レイトレーシング法による光の分散現象の疑似再現

Virtual Presentations of the Dispersion of Light by Ray Tracing Method

森　下　伊三男
Isao Morishita

要旨

3次元物体をコンピュータグラフィックス（CG：Computer Graphics）で2次元平面に表示する方法の中に、Ray Tracing Method（レイトレーシング法）という手法がある。この方法は、2次元平面をカメラの結像面にたとえ、その面にたどり着く全ての光線の経路を計算し、その光線の発生点の情報によって色や明るさを決定していく方法である。この方法では、一般に光の波としての性質を示す現象、すなわち回折や干渉等の現象、また、屈折における分散現象を再現することは極めて困難である。本稿では、それらの現象の中から分散現象をとりあげ、レイトレーシング法を用いたCGソフトであるPOV-RAYを使って分散現象の擬似的な再現方法について検討する。

1．はじめに

3次元の物体を2次元平面上に表現する技法には、様々な方法が存在する。その中で、現実に比較的近い状態で再現できる方法にレイトレーシング法というものがある。この方法は画像を表示する2次元平面をカメラの結像面と考え、カメラのレンズを通して入ってくる光の色彩・明度等を結像面の各点で計算し、それらの点の集まりとして2次元画像を得ようとする方法である。

例えば、結像面のある点での色彩・明度等は次のように決定される。まず、その点に到達する光線を、カメラから逆に被写体の方向へたどっていく。そして、たどり着いた被写体のその場所にどのような光（色や明るさ）が当たっているかが計算され、最終的に、その場所での色彩・明度等が決定されて結像面での点の情報となる。もし、途中の光線の経路に鏡が有ればその場所での物理法則に従って反射し、もし、光が透過できる物体が存在すれば、その物体の光学的な性質に従って強度の減衰や色彩への影響、光線の屈折などが計算される。屈折や反射については、光が粒子であるとして、その経路は最小時間の原理から容易に導かれることから［1］、レイトレーシング法でも容易に再現可能となっている。一方、光の分散現象や光の波としての性質である回折・干渉などの現象は再現が極めて困難となっている。

本稿では、レイトレーシング法では再現することが難しい分散現象を取り上げ、擬似的

に分散現象を再現する方法について試みる。レイトレーシング法を利用したＣＧソフトウェアはいろいろ存在するが、本稿では、広く利用されているフリーソフトウェアであるPOV-RAYを用いることにする［２］。

２．レイトレーシング法によるCG

CGにより画像を生成することは、キャンパスとなるフレームにどのような光の情報を与えるか、ということである。ある物体がフレーム内に描かれるということは、光源からの光が物体にあたり、その物体の属性によって反射される色、反射率、反射形態などが決められ、その物体表面からの光がフレーム内に到達することによって画像要素として情報が与えられることを意味している。しかし、光源からの光線全てを追跡することは計算機資源である計算時間や記憶容量の制限から考えて現実的ではない。そこで、レイトレーシング法では、フレームの所にカメラを置き、フレーム内の各点（ドット）から、逆向きの光線、すなわち、カメラに到達するであろう光線をカメラから実際の光の進行方向とは逆の方向にたどっていく光線を考える。すなわち、光線逆行の原理に基づいた方法である［３］。例えば、その逆向きの光線が物体に当たった場合には、その物体の属性に従って反射光を計算し、反射したその先にどのような光源があるかによってドットの色特性が決定される。このステップをフレーム内の全てのドットについて計算すればＣＧ画像が作成できる。物体が透明であれば、表面での反射と共に、透過した光線についても計算が必要となる。また、反射や透過した光が再び別の物体に当たった場合、そこから更に反射や透過した光線を追跡することになる。

したがって、レイトレーシング法によるＣＧは計算機資源、特に計算時間が多く必要となる。近年までは大型計算機での実行が主であり、パソコン（パーソナルコンピュータ）による実行は極めて困難であった。しかし、1989年にリリースされたPOV-Ray (Persistence of Vision Ray Tracer) により、パソコンで簡単にCG画像を作成することが可能となった［２］。POV-Rayはレイトレーシング法によるレンダリングソフトであり、フリーソフトウェアとして広く普及している。現在、第3.5版がリリースされている。

３．光の分散現象

光はガラスなどの透明な物体に入射したとき、一部は表面で反射され、一部は物体内部に侵入していく。内部に入った光は物体から外部に出る時に、再度内部で一部が反射し一部が外部に出て行く。物体に入射する時あるいは物体から出てくる時、物体の持つ屈折率に従って光の進行方向が変化する。光が屈折するとき、屈折率は光の振動数によって異なり、いわゆるプリズムによる分光現象を見ることができる。例えば、水晶は可視光の範囲で空気に対して 1.5770 から 1.5419（波長 303.4nm ～ 656.3nm の範囲）まで変化する［４］。従って、幅の狭いスリットを通過した白色光がプリズムに入射した時、そこから出てくる光は虹のようないわゆるスペクトルを持った

光の帯となる。ただし、スリット幅が広い場合、分散した光が互いに重なり合い白色光に近くなる。この場合は、光の帯の両端にのみ色が出現し、いわゆる色収差のようにエッジ領域にスペクトルの一部が現れる。

4．レイトレーシング法における分散現象の疑似表示

レイトレーシング法では、光源に至る光の経路をカメラ側からたどっていくため、分散現象を再現することは難しい。そこで、本稿では、分散現象を再現する為の擬似的な方法として、光源を3原色に分け、物体から透過してくる3原色を重ね合わせることにより、透過光に分散現象と同じような表現(スペクトルの再現)をさせる方法を試みた。図式的に示すと図1の様になる。

図1(a)は本来の光の経路であり、スリットを通りプリズムに入射する光はプリズム内で波長による屈折率の違いによって分光し、プリズムから出た光はスクリーンにスペクトルを形成する。図1(b)はレイトレーシング法によって擬似的に分散を再現させる方法である。ここでは、光源を複数用意し、その位置を互いに少しずらしてセットし、プリズムの方向に向ける。光の波長によって屈折率を変えることは困難なため、この場合はどの色の光に対しても同じ屈折率が与えられている。例えば、3原色(RGB)を用意してプリズムの方向に光を当てる。この場合、虹のようなきれいなスペクトルは再現できないが、一見それらしい表現が可能となる。ちなみに、その場合のコードをリスト-1に示し、その実行結果を図2に示す。

このリストでは光をスリットに通すと光度が弱くなりすぎるため、スポットライトで代用し、赤(R)・緑(G)・青(B)の3原色をセットしている。屈折率の値そのものはここでは本質的に影響しないので、プリズムの屈折率としては光学ガラス(フリントF2)の 656.3nm での値(1.615)を用いた。また、空間を見やすくするため、X軸(画面左右方向の軸)、Y軸(画面上下方向の軸)、Z軸(画面奥行き方向の軸)を入れてある。プリズムはZ軸に平行にセットし、光はX軸上でX軸と平行に画面左から右方向に向けて投影されている。ただし、光源は互いにY軸方向に微少にずれた位置にセットさ

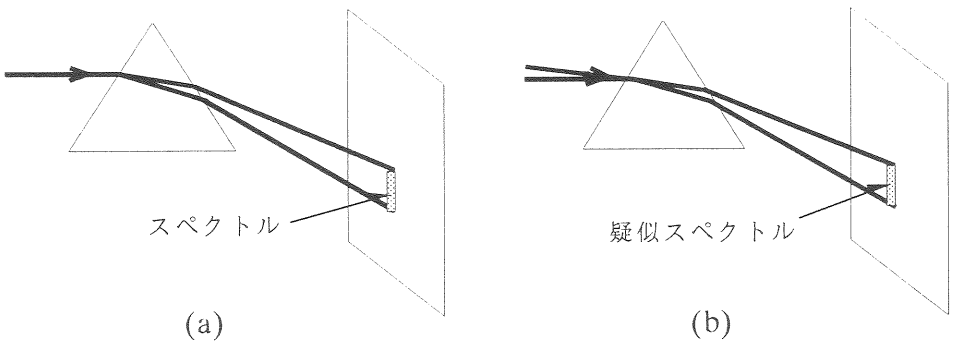

図1　スペクトル生成の比較

```
#include "colors.inc"
#include "shapes.inc"
#include "shapes2.inc"
#include "glass.inc"
#include "lesson.inc"

camera {
   location < -2, 10, -30>
   look_at < 0, 0, 0>
   angle 15
}

light_source { <-10,10,-10> color 0.6*White }

light_source {
   < -8, 0.1, 0 >
   color 50*Red
   spotlight
   point_at < 0, 0, 0>
   radius 1
   falloff 2
}

light_source {
   < -8, 0, 0>
   color 50*Green
   spotlight
   point_at < 0, 0, 0>
   radius 1
   falloff 2
}
//  右上に続く
```

リスト－1　　疑似表現のコードの例

```
light_source {
   < -8, -0.1, 0>
   color 50*Blue
   spotlight
   point_at < 0, 0, 0>
   radius 1
   falloff 2
}

object { JIKU }

prism {
      linear_sweep
      linear_spline
      0, 6, 3, <0,0>, <2,0>, <1,1.615>
      rotate <-90,0,0>
      translate <-2.5,-0.7,3>
      texture {
         T_Glass4
         finish {
            diffuse 0.05
            refraction on
            ior 4.2
         }
      }
   }

object {
   Square_X
   pigment { color White }
   scale 2
   translate <3, 0, 0>
}
```

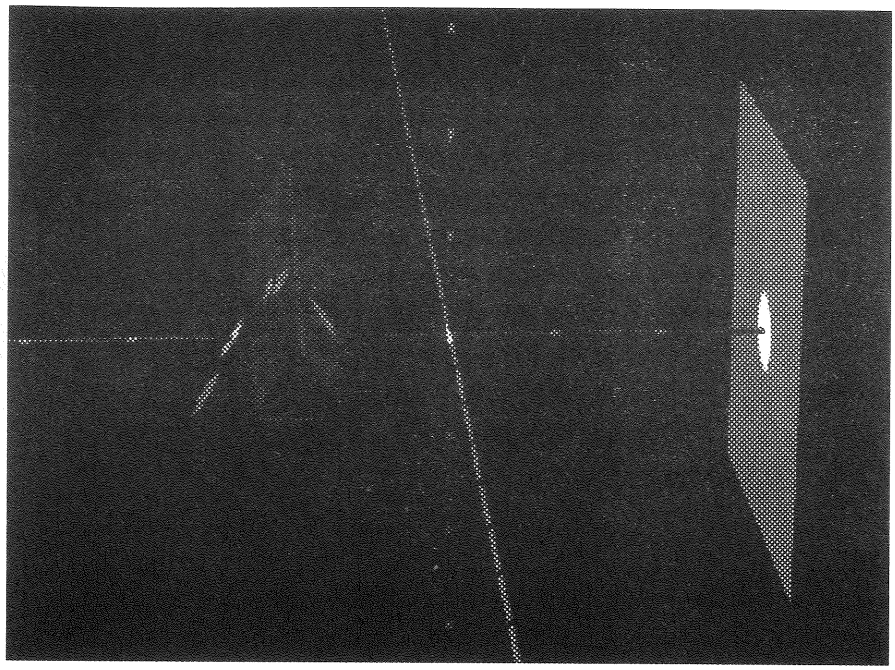

図2　リスト1の実行結果

れている。このリストを実行して作成された画像である図2から分かるように、プリズムの向かって左側の面に当たった光がプリズム内で反射してプリズム表面に明るいスポットとして現れている。また、横軸の線がプリズム内の光の屈折によって図の下の方にずれて見えている。モノクロ画面のため、はっきり分からないが、右側のスクリーンには、使用した3原色のスペクトルが再現されている。ただし、本来、光が屈折してスクリーンに到達すべきであって、スクリーンの中央(横軸の上)にスポットが現れるのは、計算が明らかに間違っている事を示している。先に述べたように横軸がずれて見える事から考えて、プリズムによる光の屈折は確かに再現されている。しかし、スクリーン上への投映の場合、光の屈折が考慮されていない。カメラに到達する光線を計算するとき、直接透過光がカメラに入射する場合には屈折が適用されている。しかし、カメラからまず物体に達し、そこで反射して、その後でプリズムを通過して光源に至る経路の場合、直接光ではなく反射光であるために屈折が計算されない、という設定になっていると考えられる。現在、 POV-Ray の仕様を調査中であり、この点についてはいずれ明らかにしたい。

なお、リストで include したファイルの中で、 colors.inc 、 shapes.inc 、 shapes2.inc 、 glass.inc はいずれもシステムに備わったインクルードファイルであるが、 lesson.inc は軸を描くためのインクルードファイルであり、 object { JIKU }によって軸が描かれている[6]。本稿では、 POV-Ray そのもののコードの記述法などは一切説明していないが、それらについては、公式サイトを参考願いた

い[2]。

5．おわりに

光のさまざまな性質を現す現象のうち、レイトレーシング法によって再現が容易に可能なものは主として屈折と反射であり、擬似的に虹や薄膜干渉の色パターンを表現できるにとどまっている［５］。光の波長に依存して屈折率が異なることによる分散現象や波としての性質が顕著に表れる回折や干渉の現象については、現状では表現が極めて困難となっている。

本稿では、再現が困難な現象の中から、光の分散現象について擬似的に表現する方法を提示した。あくまで擬似的な方法であり、物理法則に完全に従った再現ではない。現実の問題、すなわち計算量、記憶量などの制限から、レイトレーシング法で分散現象を再現することは極めて困難であり、物理法則に全く矛盾無いような表現はほぼ不可能に近い。これは、分散現象に限ったことではなく、回折や干渉も同様である。本稿では、これらの現象については、全く扱っていない。これについては今後の課題としたい。

［参考文献］

［1］Feynman, Leighton and Sands "The Feynman Lectures on Physics, Vol.1", Addison-Wesley Pub., 1963.

［2］POV-Rayの公式サイトは http://www.povray.org/

日本語で書かれた簡単な入門書は、

小室日出樹「POV-Rayではじめるレイトレーシング（改訂二版）」、アスキー出版局、1999．

［3］例えば、吉原邦夫「物理光学」、共立出版、1966.

［4］例えば、国立天文台編「理科年表」、丸善、2003．

［5］例えば、POV-Rayインストールフォルダー内のscenes¥advanced¥diffractr.pov

［6］小室日出樹「POV-Rayで学ぶ実習コンピュータグラフィックス」、アスキー出版局、2000.

森下伊三男　（経営学部情報管理学科教授）