

CプログラミングIII (2024)

<STEP 2>

3次元三角形の透視投影と輝度計算

STEP1に引き続いて、以下の処理を行うようにプログラムを拡張せよ.

(想定完成日時=10月29日12:15)(3週)

(レポート提出は不要)

<STEP 2>

三角形の透視投影と輝度計算

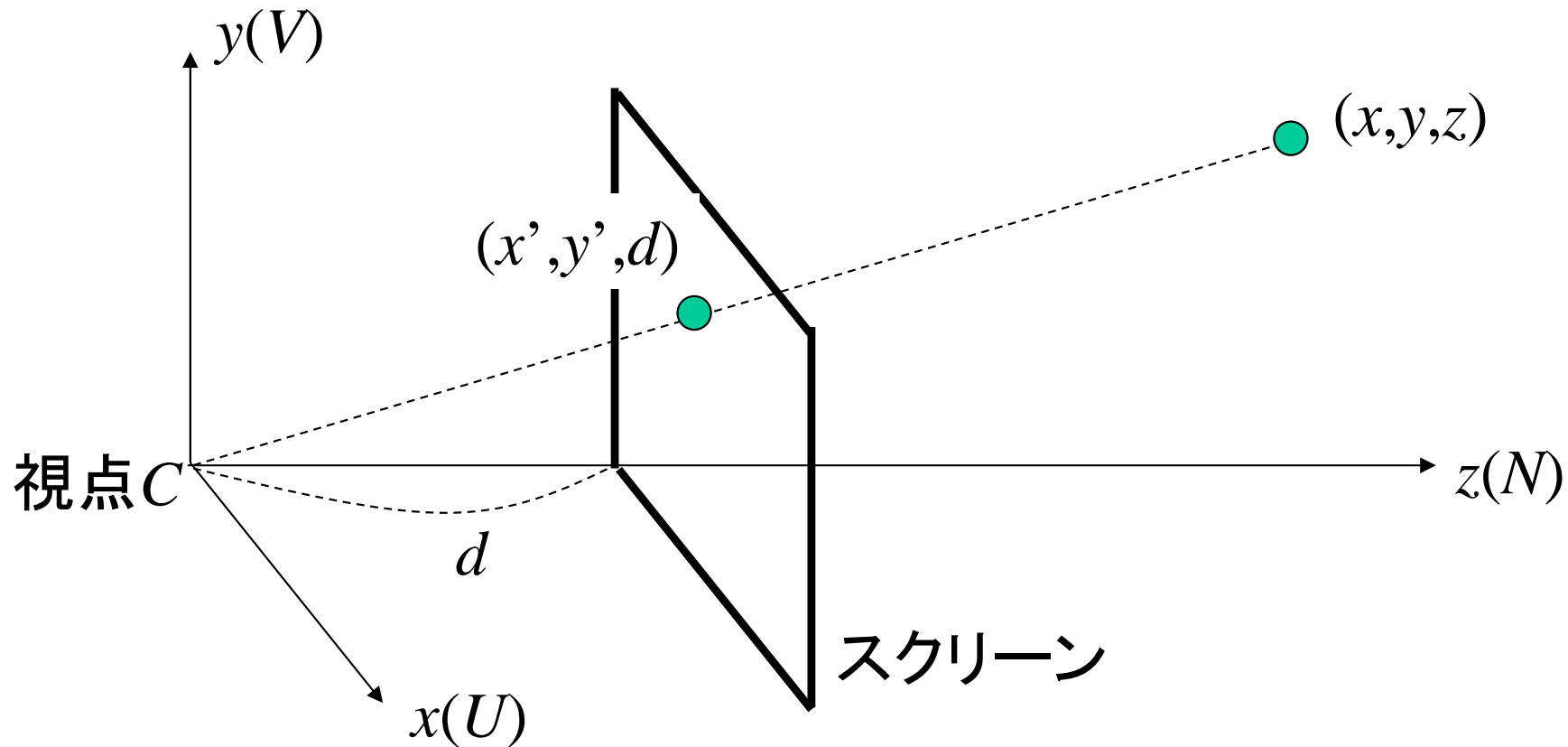
3次元三角形一つの透視投影と画像出力
ただし、以下の項目毎に追加していくと良い。

- 透視投影(視点固定): 左手系. 視点は原点. 視線は z 軸. 焦点距離を指定.
(右手系にする場合は, 視線を z 軸のマイナス方向に.)
- フラットシェーディング: 光源指定(位置と色). 三角形面の元の色を指定. 三角形面の法線ベクトル. 三角形の重心の輝度計算.
- (可能ならスムーズシェーディング)

STEP2対応

以下のスライドは「CG説明」からの抜粋であり、STEP2で主に必要な部分である。

スクリーン座標系(左手系)



カメラ座標系において透視投影を行う

透視投影 $x'=(d/z)x, y'=(d/z)y, z'=d$ ただし, $z>0$

光源が1つの場合

$$\text{輝度 } I = I_a + I_d + I_s$$

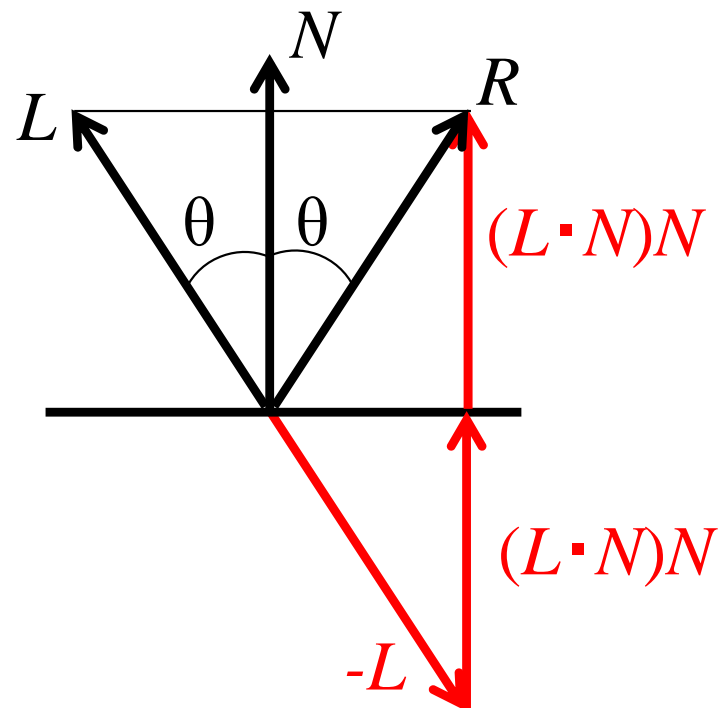
$$= K_a * I_{\text{amb}} + k_d * I_{\text{in}} * \cos\theta + K_s * I_{\text{in}} * \cos^n\alpha$$

$$= K_a * I_{\text{amb}} + K_d * I_{\text{in}} * (L \cdot N) + K_s * I_{\text{in}} * (R \cdot V)^n$$

光源が複数の場合は2, 3項目を光源数分たし合わせる.

正反射ベクトル

$$R = 2(L \cdot N)N - L$$



$$\text{輝度 } I = K_a * I_{\text{amb}} + K_d * I_{\text{in}} * (L \cdot N) + K_s * I_{\text{in}} * (R \cdot V)^n$$

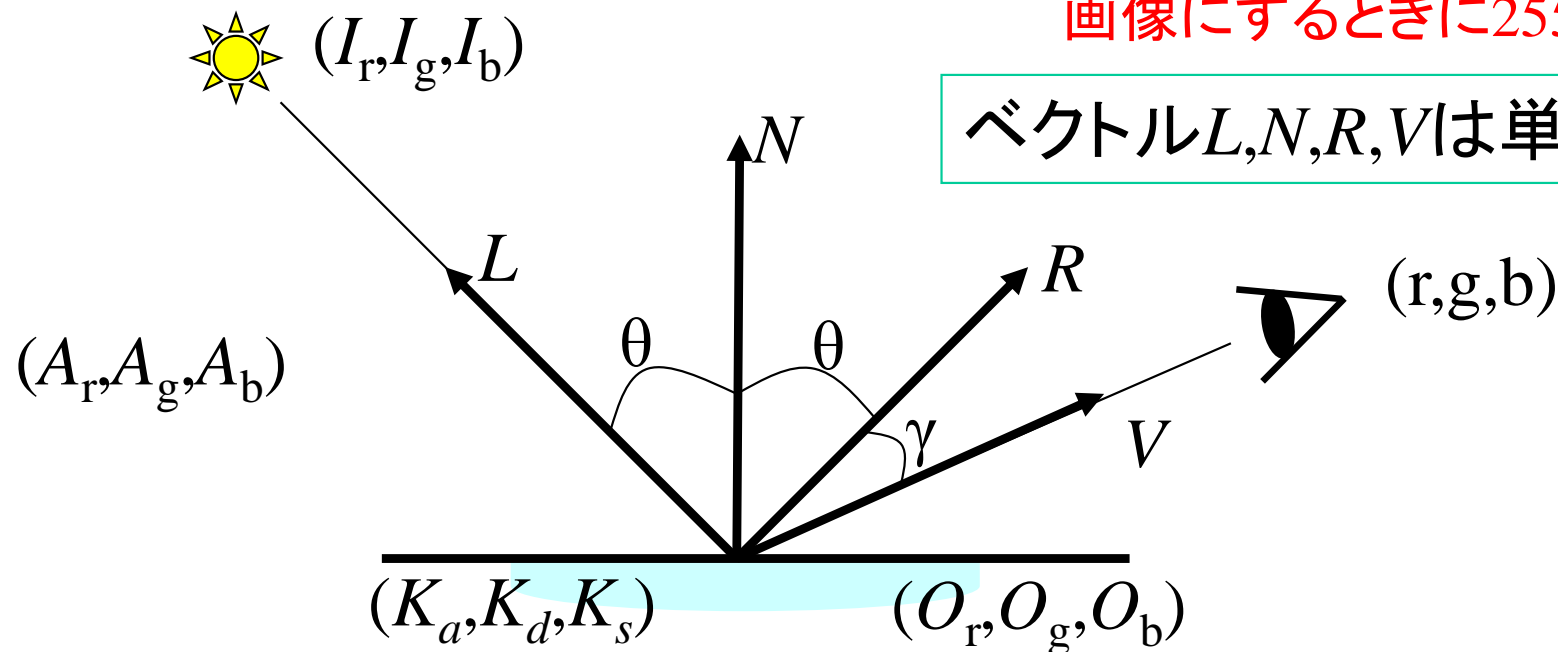
物体の元の色を (O_r, O_g, O_b) , 光源の色を (I_r, I_g, I_b) , 環境光の色を (A_r, A_g, A_b) とすれば, 表示色 (r, g, b) は,

$$r = K_a * O_r * A_r + K_d * O_r * I_r * (L \cdot N) + K_s * I_r * (R \cdot V)^n$$

$$g = K_a * O_g * A_g + K_d * O_g * I_g * (L \cdot N) + K_s * I_g * (R \cdot V)^n$$

$$b = K_a * O_b * A_b + K_d * O_b * I_b * (L \cdot N) + K_s * I_b * (R \cdot V)^n$$

色は $[0.0, 1.0]$ の範囲で計算し,
画像にするとときに255倍する.



ベクトル L, N, R, V は単位ベクトル

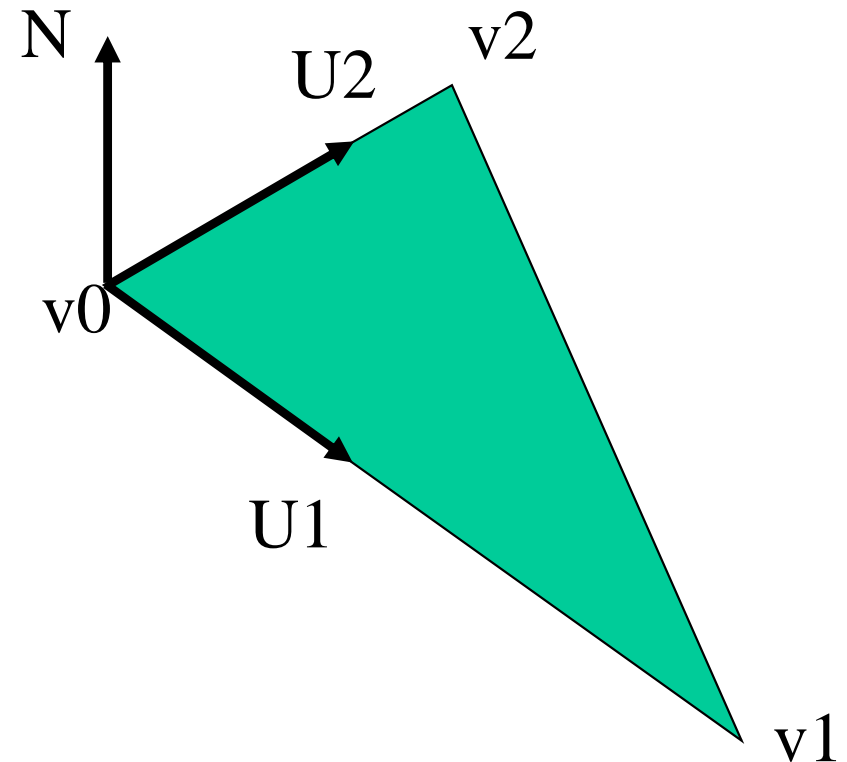
3角形面の法線ベクトル(右手系)

3角形面 $f = (v_0, v_1, v_2)$
に対して,
ベクトル $U_1 = v_1 - v_0$
ベクトル $U_2 = v_2 - v_0$
とすると,
法線ベクトル N は,
 $N = U_1 \times U_2$

$$U_1 = (x_1, y_1, z_1)$$

$$U_2 = (x_2, y_2, z_2)$$

$$U_1 \times U_2 = (y_1 z_2 - z_1 y_2, z_1 x_2 - x_1 z_2, x_1 y_2 - y_1 x_2)$$



実習

次回の解説は11月05日