

3次元幾何計算(3次元キャリパー)

(Ver.1.1)

2013 年 6 月

株式会社 アイディール

目次

eyemClp3dDistanceTwoPoints	1
機能 2点間の距離	1
eyemClp3dCenterTwoPoints.....	2
機能 2点の中点座標.....	2
eyemClp3dLineTwoPoints.....	3
機能 2点を通る直線.....	3
eyemClp3dFootOfPerpendicularToLine.....	4
機能 点から直線へ下した垂線の足.....	4
eyemClp3dVerticalLinePointAndLine	5
機能 指定点を通り, 指定直線に垂直な直線	5
eyemClp3dDistancePointToLine	6
機能 点と直線の距離.....	6
eyemClp3dDistanceTwoLines	7
機能 2直線の最短距離	7
eyemClp3dPlaneThreePoints.....	8
機能 3点を通る平面	8
eyemClp3dVerticalPlanePointAndVector	9
機能 指定点を通り, 指定ベクトルに垂直な平面.....	9
eyemClp3dVerticalLinePointAndPlane	10
機能 指定点を通り, 指定平面に垂直な直線	10
eyemClp3dDistancePointToPlane	11
機能 点と平面の距離.....	11
eyemClp3dIntersectionLineAndPlane	12
機能 直線と平面の交点	12
eyemClp3dAngleLineAndPlane	13
機能 直線と平面の交角	13
eyemClp3dFootOfPerpendicularToPlane.....	14
機能 点から平面へ下した垂線の足.....	14
eyemClp3dIntersectionTwoPlanes	15
機能 2平面の交線	15
eyemClp3dAngleTwoPlanes.....	16
機能 2平面の交角	16

eyemClp3dDistanceTwoPoints

機 能	2点間の距離	
形 式	<pre>#include "eyemLib.h" void eyemClp3dDistanceTwoPoints (EyemOcsDXYZ *tpPoint1, EyemOcsDXYZ *tpPoint2, double *dpDist);</pre>	
解 説	2点 P_1 , P_2 の距離 (ユークリッド距離) を求めます.	
引 数	*tpPoint1	点 P_1 です.
	*tpPoint2	点 P_2 です.
	*dpDist	2点 P_1 , P_2 の距離が格納されます.
戻り値	ありません.	
留意事項	特にありません.	

eyemClp3dCenterTwoPoints

機 能 2点の中点座標

形 式 `#include "eyemLib.h"`
`void eyedClp3dCenterTwoPoints (EyemOcsDXYZ *tpPoint1,`
`EyemOcsDXYZ *tpPoint2, EyemOcsDXYZ *tpCenter);`

解 説 2点 P_1 , P_2 の中点座標を求めます.

引 数 `*tpPoint1` 点 P_1 です.
`*tpPoint2` 点 P_2 です.
`*tpCenter` 2点 P_1 , P_2 の中点座標が格納されます.

戻り値 ありません.

留意事項 特にありません.

eyemClp3dLineTwoPoints

機 能 2点を通る直線

形 式 `#include "eyemLib.h"`
`int eyedClp3dLineTwoPoints (EyemOcsDXYZ *tpPoint1, EyemOcsDXYZ *tpPoint2,`
`EyemOcsDPV *tpLine);`

解 説 2点 P_1 , P_2 を通る直線 (ベクトル方程式: $\mathbf{x} = \mathbf{p} + t\mathbf{v}$) を求めます.

引 数 `*tpPoint1` 点 P_1 です.
`*tpPoint2` 点 P_2 です.
`*tpLine` 2点 P_1 , P_2 を通る直線のパラメータ (直線上の1点 \mathbf{p} と方向ベクトル \mathbf{v}) が格納されます. なお, 方向ベクトル \mathbf{v} は単位ベクトルとなっています.

戻り値 エラー報告です.

FUNC_OK	正常終了
FUNC_CANNOT_CALC	計算が不可 (2点 P_1, P_2 が一致)

留意事項 特にありません.

eyemClp3dFootOfPerpendicularToLine

機 能 点から直線へ下した垂線の足

形 式

```
#include "eyemLib.h"

int      eyemClp3dFootOfPerpendicularToLine ( EyemOcsDXYZ *tpPoint,
                                              EyemOcsDPV *tpLine, EyemOcsDXYZ *tpFoot );
```

解 説 点 P から直線 L へ下した垂線の足 (交点座標) を求めます.

引 数

*tpPoint	点 P です.
*tpLine	直線 L です.
*tpFoot	点 P から直線 L へ下した垂線の足の座標が格納されます.

戻り値 エラー報告です.

FUNC_OK	正常終了
FUNC_CANNOT_CALC	計算が不可 (直線 L が直線でない)

留意事項 特にありません.

eyemClp3dVerticalLinePointAndLine

機 能 指定点を通り, 指定直線に垂直な直線

形 式

```
#include "eyemLib.h"

int      eyemClp3dVerticalLinePointAndLine ( EyemOcsDXYZ *tpPoint,
                                              EyemOcsDPV *tpLine, EyemOcsDPV *tpVertical );
```

解 説 指定された点 P を通り, 指定された直線 L に垂直な直線 (ベクトル方程式: $\mathbf{x} = \mathbf{p} + t\mathbf{v}$) を求めます.

引 数

*tpPoint	点 P です.
*tpLine	直線 L です.
*tpVertical	点 P を通り, 直線 L に垂直な直線のパラメータ (直線上の1点 \mathbf{p} と方向ベクトル \mathbf{v}) が格納されます. なお, 方向ベクトル \mathbf{v} は単位ベクトルとなっています.

戻り値 エラー報告です.

FUNC_OK	正常終了
FUNC_CANNOT_CALC	計算が不可 (直線 L が直線でない)

留意事項 特にありません.

eyemClp3dDistancePointToLine

機 能 点と直線の距離

形 式

```
#include "eyemLib.h"

int      eyemClp3dDistancePointToLine ( EyemOcsDXYZ *tpPoint,
                                          EyemOcsDPV *tpLine, double *dpDist );
```

解 説 点 P と直線 L の距離(ユークリッド距離)を求めます.

引 数

*tpPoint	点 P です.
*tpLine	直線 L です.
*dpDist	点 P と直線 L の距離が格納されます.

戻り値 エラー報告です.

FUNC_OK	正常終了
FUNC_CANNOT_CALC	計算が不可(直線 L が直線でない)

留意事項 特にありません.

機 能	2直線の最短距離				
形 式	<pre>#include "eyemLib.h" int eyemClp3dDistanceTwoLines (EyemOcsDPV *tpLine1, EyemOcsDPV *tpLine2, double *dpDist);</pre>				
解 説	2直線 L_1 , L_2 の最短距離を求めます.				
引 数	<p>*tpLine1 直線 L_1 です.</p> <p>*tpLine2 直線 L_2 です.</p> <p>*tpPoint 2直線 L_1, L_2 の最短距離が格納されます.</p>				
戻り値	<p>エラー報告です.</p> <table border="1"> <tr> <td>FUNC_OK</td><td>正常終了</td></tr> <tr> <td>FUNC_CANNOT_CALC</td><td>計算が不可(直線 L_1, L_2 が直線でない)</td></tr> </table>	FUNC_OK	正常終了	FUNC_CANNOT_CALC	計算が不可(直線 L_1 , L_2 が直線でない)
FUNC_OK	正常終了				
FUNC_CANNOT_CALC	計算が不可(直線 L_1 , L_2 が直線でない)				
留意事項	特にありません.				

eyemClp3dPlaneThreePoints

機 能 3点を通る平面

形 式 `#include "eyemLib.h"`
`int eyedClp3dPlaneThreePoints (EyemOcsDXYZ *tpPoint1,`
`EyemOcsDXYZ *tpPoint2, EyemOcsDXYZ *tpPoint3,`
`EyemOcsDABCD *tpPlane);`

解 説 3点 P_1 , P_2 , P_3 を通る平面 $ax + by + cz + d = 0$ を求めます. なお, 3点は三角形を作る位置になければいけません.

引 数 `*tpPoint1` 点 P_1 です.
`*tpPoint2` 点 P_2 です.
`*tpPoint3` 点 P_3 です.
`*tpPlane` 3点 P_1 , P_2 , P_3 を通る平面 $ax + by + cz + d = 0$ の係数 a , b , c , d が格納されます. なお, 法線ベクトル (a, b, c) は単位ベクトルとなっています.

戻り値 エラー報告です.

FUNC_OK	正常終了
FUNC_CANNOT_CALC	計算が不可 (3点が三角形を作らない)

留意事項 特にありません.

eyemClp3dVerticalPlanePointAndVector

機 能 指定点を通り, 指定ベクトルに垂直な平面

形 式

```
#include "eyemLib.h"

int      eyemClp3dVerticalPlanePointAndVector ( EyemOcsDXYZ *tpPoint,
                                                EyemOcsDXYZ *tpVector, EyemOcsDABCD *tpPlane );
```

解 説 指定された点 P を通り, 指定されたベクトル \mathbf{v} に垂直な平面 $ax + by + cz + d = 0$ を求めます.

引 数

*tpPoint	点 P です.
*tpVector	ベクトル \mathbf{v} です.
*tpPlane	点 P を通り, ベクトル \mathbf{v} に垂直な平面 $ax + by + cz + d = 0$ の係数 a , b , c , d が格納されます. なお, 法線ベクトル (a, b, c) は単位ベクトルとなっています.

戻り値 エラー報告です.

FUNC_OK	正常終了
FUNC_CANNOT_CALC	計算が不可 (ベクトル \mathbf{v} がゼロベクトル)

留意事項 特にありません.

eyemClp3dVerticalLinePointAndPlane

機 能 指定点を通り, 指定平面に垂直な直線

形 式

```
#include "eyemLib.h"

int      eyemClp3dVerticalLinePointAndPlane ( EyemOcsDXYZ *tpPoint,
                                              EyemOcsDABCD *tpPlane, EyemOcsDPV *tpLine );
```

解 説 指定された点 P を通り, 指定された平面 α に垂直な直線 (ベクトル方程式: $\mathbf{x} = \mathbf{p} + t\mathbf{v}$) を求めます.

引 数

*tpPoint	点 P です.
*tpPlane	平面 α です.
*tpLine	指定された点 P を通り, 指定された平面 α に垂直な直線のパラメータ (直線上の1点 \mathbf{p} と方向ベクトル \mathbf{v}) が格納されます. なお, 方向ベクトル \mathbf{v} は単位ベクトルとなっています.

戻り値 エラー報告です.

FUNC_OK	正常終了
FUNC_CANNOT_CALC	計算が不可 (平面 α が平面でない)

留意事項 特にありません.

eyemClp3dDistancePointToPlane

機 能 点と平面の距離

形 式 `#include "eyemLib.h"`
`int eyedClp3dDistancePointToPlane (EyemOcsDXYZ *tpPoint,`
`EyemOcsDABCD *tpPlane, double *dpDist);`

解 説 点 P と平面 α の距離を求めます.

引 数 `*tpPoint` 点 P です.
`*tpPlane` 平面 α です.
`*dpDist` 点 P と平面 α の距離が格納されます.

戻り値 エラー報告です.

FUNC_OK	正常終了
FUNC_CANNOT_CALC	計算が不可 (平面 α が平面でない)

留意事項 特にありません.

eyemClp3dIntersectionLineAndPlane

機 能 直線と平面の交点

形 式 `#include "eyemLib.h"`
`int eyedClp3dIntersectionLineAndPlane (EyemOcsDPV *tpLine,`
`EyemOcsDABCD *tpPlane, EyemOcsDXYZ *tpIntersect);`

解 説 直線 L と平面 α の交点座標を求めます.

引 数 `*tpLine` 直線 L です.
`*tpPlane` 平面 α です.
`*tpIntersect` 直線 L と平面 α の交点座標が格納されます.

戻り値 エラー報告です.

FUNC_OK	正常終了
FUNC_CANNOT_CALC	計算が不可 (直線 L が直線でない, または平面 α が平面でない, または L と α が平行)

留意事項 特にありません.

eyemClp3dAngleLineAndPlane

機 能 直線と平面の交角

形 式 `#include "eyemLib.h"`
`int eyedClp3dAngleLineAndPlane (EyemOcsDPV *tpLine,`
`EyemOcsDABCD *tpPlane, double *dpAngle);`

解 説 直線 L と平面 α の交角 (小さい方の角) を求めます.

引 数 `*tpLine` 直線 L です.
`*tpPlane` 平面 α です.
`*dpAngle` 直線 L と平面 α の交角 (小さい方の角:rad単位) が格納されます.

戻り値 エラー報告です.

FUNC_OK	正常終了
FUNC_CANNOT_CALC	計算が不可 (直線 L が直線でない, または平面 α が平面でない, または L と α が平行)

留意事項 特にありません.

eyemClp3dFootOfPerpendicularToPlane

機 能 点から平面へ下した垂線の足

形 式

```
#include "eyemLib.h"

int      eyemClp3dFootOfPerpendicularToPlane ( EyemOcsDXYZ *tpPoint,
                                                EyemOcsDABCD *tpPlane, EyemOcsDXYZ *tpFoot );
```

解 説 点 P から平面 α へ下した垂線の足(交点座標)を求めます.

引 数

*tpPoint	点 P です.
*tpPlane	平面 α です.
*tpFoot	点 P から平面 α へ下した垂線の足の座標が格納されます.

戻り値 エラー報告です.

FUNC_OK	正常終了
FUNC_CANNOT_CALC	計算が不可(平面 α が平面でない)

留意事項 特にありません.

eyemClp3dIntersectionTwoPlanes

機 能 2平面の交線

形 式 `#include "eyemLib.h"`
`int eyedClp3dIntersectionTwoPlanes (EyemOcsDABCD *tpPlane1,`
`EyemOcsDABCD *tpPlane2, EyemOcsDPV *tpLine);`

解 説 平面 α_1 , α_2 の交線(ベクトル方程式: $\mathbf{x} = \mathbf{p} + t\mathbf{v}$)を求めます.

引 数 `*tpPlane1` 平面 α_1 です.
`*tpPlane2` 平面 α_2 です.
`*tpLine` 平面 α_1 , α_2 の交線のパラメータ(直線上の1点 \mathbf{p} と方向ベクトル \mathbf{v}) が
 格納されます. なお, 方向ベクトル \mathbf{v} は単位ベクトルとなっています.

戻り値 エラー報告です.

FUNC_OK	正常終了
FUNC_CANNOT_CALC	計算が不可(平面 α_1, α_2 が平面でない, または2平面が平行)

留意事項 特にありません.

eyemClp3dAngleTwoPlanes

機 能 2平面の交角

形 式 `#include "eyemLib.h"`
`int eyedClp3dAngleTwoPlanes (EyemOcsDABCD *tpPlane1,`
`EyemOcsDABCD *tpPlane2, double *dpAngle);`

解 説 平面 α_1 , α_2 の交角 (小さい方の角) を求めます.

引 数 `*tpPlane1` 平面 α_1 です.
`*tpPlane2` 平面 α_2 です.
`*dpAngle` 平面 α_1 , α_2 の交角 (小さい方の角 : rad単位) が格納されます.

戻り値 エラー報告です.

FUNC_OK	正常終了
FUNC_CANNOT_CALC	計算が不可 (平面 α_1 , α_2 が平面でない, または2平面が平行)

留意事項 特にありません.

改訂履歴

Version No.	内 容
1.0	• 新規発行
1.1	• 項目順の入れ替え