

mod_SC3D_A 説明

Index

- [一言説明](#)
- [数学的な事前定義](#)
- [カメラや被写体に関する説明](#)
- [命令・関数リファレンス](#)

< 一言説明 >

HSP の標準命令だけで 3D 描画を行う為の座標計算を行うモジュール。

< 依存するモジュール >

`mod_CalcNum`

必ず上記のモジュールを先にインクルードすること。

< 数学的な事前定義 >

あるベクトル $\rightarrow V$ について、「 $\rightarrow V$ まわりの回転」と言うときは、時計の背中に $\rightarrow V$ の尻を鉛直にくっつけた時に時計の針が回る向きを正の回転とする。

角度は、特に断りのない限り**ラジアン**で表現する。

< カメラや被写体に関する説明 >

Index

- [グローバル空間](#)
 - [グローバル座標系](#)
- [カメラ](#)
 - [オイラー角による表現](#)
 - [axis-angle による表現](#)
- [カメラのスクリーンと HSP の画面の関係](#)
- [線分](#)
- [球](#)
- [構造体](#)
 - [カメラ構造体](#)

< グローバル空間 >

< グローバル座標系 >

このモジュールで扱うものの位置を表現するための共通の座標系。

座標は (x_g, y_g, z_g) で表す。

原点は点 0_g で表す。

グローバル座標系における単位ベクトル $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$ をそれぞれ $\rightarrow e_g_x$, $\rightarrow e_g_y$, $\rightarrow e_g_z$ で表す。

基本的に、**地面は x_g, y_g 平面**とし、 z_g 軸が点を指すイメージ。そうでないといけない訳ではないが、**その想定**でこのモジュールは作ってある。

< カメラ >

「眼点」とその背後のスクリーンから成る。これでグローバル空間内のものを見る。

カメラには「カメラ座標系」という独自の座標系があるが、**大きさの尺度はグローバル座標系と同じ**。(すなわち、グローバル座標系での1とカメラ座標系での1は同じ大きさ。)

座標は (x_c, y_c, z_c) で表す。

原点は点 0_c で表す。

カメラ座標系座標が (a, b, c) であるからといってグローバル座標もそうだとは限らないことに留意。カメラが傾いていたりするとグローバル座標とは異なってくる。

カメラ座標系における単位ベクトル $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$ をそれぞれ $\rightarrow e_c_x$, $\rightarrow e_c_y$, $\rightarrow e_c_z$ で表す。

x_c, y_c 平面がスクリーンである。カメラ座標系座標 $(0, 0, 1)$ の位置に「**眼点**」 E_c がある。

よって、**カメラにとっての「前方」は z_c 軸の正の向き**である。

ベクトル $0_c \rightarrow E_c$ を「カメラの視線ベクトル」とする。大きさは1である。

カメラにとっての「**天**」は **y_c 軸の正の向き**にある。

被写点と眼点を結ぶ直線とスクリーンとの交点が「像」である。

視野角は 120° である。

カメラには「カメラの基本状態」というが存在する。これは、以下の条件の連立により定義される。

- カメラ座標系の原点がグローバル座標系の原点と重なる。
- カメラ座標系の x_c, y_c, z_c 軸の正の向きがそれぞれグローバル座標系の x_g, y_g, z_g 軸の正の向きと一致する

カメラの状態は、位置と姿勢で決まる。
位置については、 0_c のグローバル座標を決めればよい。

姿勢は、 $\rightarrow e_{c_x}, \rightarrow e_{c_y}, \rightarrow e_{c_z}$ で定まる。

姿勢を決定する方法は以下の2つを用意する。

- オイラー角による表現
- axis-angle による表現

< オイラー角による表現 >

カメラの基本状態をベースとして、 x_c, y_c, z_c 軸回りに、この順にカメラを回転させる。

x_c, y_c, z_c 軸回りにカメラ座標系における単位ベクトルを回転させればよい。

< axis-angle による表現 >

視線ベクトルを定め、その回りにカメラを回転させる。但し、回転前の基準は、カメラが「最も立っている」状態を0とする。

視線ベクトル回りにカメラ座標系における単位ベクトルを回転させればよい。

ターゲットの点をTとする。

まず $0_c \rightarrow T$ を求め、その長さを1に調節したものを $\rightarrow e_{c_z}$ とする。

次にカメラの頭の向きを決めるために $\rightarrow e_{c_y}$ を決める。これが少し複雑。

(i) $\rightarrow e_{c_z}$ が y_g と平行でない場合

$\rightarrow e_{c_z}$ と $\rightarrow y_{g_z}$ の内積が最大になるようにする。
詳しくは別資料を参照

(ii) $\rightarrow e_{c_z}$ が y_g と平行である場合

$\rightarrow e_{c_y} = \rightarrow e_{g_y}$ とする。

< カメラのスクリーンと HSP の画面との関係 >

(1 [px] の大きさ) ÷ (グローバル座標系での1の大きさ) を「P/G定数 (Pixel/Global 定数)」とする。

よって、スクリーン上で大きさがxの影は、画面では (P/G定数)*x [px] で描画すればよい

カメラの眼点とスクリーンと視野角の関係から、スクリーン上で実際に使用されるのは半径 $\sqrt{3}$ の円である。

個人的には、この円の直径が大体 400px に相当すれば使いやすいと思うので、
 $400 : \sqrt{3} \approx 230 : 1$ より、P/G定数の値は 230 が良いと思う。

スクリーン上の像は実際の被写体の姿勢とは上下左右が逆になっているので、画面の描画するときは補正が必要である。
さらに、コンピュータの画面はy軸が下を向いていることを考慮すると、HSPの画面のx，y軸の正の向きはx_c，y_c軸の正の向きと同じにとればよい。

0_cと画面の中心の座標はユーザーの都合に任せる。

< 線分 >

線分の両端がいつも視野内に来るとは限らない。
一部しか見えないときもある。
そのとき、視野円錐面と線分の共有点の座標が分かれば、その2点で作られる線分を描画すれば、
一部だけ映っているのと同じ事になる。
つまり、クリッピングするということである。
詳しくは別資料を参照。

< 球 >

0_cと球の中心との距離が分かれば、それに応じて円を描けば球のように見える。

< 構造体 >

複数のデータをひとまとめにしたパック。関連のあるデータをまとめて扱う目的で作られている。

< カメラ構造体 >

実数型配列変数。
カメラの設定を行うときにデータの受け渡しに使う。
各要素の意味は以下の通り。

要素	意味
(0), (1), (2)	0_cのグローバル座標
(3)	カメラの姿勢決定手段。 0：オイラー角 1：axis-angle
(4), (5), (6), (7)	姿勢制御用数値 (i) オイラー角の場合 (4), (5), (6)にそれぞれ、x_c軸, y_c軸, z_c軸まわりの回転角度を指定。 回転はこの順に行われることに留意。 (7)は無視。 (ii) axis-angleの場合

	(4), (5), (6)にそれぞれ、カメラが見つめる点をグローバル座標を指定。 (7)に、カメラがその点を見つめた後の、z_c 軸まわりの回転角度を指定。
(8), (9)	0_c の、HSP の画面上での座標。 すなわち、画面上での描画の中心の座標。
(10)	P/G 定数の値。230 が目安。

< 命令・関数リファレンス >

命令・関数一覧

- [MSC3D_A_init](#)
- [MSC3D_A_SetCam](#)
- [MSC3D_A_GetShadow](#)
- [MSC3D_A_GetLineShadow](#)
- [MSC3D_A_GetSphere](#)

命令・関数	説明
MSC3D_A_init	モジュールの初期化
MSC3D_A_SetCam	カメラの設定 [書式] MSC3D_A_SetCam 変数 1 変数 1 : カメラ構造体
MSC3D_A_GetShadow	座標変換。 グローバル空間内のある点 A のスクリーンに映る像の情報を取得する。 [書式] MSC3D_A_GetShadow P1, P2, P3 , 変数 1, P4 P1, P2, P3 : A のグローバル座標 P4 : 視野角チェック停止フラグ A が視野内にあるかどうかのチェックを行わずに、映るものとして計算するフラグ。 (0, 1)=(チェックを行う, 行わない) 通常は 0 を指定する。

もし 1 を指定して、かつ A が眼点の後ろ側にある等して絶対に映らない場合は、計算がめちゃくちゃになり、エラーが起こる可能性が高い。

1 を指定するのは、視野円錐スレスレのところにある点を確実に映したい等、特別な事情がある場合のみ。

変数 1：結果を格納する実数型配列変数

+++++++ (変数 1 の構造) ++++++

要素	説明
(0), (1), (2)	像のグローバル座標
(3), (4)	像のカメラ座標系座標
(5), (6)	像の HSP の画面上での座標

A がスクリーンに映る場合は stat の値が 1 になり、映らない場合は 0 になる。

MSC3D_A_GetLineShadow

座標変換。
グローバル空間内のある線分 AB がカメラに写るかどうか、また、映るならその像の情報を取得する。

[書式]

MSC3D_A_GetLineShadow P1, P2, P3 , P4, P5, P6 ,
変数 1

P1, P2, P3：A の座標
P4, P5, P6：B の座標
変数 1：結果を格納する実数型配列変数

+++++++ (変数 1 の構造) ++++++

要素	説明
(0)	クリッピングが行われたかどうか。 0：行われなかった 1：行われた
(1), (2), (3) (4), (5), (6)	クリッピングが行われた場合に記録される。 クリッピング後の線分の端点のグローバル座標。 (1), (2), (3) が 1 つ目の端点。 (4), (5), (6) が 2 つ目の端点。
(7), (8), (9) (10), (11), (12)	線分の像の端点のグローバル座標。 (7), (8), (9) が 1 つ目の端点。 (10), (11), (12) が 2 つ目の端

		点。
	(13), (14), (15), (16)	線分の像の端点のカメラ座標系座標。 (13), (14)が1つ目の端点。 (15), (16)が2つ目の端点。
	(17), (18) (19), (20)	線分の像のHSPの画面上での座標。 (17), (18)が1つ目の端点。 (19), (20)が2つ目の端点。
ABがスクリーンに写る場合は stat の値が1になり、映らない場合は0になる。		

MSC3D_A_GetSphere	座標変換。 グローバル空間内のある球 S の、スクリーン上の像の情報を取得する。										
	ただし、S の中心がスクリーンに映らない場合は S は映らないものとして判断される。										
	[書式]										
	MSC3D_A_GetSphere P1, P2, P3, P4 , 変数 1										
	P1, P2, P3 : の中心のグローバル座標 P4 : 半径 変数 1 : 結果を格納する実数型配列変数										
	+++++++ (変数 1 の構造) ++++++										
	<table><tr><th>要素</th><th>説明</th></tr><tr><td>(0), (1), (2)</td><td>像の中心のグローバル座標</td></tr><tr><td>(3), (4)</td><td>像の中心のカメラ座標系座標</td></tr><tr><td>(5), (6)</td><td>像の中心の HSP の画面上での座標</td></tr><tr><td>(7)</td><td>像 (=円) の HSP の画面上での半径 (px)</td></tr></table>	要素	説明	(0), (1), (2)	像の中心のグローバル座標	(3), (4)	像の中心のカメラ座標系座標	(5), (6)	像の中心の HSP の画面上での座標	(7)	像 (=円) の HSP の画面上での半径 (px)
	要素	説明									
	(0), (1), (2)	像の中心のグローバル座標									
	(3), (4)	像の中心のカメラ座標系座標									
(5), (6)	像の中心の HSP の画面上での座標										
(7)	像 (=円) の HSP の画面上での半径 (px)										
S の中心がスクリーンに映る場合は stat の値が1になり、映らない場合は0になる。											