

ESライブラリ&& ゲームプログラミング

バーテックスシェーダー編 - 第5回 ランバート照明モデル

ランバート照明モデル

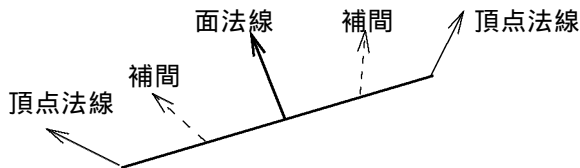
- ・ランバートの余弦則によるライティングをランバート拡散照明と呼ぶ
- ・ランバート照明モデルは、ランバートの余弦則による拡散反射光と環境光によるライティング

概要

環境光演算と拡散反射光演算により、モデルに基本的な陰影処理が施されます。拡散反射で多く使われる計算方法がランバートの余弦則です。この方法では、ライトの向きと頂点の法線から色の強弱を決めますが、ほとんどのモデルは、回転やスケーリングされています。法線もそれらを考慮しなければ、正しい陰影になりません。

法線

3Dでの法線は、面に垂直な方向ベクトルのことをいいます。Direct3Dでは、面を直接定義せず、3つの頂点で三角形を作成し、囲まれた領域を面としています。よって、面の法線も直接指定することができません。頂点ごとに法線を定義し、頂点間の法線は(グローシェーディングモードの場合)、頂点色と同様に補間されるようになっています。そのため、なめらかな陰影がつけられます。



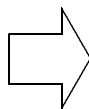
法線のワールド変換

法線も座標同様、ワールド変換しなければ、陰影などで正しい結果が得られません。ワールド変換が回転のみであれば、そのまま「法線×ワールド変換行列」で求めることができます。しかし、ほとんどのモデルは、移動やスケーリングもされています。

法線は方向ベクトルなので位置を持っていません。位置を持たない方向ベクトルにワールド変換行列の移動成分を掛けてしまうと、向きが変わってしまいます。平行光源では、位置が変わっても反射の向きは変わりません。しかし、スケーリングのされ方によっては、法線の向きが変わります。

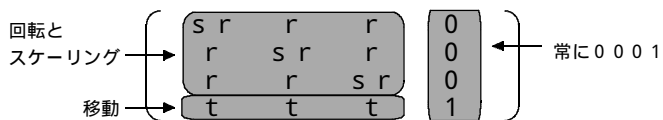
$$\begin{array}{l} \text{方向ベクトル} \\ (1, 1, 1) \end{array} \times \begin{array}{l} (0, 2, 0) \text{へ移動する} \\ \text{ワールド変換行列} \end{array} = \begin{array}{l} \text{方向ベクトルが} \\ (0, 2, 0) \text{になる} \dots \text{大きさも向きも変わる} \end{array}$$

以下のようなモデルを
スケーリングによって変形



反射の方向が変わるので
法線の向きを変えなければ
ならない。また、法線の
長さも変わらないよう
考慮する必要がある

これらを考慮する場合は、「法線×ワールド変換行列」では求めることができません。まず、Direct 3Dのワールド変換行列は、以下のようになっています。



移動成分は4行目にあります。また、4列目は常に(0 0 0 1)となっています。座標は3次元ベクトルのため、4行4列との乗算を行うには、w成分を追加した4次元ベクトルで計算しています。

ここで、回転とスケールリングのみであれば、3次元ベクトルと3行3列の行列で計算することができます。4行目と4列目を無視して計算すれば、回転とスケールリングのみ反映できそうです。しかし、スケールリングの変換行列は、ベクトルを倍にするだけです。方向ベクトルの「大きさ」が変わるだけで、向きは変わりません。大きさが2倍になったからといって、反射の強さも2倍になるわけではないので、スケールリング係数をそのまま掛けても正しい法線になりません。さらに、照明演算での方向ベクトルは、正規化して±1.0にしないとしないため、スケールリング行列をそのまま用いても意味がありません。

正しい結果を得るには、

「ワールド変換行列の逆行列のさらに転置行列」

を法線に掛け、結果を正規化する必要があります。

(ワールド座標の法線 = ローカル座標の法線 × (ワールド変換行列⁻¹)^{接ベクトル}
- 1...逆行列、接ベクトル...転置行列(法線ベクトルに対し、接ベクトルは90度傾いているため)
法線ベクトルと接ベクトルは90度傾いているため互いの内積は0になる、という性質から
= ワールドの法線ベクトル 内積 ワールドの接ベクトル
= ワールドの法線ベクトル 内積 (ローカルの接ベクトル × ワールド変換行列)
= (ワールドの法線ベクトル × ワールド変換行列^{接ベクトル}) 内積 ローカルの接ベクトル
上記を変形すると、「ワールドの法線」は「ローカルの法線 × ワールド行列の逆行列の転置行列」

逆行列・転置行列

ある行列Aに乗算すると単位行列になる行列を行列Aの「逆行列」といいます。「行列」×「逆行列」=「単位行列」ということです。また、行列の行と列を入れ替えたものを転置行列といえます。たとえば、i行j列の成分は、j行i列に入れ替わります。転置行列は、対角線で成分を折り返した行列といえます。

3Dグラフィクスでは、しばしば逆行列や転置行列が必要となります。逆行列を求めるのは非常に難しいのですが、Direct3Dエクステンションライブラリでは、逆行列はD3DXMatrixInverse関数、転置行列はD3DXMatrixTranspose関数で簡単に作成することができます。また、ESライブラリでもMatrix_Invert関数で逆行列、Matrix_Transpose関数で転置行列が作成できます。

ハーフランバートライティング

ランバート拡散照明では、明暗がかなりはっきりと出ます。ライトの方向に対し、法線が90度以上になると、黒になってしまいます。現実世界ではそのようなことはなく、多少なりとも色が見えます。一般的には、間接反射した色が付く、ということでアンビエントによる環境光を付け足し、ある程度色が出るようにしています。

ランバート拡散照明では、ランバートの余弦則を用いてますが、余弦=コサインカーブが急激に変化するため、明暗がはっきりと出ています。そこで考え出されたのがハーフランバートライティング(Half Lamber Lighting)です。

ハーフランバートライティングでは、急激に変化するコサインカーブに細工をして緩やかにし、さらに暗部を持ち上げるようにしています。具体的には、

「ランバートの余弦則のコサインカーブが半分になるように 1/2 を掛け、さらに 1/2 を足して、さらにこれを2乗」

しています。ハーフランバートライティングは、Valve Softwareが'98年の「Half-Life」から採用している照明演算です。計算式を見てもわかるとおり、物理的にはまったく正しくないものですが、「それらしく」見えるため、さまざまなゲームで採用されています。

課 題

前回作成したディフューズ(+アンビエント+エミッシブ)ライティングを変更し、回転・スケーリングが正しく反映された陰影付けがされるようにしましょう。

- ヒント1 : ワールド変換行列の「逆行列の転置行列」が必要になります
- ヒント2 : 上記は、プログラム側でしか作成できません(シェーダーでは逆行列が計算できない)
- ヒント3 : シェーダーのグローバル変数に逆行列の転置行列を格納できるよう、以下を追加します
「float4x4 g_WIT;」
- ヒント4 : "g_WIT"に「ワールド変換行列の逆行列の転置行列」をモデル毎に計算し、渡します
- ヒント5 : シェーダーでは、頂点色計算前に「法線×g_WIT」で法線の向きを計算します
- ヒント6 : 「(float3x3)g_WIT」とすると、3行3列の行列にキャストできます
- ヒント7 : さらに念のため、上記計算後「normalize(法線変数名)」として、法線を正規化します
- ヒント8 : 「光源方向(L)と7で求めた法線(N)の内積(dot)」をランバートの余弦則に用います
- ヒント9 : まとめると、前回作成したプログラムの「法線」に「ワールド変換行列の逆行列の転置行列を掛ける」ようにする、ということです

応用問題 : ハーフランバートライティングを行きましょう

- ヒント1 : まず、ディフューズ色はこれまでどおり計算します
- ヒント2 : ヒント1 × 0.5 ... 1 / 2 を掛ける
- ヒント3 : ヒント2 + 0.5 ... 1 / 2 を足す
- ヒント4 : ヒント3 × ヒント3 ... 2 乗する
- ヒント5 : 以上がハーフランバートライティングでのディフューズ色となります