

# Path Coherency in Ray Tracing

## 4Q-2

森島 裕 川合 慧

東京大学 理学部

### 1. はじめに

計算機の高速度化、高速な計算機や高品質なCRTの低価格化に伴い、3次元グラフィクスにおいてリアリズムの実現に関心が払われるようになってきた。リアリズムを実現するいくつかの手法の中で、光線追跡法[1]は、鏡面による映り込み、透明物体による屈折、影など3次元の奥行き感を出す現象を統一的に扱い高品質な画像を生成するアルゴリズムとして高い評価を得ている。

しかし、光線追跡法には画像生成に極めて時間がかかりエリアシングを除去し難いという2つの欠点がある。画像生成に時間がかかる原因の1つは各画素を全く独立に扱ってコヒーレンシを利用していないからであり、エリアシングが除去し難いのはその画素付近で画像が大きく変化しているのに画素の中心点でのサンプリングしか行なわれないからである。これら2つの欠点は画素の大きさに比べた大小のため見かけは異なるものの、どちらも画像の複雑さを測る方法の欠如によるものである。

本稿では、光線追跡法で生成される画像の複雑さを測る方法として、普通の意味のコヒーレンシを拡張したPath Coherency (PC) という概念を導入し、PCがあるスクリーン領域を求める方法を提案し、PCを利用して高速化、エリアシングの防止を行なう光線追跡法アルゴリズムを提案する。

### 2. Path Coherency (PC)

3次元グラフィクスにおいて、コヒーレンシは普通、可視物体が同一であることを意味し、画像の表示を高速に行なうためなどに利用される。例えば、光線追跡法を高速化するためにコヒーレンシを利用して、可視物体が同一の領域にスクリーンを分割して、光線と当たる可能性のある物体を限定することはよく行なわれる。

しかし、光の多段の反射・屈折の現象をシミュレートして3次元の奥行き感を出す光線追跡法においては、単なる可視物体の同一性だけでなく、反射・屈折の効果によって間接的に見える物体までを含めた同一性を考える

ことが必要である。我々はこれを、光線のpathまでを含めた同一性という意味で、Path Coherency (PC) と呼ぶことにする。普通の意味のコヒーレンシは1段のPCと考えることができる。PCがある領域では画像は緩やかに変化し、PCがない領域では画像は急激に変化していると考えられる。PCがある領域を求めることは、(普通の意味でのコヒーレンシと同様に) 画像の生成・表示を高速化したり、テクスチャなどによるエリアシングを防止するために役立つ。

### 3. PCがある領域の求め方

PCがある領域を求めるためには、全領域における光線の集合(光束)を追跡する必要がある。2段以上のPCを求めるためには、光束の反射・屈折を計算する必要がある。平面による反射は、光束が平面の鏡像の位置にあったものとみなして追跡するだけでよい[2]。しかし、屈折や曲面による反射を行なうと光束は大きく広がり、形を複雑に変化させ、1点から出ているとみせなくなり、光束の内部の光線が光束の側面に出たりする。

光束に対して反射や曲面による屈折の方程式を解くことは困難なことが多い。我々は、光束を完全に含む円すいを追跡することにする(図1)。光束を含む円すいが

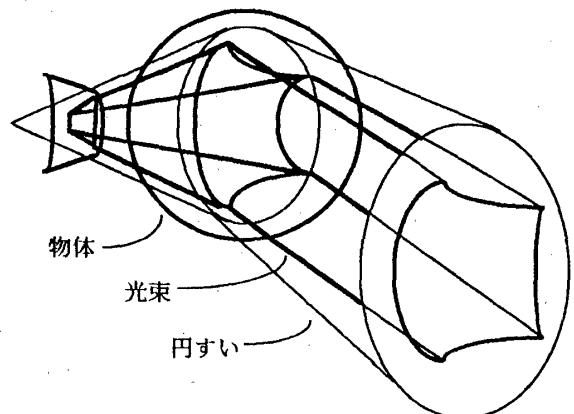


図1

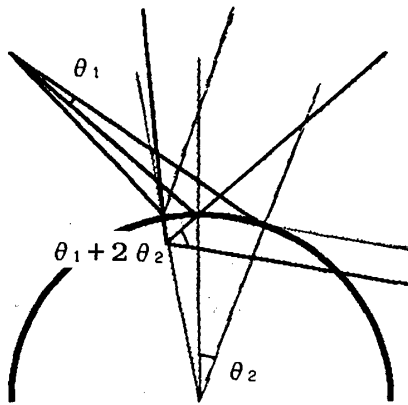


図2

最も近くで交わる物体で完全にカットされているならば、光束もその物体で完全にカットされているので、PCがある。光束を含む頂角 $\theta_1$ の円すいが反射物体に当たったときに発生させるべき新しい円すいは次のように求めることができる。まず、円すいの中心線の反射線を新しい円すいの中心線にする。円すいが当たる全領域（を含む領域）での法線と円すいの中心線が交わる点での法線との角度のずれの最大値を $\theta_2$ とする。簡単な幾何から、新しい光束中の光線と新しい円すいの中心線との角度のずれは $\theta_1 + 2\theta_2$ 以下になる。従って、頂角を $\theta_1 + 2\theta_2$ とし、古い円すいと当たる全領域を含むように頂角を決めれば新しい円すいが決定する（図2）。

屈折の場合も同様である。

#### 4. PCを利用した光線追跡法アルゴリズム

Warnockのアルゴリズムのようにスクリーンの4分割を繰り返しながら、ある段のPCが領域全体にあるかを調べ、PCがある領域では補間によって輝度を求める。

0) 最初はスクリーン全領域の1段のPCを考える。

1) スクリーン正方形Aの4隅でのPCの確認

Aの4隅でもう1段先まで光線を追跡し、光線が当たる物体を調べる。（交点における法線ベクトル、光源から交点へ方向ベクトルなども求める。）Aが画素の大きさに比べて充分小さくなったならば、3)に行き4隅での輝度値を平均して終了する。物体が異なればPCがないのでAを4分割して1)に戻る。

2) A全領域でのPCの確認

それまでの追跡で平面反射しかしていない場合はAから出た光束はAの4隅での光線を辺とする4角すい台の形を保っているが、屈折や曲面反射をしている場合はAから出た光束は複雑な形になっているので、A全領域を

含む円すいをもう1段追跡する。それぞれ4角すい台か円すい台が、他の物体に交わっていないければA全領域でPCがあることがわかる。PCが確認できなければAを4分割して1)へ戻る。

3) 輝度値の計算

Aの4隅で1)において求めたベクトルを補間して、Aの内部の画素での輝度値を求める。特に、ベクトルがあまり変化していない部分では輝度値を直接補間する。その輝度値に、反射・屈折による減衰を与える係数（と、画素の大きさより細かく分割が進んでいる場合にはその面積比）を乗じてフレーム・バッファに加える。

（上記の方法は、影を無視している。影を考慮するためには、光線追跡する前に物体の1部に影を落とす物体を求めておき、2つの物体と光源との組（と、影の位置が正確に分かる時には影の位置）を覚えておく。光線追跡中に光束が影のある物体と当たった時には、影の位置が正確に分かる場合には影の境界で補間計算を変え、正確に分からない場合には交点の位置を補間で求め影が落ちているかを検査する。）

4) 反射・屈折方向の追跡

光束が当たる物体が反射も屈折もしないか、反射・屈折による減衰を与える係数の積が充分小さくなるか、反射・屈折の段数が充分大きくなれば終了する。そうでなければ、反射・屈折各方向に対してこのアルゴリズムを再帰的に適用する。

#### 5. おわりに

PCを利用して光線追跡法の高速度化、エリアシングの防止を行なう方法を提案した。円すいを利用する方法には[3]もあるが、[3]ではアンチ・エリアシングのために円すい光束を使い、円すいと交わる領域で物体の曲率を一定とみなし、複雑なレンズの方程式を解いている。我々は、PCを求めるために円すいを利用し、計算が簡単になるように実際の光束よりも大きな円すいを使った。

#### 参考文献

- [1] T. Whitted, "An Improved Illumination Model for Shaded Display," CACM, 23(6), 343-349.
- [2] P. S. Heckbert and P. Hanrahan, "Beam Tracing Polygonal Objects," Computer Graphics, 18(3), 119-127.
- [3] J. Amanatides, "Ray Tracing with Cones," Computer Graphics, 18(3), 129-135.