レイトレーシングによるコンピュータグラフィクス入門 - レイと三角形の交差判定 -

金森 由博*

2013 年 4 月 24 日

レイトレーシングでは、各画素から放出されたレイと物体との交点を計算する必要がある。雛形のソースコードにはすでに球と平面について交差判定が実装されている。このレジュメではレイと三角形の交差判定について学ぶ。

1 課題

雛形のプログラムで設定ウィンドウにおいて、Scene タブの Load ボタンでシーン定義ファイル `default.cfg` を読み込むと、プレビュー画面に四面体が表示されるはずである。この結果は 4 つの三角形を OpenGL によって描画している。設定ウィンドウで Execute ボタンを押すと、これらの三角形が消えてしまう。これはなぜかというと、三角形 (`Triangle` クラス) の `hit` 関数および `shadowHit` 関数を実装が実装されておらず、レイトレーシングが行えていないからである。このレジュメおよび球 (`Sphere` クラス) や平面 (`Plane` クラス) の `hit` 関数および `shadowHit` 関数を参考にして、`Triangle` クラス (ソースコードは `Triangle.cpp`) の `hit` 関数および `shadowHit` 関数を実装しよう。うまく実装できれば、OpenGL による表示と同様な結果がレイトレーシングによって表示されるはずである。

このレジュメの課題となる「レイと三角形の交差判定」について述べる前に、参考として、レイと球およびレイと平面の交差判定について説明する。

2 レイと球の交差判定

球は中心点 $\mathbf{c}$ から半径 r だけ離れた点 $\mathbf{x}$ の集合として定義される。

$$\|\mathbf{x} - \mathbf{c}\| = r \tag{1}$$

一方、レイは始点 $\mathbf{o}$ から単位ベクトル $\mathbf{d}$ の方向に伸びる半直線として定義される。レイ上の点 $\mathbf{p}$ は、パラメータ t を用いて次のように書ける。

$$\mathbf{p}(t) = \mathbf{o} + t\,\mathbf{d} \tag{2}$$

ここでパラメータ t は、レイの始点 $\mathbf{o}$ から点 $\mathbf{p}(t)$ までの距離を表しており、$t > 0$ である。

* kanamori@cs.tsukuba.ac.jp

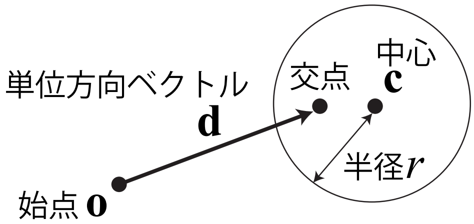

図 1　レイと球の交差判定.

では、レイと球の交点を求めてみよう (図 1)。レイ上の点 $\mathbf{p}(t)$ が球面上にある、ということであるから、式 (1) における点 $\mathbf{x}$ に $\mathbf{p}(t)$ を代入する。

$$\begin{aligned}\|\mathbf{p}(t)-\mathbf{c}\| &= r\\ \|(\mathbf{o}+t\,\mathbf{d})-\mathbf{c}\| &= r\\ \|t\,\mathbf{d}-(\mathbf{c}-\mathbf{o})\| &= r\end{aligned}$$

この式の両辺を 2 乗して、パラメータ t について整理すると、t に関する次のような二次方程式が得られる。

$$t^2-2\,t\,\mathbf{d}\cdot(\mathbf{c}-\mathbf{o})+\|\mathbf{c}-\mathbf{o}\|^2-r^2=0 \tag{3}$$

ここで $\mathbf{d}$ は単位ベクトルなので $\|\mathbf{d}\|=1$ であることに注意してほしい。もし式 (3) が実数解を持ち、しかもそのときの解 t が $t\geq 0$ なら、レイと球は交点を持つ。二次方程式 (3) が実数解を持つかどうか、判別式 D を計算してみよう。

$$D=\{\mathbf{d}\cdot(\mathbf{c}-\mathbf{o})\}^2+\|\mathbf{c}-\mathbf{o}\|^2-r^2 \tag{4}$$

この判別式 D が $D\geq 0$ ならば、二次方程式 (3) が実数解を持つ。$D=0$ ならばレイと球が接している (交点は 1 つだけ) という可能性があり、$D>0$ ならばレイと球は 2 つの交点を持つ可能性がある。実際に交点を持つかどうかは解 t が $t\geq 0$ であるかどうかを確かめる必要がある。$D>0$ のとき解 t は、解の公式より

$$t_1=\mathbf{d}\cdot(\mathbf{c}-\mathbf{o})-\sqrt{D},\qquad t_2=\mathbf{d}\cdot(\mathbf{c}-\mathbf{o})+\sqrt{D} \tag{5}$$

の 2 つとなる。この 2 つのうちのどちらを選べばよいだろうか。レイトレーシングで必要なのは、始点からより近い交点のみである。つまり、$t\geq 0$ でより小さい方を選べばよい。明らかに $t_1<t_2$ であるから、$t_1\geq 0$ なら t_1 が解、$t_1<0$ かつ $t_2\geq 0$ なら t_2 が解、$t_2<0$ なら解なし、つまり交点はないということになる。最終的に得られた解 t をレイの式 (2) に代入すれば、交点 $\mathbf{p}(t)$ が得られる。

交点が求まったら、その後に陰影を計算する必要があり、陰影の計算のために交点での法線ベクトルを求めておく必要がある。球面上の点 $\mathbf{x}$ での法線ベクトル $\mathbf{n}$ は、中心点 $\mathbf{c}$ から点 $\mathbf{x}$ に向かうベクトルを正規化すれば得られる。

$$\mathbf{n}=\frac{\mathbf{x}-\mathbf{c}}{\|\mathbf{x}-\mathbf{c}\|} \tag{6}$$

3　レイと平面の交差判定

平面上の点 $\mathbf{x}$ は、平面の法線ベクトル $\mathbf{n}$ および原点と平面との符号付き距離 d を用いた次の式を満たす。

$$\mathbf{x}\cdot\mathbf{n}=d \tag{7}$$

レイと平面の交点を求めるために、先ほどと同様に、点 $\mathbf{x}$ としてレイ上の点 $\mathbf{p}(t)$ を代入する。

$$\begin{aligned}\mathbf{p}(t)\cdot\mathbf{n} &= d\\ (\mathbf{o}+t\,\mathbf{d})\cdot\mathbf{n} &= d\\ t\,\mathbf{d}\cdot\mathbf{n} &= d-\mathbf{o}\cdot\mathbf{n}\\ t &= \frac{d-\mathbf{o}\cdot\mathbf{n}}{\mathbf{d}\cdot\mathbf{n}}\end{aligned}$$

こうして得られたパラメータ t が $t \geq 0$ であれば、点 $\mathbf{p}(t)$ はレイと平面の交点である。

4　レイと三角形の交差判定

三角形は 3 つの点 $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$ から定義される。これらの 3 頂点が同一直線上になければ平面がただひとつ決まる。CG では物体形状を表現するのに、各面が三角形からなる多面体 (三角形メッシュ) で表現されることが多い。そのため、レイトレーシングにおいてレイと三角形の交差判定は重要である。

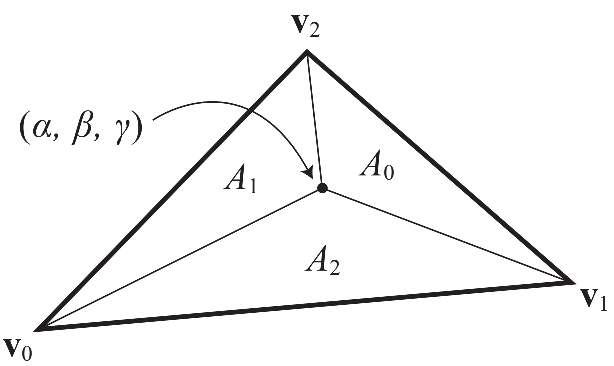

図 2　重心座標.

以下、点の座標は 3 次元の列ベクトルで表現する。3 点 $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$ で定義される平面上の点 $\mathbf{p}$ を表すには、**重心座標 (barycentric coordinate)** がよく使われる。

$$\mathbf{p}(\alpha,\beta,\gamma) = \alpha\mathbf{v}_0 + \beta\mathbf{v}_1 + \gamma\mathbf{v}_2 \tag{8}$$

ただし、点 $\mathbf{p}$ が平面上に存在するための制約として、α, β, γ は

$$\alpha + \beta + \gamma = 1 \tag{9}$$

を満たす。ここで点 $\mathbf{p}$ が三角形の内部にある条件は

$$0 < \alpha < 1, \quad 0 < \beta < 1, \quad 0 < \gamma < 1 \tag{10}$$

である。α, β, γ のうちどれかひとつが 0 であれば点 $\mathbf{p}$ は三角形の辺上、ふたつが 0 であれば頂点上にある。α, β, γ は、図 2 の A_0, A_1, A_2 のそれぞれ面積と三角形の面積 A との比を表す。

$$\alpha = A_0/A, \quad \beta = A_1/A, \quad \gamma = A_2/A \tag{11}$$

これらの式は点 $\mathbf{p}$ が三角形の外側にある場合でも、A_0, A_1, A_2 が符号付き面積であれば成立する。

式 (8) において $\alpha = 1 - \beta - \gamma$ として α を消去すると

$$\mathbf{p}(\beta, \gamma) = \mathbf{v}_0 + \beta(\mathbf{v}_1 - \mathbf{v}_0) + \gamma(\mathbf{v}_2 - \mathbf{v}_0) \tag{12}$$

点 $\mathbf{p}$ が三角形の内部にある条件を書きなおすと

$$\beta + \gamma < 1, \quad 0 < \beta, \quad 0 < \gamma \tag{13}$$

となる。

3 次元空間において、始点 $\mathbf{o}$ から方向 $\mathbf{d}$ に向かうレイ $\mathbf{p}(t) = \mathbf{o} + t\mathbf{d}$ が平面に衝突するとき、次の式が成り立つ。

$$\mathbf{o} + t\mathbf{d} = \mathbf{v}_0 + \beta(\mathbf{v}_1 - \mathbf{v}_0) + \gamma(\mathbf{v}_2 - \mathbf{v}_0) \tag{14}$$

t, β, γ を含む項を左辺にまとめて

$$-t\mathbf{d} + \beta(\mathbf{v}_1 - \mathbf{v}_0) + \gamma(\mathbf{v}_2 - \mathbf{v}_0) = \mathbf{o} - \mathbf{v}_0 \tag{15}$$

$\mathbf{e}_1 = \mathbf{v}_1 - \mathbf{v}_0$, $\mathbf{e}_2 = \mathbf{v}_2 - \mathbf{v}_0$, $\mathbf{r} = \mathbf{o} - \mathbf{v}_0$ とおき、行列の形式にして整理すると

$$(-\mathbf{d},\, \mathbf{e}_1,\, \mathbf{e}_2) \begin{pmatrix} t \\ \beta \\ \gamma \end{pmatrix} = \mathbf{r} \tag{16}$$

ここで $(-\mathbf{d},\, \mathbf{e}_1,\, \mathbf{e}_2)$ は 3×3 行列である。クラメル (Cramer) の公式を用いてこれを解くと

$$\begin{pmatrix} t \\ \beta \\ \gamma \end{pmatrix} = \frac{1}{|-\mathbf{d},\, \mathbf{e}_1,\, \mathbf{e}_2|} \begin{pmatrix} |\mathbf{r},\, \mathbf{e}_1,\, \mathbf{e}_2| \\ |-\mathbf{d},\, \mathbf{r},\, \mathbf{e}_2| \\ |-\mathbf{d},\, \mathbf{e}_1,\, \mathbf{r}| \end{pmatrix} \tag{17}$$

ただし $|\cdot|$ は行列式を表す。一般に、3 次元のベクトル $\mathbf{A}, \mathbf{B}, \mathbf{C}$ に対して $|\mathbf{A},\, \mathbf{B},\, \mathbf{C}| = -(\mathbf{A} \times \mathbf{C}) \cdot \mathbf{B} = -(\mathbf{C} \times \mathbf{B}) \cdot \mathbf{A}$ が成り立つことから

$$\begin{pmatrix} t \\ \beta \\ \gamma \end{pmatrix} = \frac{1}{(\mathbf{d} \times \mathbf{e}_2) \cdot \mathbf{e}_1} \begin{pmatrix} (\mathbf{r} \times \mathbf{e}_1) \cdot \mathbf{e}_2 \\ (\mathbf{d} \times \mathbf{e}_2) \cdot \mathbf{r} \\ (\mathbf{r} \times \mathbf{e}_1) \cdot \mathbf{d} \end{pmatrix} = \frac{1}{\mathbf{u} \cdot \mathbf{e}_1} \begin{pmatrix} \mathbf{v} \cdot \mathbf{e}_2 \\ \mathbf{u} \cdot \mathbf{r} \\ \mathbf{v} \cdot \mathbf{d} \end{pmatrix} \tag{18}$$

ただし、$\mathbf{u} = \mathbf{d} \times \mathbf{e}_2$, $\mathbf{v} = \mathbf{r} \times \mathbf{e}_1$ とおいた。ここで得られた β, γ が式 (13) を満たす場合、レイと三角形は交差する。

三角形の単位法線ベクトル $\mathbf{n}$ は、反時計回りを正の向きとすると次式で与えられる。

$$\mathbf{n} = \frac{(\mathbf{v}_1 - \mathbf{v}_0) \times (\mathbf{v}_2 - \mathbf{v}_0)}{\|(\mathbf{v}_1 - \mathbf{v}_0) \times (\mathbf{v}_2 - \mathbf{v}_0)\|} \tag{19}$$

5 hit 関数と shadowHit 関数

課題のプログラムにおいて、レイと物体との交差判定を行うには、それぞれの物体クラスの `hit` 関数と `shadowHit` 関数を実装すればよい。これらの関数はいずれも、`RecursiveRayTracer::traceRec` 関数で呼び出される。これら 2 つの関数の違いについて説明する。`hit` 関数は、交点を求める際に、あとで陰影の計算に必要な、法線ベクトル、交点の座標、物体の材質なども併せて計算する。これらの情報は `HitRecord` 構造体に格納されている。

HitRecord.h

```
struct HitRecord
{
        MyAlgebra::Real t; // ray parameter (used like o + t * d)
        MyAlgebra::vec3 normal; // surface normal
        MyAlgebra::vec3 hit_pos; // hit position p (= o + t + d)
        MyAlgebra::vec3 tex_coords; // texture coordinate
        Material *material;
};
```

一方、`shadowHit` 関数は、レイと物体の交点が求まったあと、陰影の計算の際に呼び出される。交点から光源の方向にレイを飛ばし、光を遮る物体がないかどうかを調べる。このときに放たれるレイを特に**シャドウレイ** (*shadow ray*) と呼ぶ。`shadowHit` 関数では、シャドウレイが他の物体に当たるかどうかだけが問題で、`HitRecord` 構造体に記述されているような情報は不要である。`shadowHit` 関数は、単に `hit` 関数から `HitRecord` 構造体の計算を除いたものである。

ただし、`shadowHit` 関数についてひとつ注意がある。`hit` 関数で交点が求まったとき、数値誤差の影響で、わずかに交点が物体表面の内側にめり込んだ位置になることがある。この状態で `shadowHit` 関数を呼び出すと、レイが物体の内側からスタートして同じ物体の表面にぶつかり、結果として「光が遮られた」と判定されてしまう。すると光源からの光が届かないことになり、ところどころ不自然な影ができてしまう。これに対する対処法として、シャドウレイの始点を、法線ベクトルに沿って交点を少しだけずらした位置に設定する (図 3)。また、レイのパラメータ t の最小値を設けておき、その最小値よりも小さい場合は交点が見つからなかったものとする。なお、課題のプログラムではこれらの対策は `RecursiveRayTracer::traceRec` 関数の中ですでに実装されているので、自分で実装する必要はない。

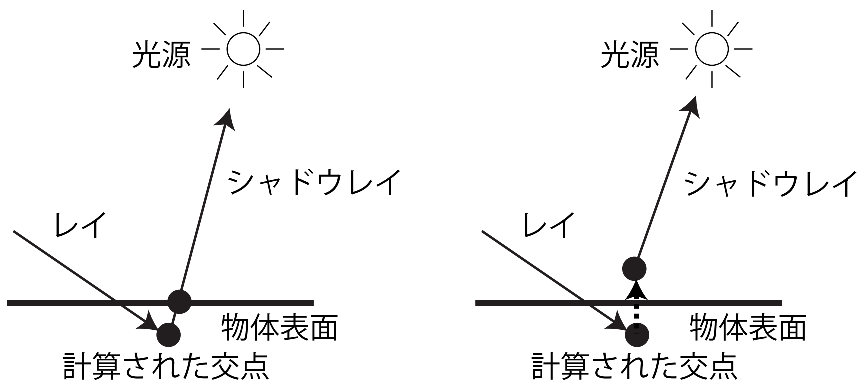

図 3　シャドウレイの始点をずらす前 (左) とずらした後 (右).

6　実装上の注意

視線と物体との交差判定の関数は、プログラムで何回も呼び出されるため、効率的に計算を行う必要がある。再利用できる式は、一度計算したらその式を再利用して計算するようにする。計算途中で交差する条件を

満たさないことが明らかになった場合は、その時点で処理を終了し、「交差しない」という結果を返すようにする。また、ゼロ除算にも注意する必要がある。式 (18), (19) などでは、分母がゼロになる場合がある (どういう場合か考えよ)。ゼロで除算された値を保持する変数は数値としては不正な値が入っているので、その変数を計算に使うとその結果も正しくない結果になってしまう。これを避けるには、変数の絶対値が、適当な小さな正の値より小さいかどうかを調べ、小さい場合はそれに応じた処理を行うようにする。