

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7回 リムライティング

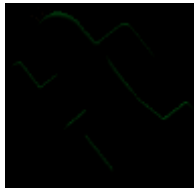
リムライティング

- ・リムライティングは、モデルが逆行を受けたとき、輪郭付近がハイライトになる照明処理
- ・暗いモデルが暗い背景にとけ込まないようにする
- ・モデルの後ろからライトを当て、光源ベクトルと法線ベクトルから計算される

リムライティング

リムライティングは、モデルの後ろからライトを当て、輪郭付近の厚みが薄い部分(光が透過しやすくなる部分)にハイライト処理を施し、輪郭(リム)を強調する照明演算です。

モデルが背景にとけ込みやすい色の場合、この処理を用いると輪郭がやさしく強調され、背景に沈み込むことを防げます。また、「逆光」となった場合に用いれば、より逆光であることを強調できます。



リムライティングなし



リムライティングあり

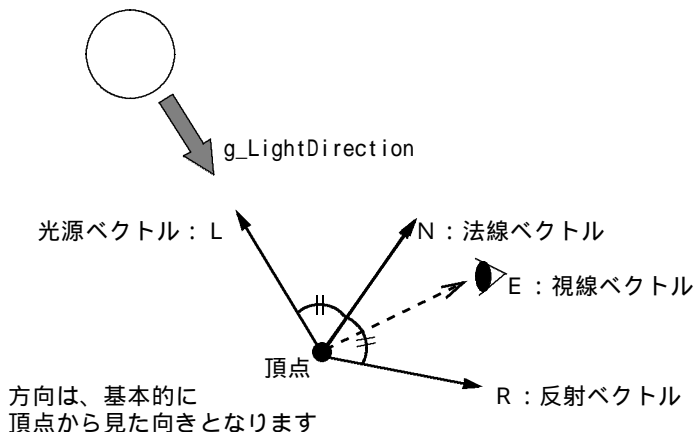
リムライティングは、視線ベクトルと法線ベクトルから、以下の計算式で求めます。

$$f_r k_r (e \cdot r_i)^{k_{rim}}$$

= リム色 * リム強さ * (視線ベクトル 内積 反射ベクトル)^{リムパワー}
= リム色 * リム強さ * (1.0 - (法線ベクトル 内積 視線ベクトル))^{リムパワー}

リム強さ = リムがより強調されます
リムパワー = リムの乗数。小さな数にするとリムの範囲が広がり、大きくすると狭くなります

リムライティングでは、光源に対し垂直方向を向いている頂点ほど明るくします。向きは、視線ベクトルと頂点の法線ベクトルの内積を取ることで調べることができます。カメラの方を向いていない場合、法線は垂直より鈍角になっているので、明るくなります。逆にカメラの方を向いている場合、ライトを後ろから当てるので、暗くなるようにします。ベクトルが正規化されていれば、カメラの方向を向いている場合は内積の値が1になり、カメラの方向を向いていない場合は内積の値は0以下になります。これまでのディフューズやスペキュラーは、内積の値をそのまま使用していましたが、リムライティングでは、ちょうど値が垂直分ずれているので、1から内積を引いた値を使います(下線部)。



同じハイライト処理にスペキュラーがありますが、リムライティングと同時に施すことは少なく、基本的に順光の場合はスペキュラー、逆光の場合はリムライティングを施します。

リムを含めた頂点色 = エミッシブ + アンビエント + ディフューズ + スペキュラー または リム

課題

リムライティングを行うバーテックスシェーダーを作成し、モデルに後ろからライトを当て、リムライティングを行ってみましょう。