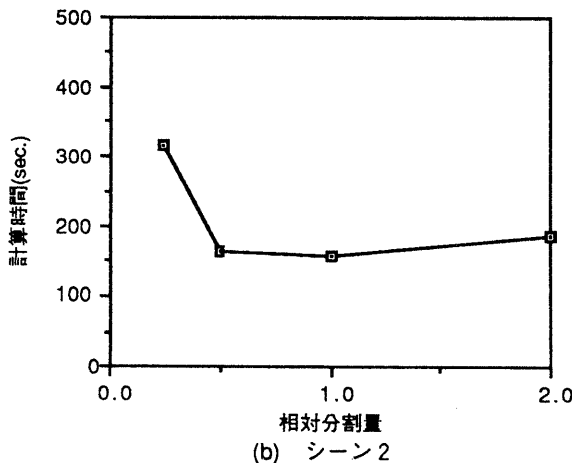
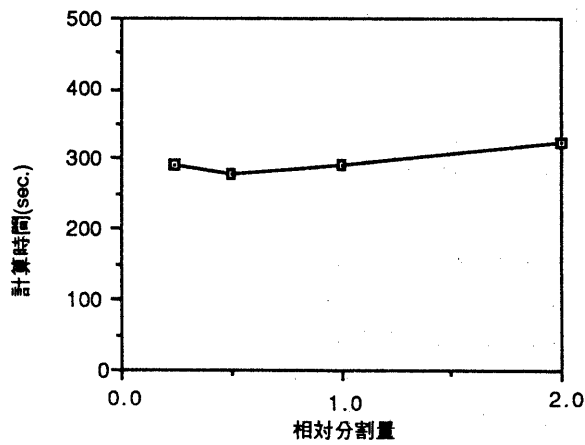


図6 形状の大きさの平均を基準としたレンダリング時間

以上のことから形状の大きさの平均からボクセル分割法の分割数を決定する方法は有効と考えられる。また、形状の大きさは有効でないまでも分割数を決定するためある程度の指針を与えてくれる。今回の報告は分割数を決定することだけに注目して処理速度を測定している。それゆえ、形状が密集している場合の最適な分割数はメモリの問題も加味すると今回の結果と異なる可能性がある。なお、最適な相対分割量に0.5から1.0という幅がある理由は不明である。

5. まとめ

レイトレーシング法における最適な空間分割法はボクセル分割法であり、その分割数をシーンを構成する形状の大きさから求める方法が有効であることを示した。本文で述べたように、形状が密集する場合は本手法だけではメモリなどの問題から、満足できる空間分割法とはいえない。今後はボクセル分割法のメモリの問題を解決する手段として階層型ボクセル分割法を検討する。階層型ボクセル分割法の初期分割数は本稿で述べた方式を用いて求める。階層の深さは3。1の最適空間分割条件から2にする。セルに登録される形状数のしきい値、第2階層の分割数は何によって決めればよいのか不明である。今後はこれらのパラメータを決定することによりさらに高速化を図る予定である。

6. 参考文献

- [1] Glassner, Andrew S., "Space Subdivision for Fast Ray Tracing", IEEE Computer Graphics and Applications, Vol.4, No.10, October 1984, pp.15-22.
- [2] 松本均, 村上公一, 「Octreeデータ構造を用いたRayTracing法」, 情報処理学会第27回全国大会, pp.1535-1536, 1983
- [3] Fijumoto, Akira, Takayuki, Tanaka, and Kansai, Iwata, "ARTS: Accelerated Ray-Tracing System," IEEE Computer Graphics and Applications, Vol.6, No. 4, April 1986, pp.16-26.
- [4] David Jevans et al, "Adaptive Voxel Subdivision for Ray Tracing", Graphics Interface'89, pp.164-17

2

[5] 動向調査レイトレーシング:高速化競うレイトレーシング、日経CG1988.12、pp98-107.

[6] 「パソコン・レイ・トレーシング・ソフトを体験しよう」、日経CG、1991年1月号