

光切断3次元計測ライブラリ

(Ver.2.3)

2018年8月

株式会社 アイディール

目次

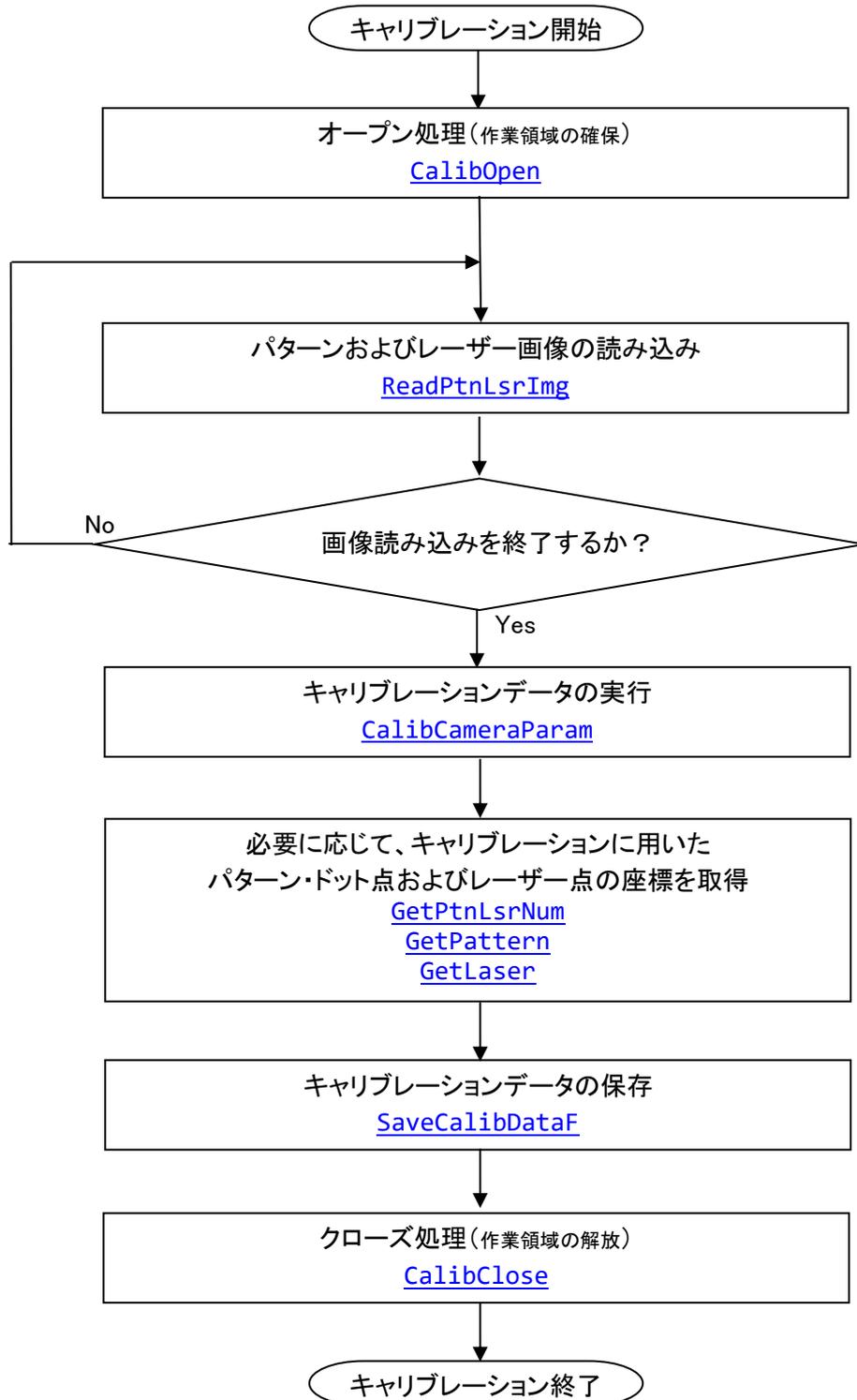
目次.....	i
1. 光切断3次元計測ライブラリの使用手順.....	1
(1) キャリブレーションフェーズ.....	1
(2) 計測フェーズ.....	2
2. ライブラリの詳細説明.....	2
CalibOpen.....	3
機能 キャリブレーションのオープン処理.....	3
SaveCalibDataF.....	5
機能 キャリブレーションデータのファイル保存.....	5
LoadCalibDataF.....	6
機能 計測のオープン処理.....	6
CalibClose.....	7
機能 クローズ処理.....	7
ReadPtnLsrImg.....	8
機能 パターン・ドットとパターン上レーザー線の座標点の読み込み.....	8
CalibCameraParam.....	10
機能 カメラキャリブレーションの実行.....	10
Measure.....	11
機能 レーザー線画像座標から3次元座標への変換.....	11
GetMeasureResult.....	13
機能 レーザー線の3次元座標の取得.....	13
ChangeLsr2Dto3D.....	14
機能 画像座標点から3次元座標点への変換.....	14
Change3Dto2D.....	15
機能 3次元座標点から画像座標点への変換.....	15
GetHomography.....	16
機能 ホモグラフィ行列の取得.....	16
GetCamIntParam.....	17
機能 カメラ内部パラメータの取得.....	17
GetCamExtParam.....	19
機能 カメラ外部パラメータの取得.....	19
GetPtnLsrNum.....	20
機能 パターン・ドット点およびパターン上レーザー線の画像座標データ個数の取得.....	20
GetPattern.....	21
機能 パターン・ドット点の画像座標データの取得.....	21
GetLaser.....	22
機能 パターン上レーザー線の画像座標データの取得.....	22
ShiftLsr3D.....	23

機能	レーザー線画像点列の平行移動	23
	CalcEquipitchProfile	25
機能	レーザー線の等ピッチ 3次元座標の計算	25
	GetEquipitchProfile	27
機能	レーザー線の等ピッチ 3次元座標の取得	27
Appendix1.	キャリブレーション治具情報構造体 EyemLsmJig の設定	28
Appendix2.	ワールド座標系	30
Appendix3.	キャリブレーションの方法	31

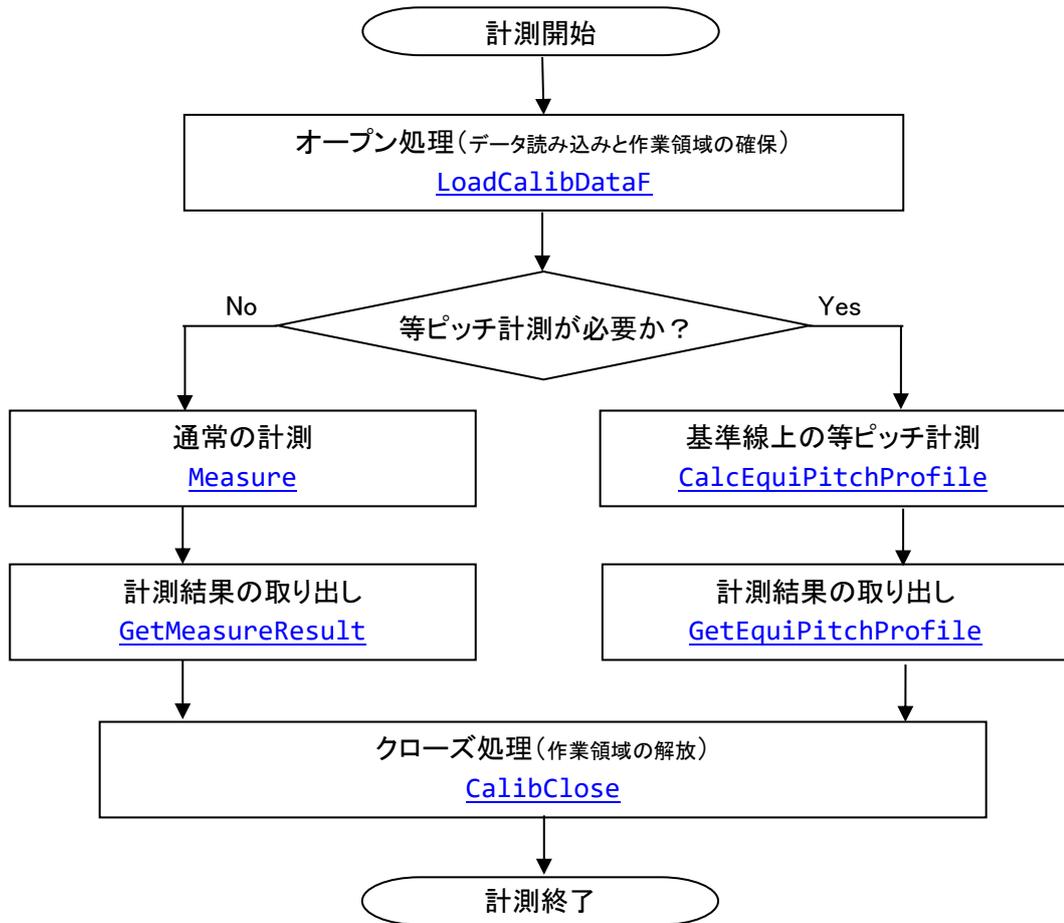
1. 光切断3次元計測ライブラリの使用手順

光切断3次元計測ライブラリでは、キャリブレーションフェーズと計測フェーズがあります。
これらの使用手順は以下の流れとなります。

(1) キャリブレーションフェーズ



(2) 計測フェーズ



2. ライブラリの詳細説明

次ページ以降に各々のライブラリの説明を行います。

CalibOpen

機能 キャリブレーションのオープン処理

クラス EyemSharpLsm.eyemLsm

形式 using EyemSharp;
using EyemSharpLsm;

```
int CalibOpen(int iSetNum, ref EyemLsmJig tpJig, ref EyemLsmCam tpCam, int  
iWldFlag, ref IntPtr vpCalib)
```

解説 作業領域の確保を行い、キャリブレーション・ディスクリプタを生成します。

引数

入力	iSetNum	キャリブレーションパターンとレーザーの画像セットの組数 3組以上で指定します。												
入力	tpJig	キャリブレーション治具情報 詳細は Appendix1 をご参照下さい。												
入力	tpCam	カメラ情報 <table border="1"><tr><td>iCam</td><td>カメラ種別。以下のいずれかを指定します。(※1)</td></tr><tr><td>iWidthL</td><td>カメラの画像メモリX方向サイズ(画素)</td></tr><tr><td>iHeightL</td><td>カメラの画像メモリY方向サイズ(画素)</td></tr></table> (※1) <table border="1"><thead><tr><th>値</th><th>意味</th></tr></thead><tbody><tr><td>0</td><td>通常のカメラ</td></tr><tr><td>1</td><td>シャインプルーフ構成のカメラ</td></tr></tbody></table>	iCam	カメラ種別。以下のいずれかを指定します。(※1)	iWidthL	カメラの画像メモリX方向サイズ(画素)	iHeightL	カメラの画像メモリY方向サイズ(画素)	値	意味	0	通常のカメラ	1	シャインプルーフ構成のカメラ
iCam	カメラ種別。以下のいずれかを指定します。(※1)													
iWidthL	カメラの画像メモリX方向サイズ(画素)													
iHeightL	カメラの画像メモリY方向サイズ(画素)													
値	意味													
0	通常のカメラ													
1	シャインプルーフ構成のカメラ													
入力	iWldFlag	ワールド座標系の種別 以下のいずれかを指定します。なお、座標系の詳細は Appendix2 を ご参照下さい。												
出力	vpCalib	キャリブレーション・ディスクリプタ なお、*vpCalibは必ず初期設定してください。 (例) IntPtr vpCalib = IntPtr.Zero;												

戻り値

値	定数	意味
0	FUNC_OK	正常終了
-1	FUNC_NOT_ENOUGH_MEM	ワークメモリ不足
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適當
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項

本ライブラリを使用した際は、キャリブレーション作業終了時に[CalibClose](#)関数を必ず実行してください。さもないと、作業領域が解放されません。

SaveCalibDataF

機能 キャリブレーションデータのファイル保存

クラス EyemSharpLsm.eyemLsm

形式 using EyemSharp;
 using EyemSharpLsm;

```
int SaveCalibDataF(string cpFilePath, IntPtr vpCalib)
```

解説 キャリブレーションデータを指定されたファイルに保存します。

引数

入力	cpFilePath	保存先のファイル・パス
入力	vpCalib	キャリブレーション・ディスクリプタ CalibOpen 関数の出力値を指定します。

戻り値

値	定数	意味
0	FUNC_OK	正常終了
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適當
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項 特にありません。

LoadCalibDataF

機能 計測のオープン処理

クラス EyemSharpLsm.eyemLsm

形式 using EyemSharp;
using EyemSharpLsm;

```
int LoadCalibDataF(string cpFilePath, ref IntPtr vpCalib)
```

解説 保存してあるファイルからキャリブレーションデータを読み込み、キャリブレーション・ディスクリプタを生成します。また、作業領域の確保を行います。

引数

入力	cpFilePath	保存してあるキャリブレーションデータのファイル・パス
出力	vpCalib	キャリブレーション・ディスクリプタ なお、*vpCalibは必ず初期設定してください。 (例) IntPtr vpCalib = IntPtr.Zero;

戻り値

値	定数	意味
0	FUNC_OK	正常終了
-1	FUNC_NOT_ENOUGH_MEM	ワークメモリ不足
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適當
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項 本ライブラリを使用した際は、作業終了時に[CalibClose](#)関数を必ず実行してください。さもないと、作業領域が解放されません。

CalibClose

機能 クローズ処理

クラス EyemSharpLsm.eyemLsm

形式
using EyemSharp;
using EyemSharpLsm;

```
void CalibClose(ref IntPtr vpCalib)
```

解説 キャリブレーションまたは計測で使用したワークメモリの解放を行います。

引数

入出力	vpCalib	キャリブレーション・ディスクリプタ CalibOpen 関数または LoadCalibDataF 関数の出力値を指定します。 vpCalibは以下の様に設定して戻します。 (例) vpCalib = IntPtr.Zero;
-----	---------	--

戻り値 ありません。

留意事項 特にありません。

ReadPtnLsrImg

機能 パターン・ドットとパターン上レーザー線の座標点の読み込み

クラス EyemSharpLsm.eyemLsm

形式 using EyemSharp;
using EyemSharpLsm;

```
int ReadPtnLsrImg(IntPtr vpCalib, int iImgNo, EyemLsmImg tpImgSet, int nAuto, ref EyemRect tpRoi)
```

解説 パターン・ドットおよびパターン上レーザー線の画像から、有効な座標点を読み込みます。

引数

入出力	vpCalib	キャリブレーション・ディスクリプタ CalibOpen 関数の出力値を指定します。								
入力	iImgNo	対象となるパターンとレーザーの画像セット 0番からの通し番号を入力します。								
入力	tpImgSet	対象となるパターンとレーザーの画像セット番号 なお、これらの画像の作成(キャリブレーションの方法)については、 Appendx3 をご参照ください。 <table border="1"><tr><td>tImgPtn</td><td>パターン・ドット画像メモリ</td></tr><tr><td>tImgLsr</td><td>パターン上レーザー線画像メモリ</td></tr></table>	tImgPtn	パターン・ドット画像メモリ	tImgLsr	パターン上レーザー線画像メモリ				
tImgPtn	パターン・ドット画像メモリ									
tImgLsr	パターン上レーザー線画像メモリ									
入力	nAuto	-1のとき、tpRoiを自動設定								
入力	tpRoi	処理範囲の矩形領域(左上座標およびサイズ) <table border="1"><tr><td>iX</td><td>左上X座標(画素)</td></tr><tr><td>iY</td><td>左上Y座標(画素)</td></tr><tr><td>iwidth</td><td>領域のX方向サイズ(画素)</td></tr><tr><td>iHeight</td><td>領域のY方向サイズ(画素)</td></tr></table>	iX	左上X座標(画素)	iY	左上Y座標(画素)	iwidth	領域のX方向サイズ(画素)	iHeight	領域のY方向サイズ(画素)
iX	左上X座標(画素)									
iY	左上Y座標(画素)									
iwidth	領域のX方向サイズ(画素)									
iHeight	領域のY方向サイズ(画素)									

戻り値

値	定数	意味
0	FUNC_OK	正常終了
-1	FUNC_NOT_ENOUGH_MEM	ワークメモリ不足
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適當
-100	FUNC_CANNOT_CALC	計算不可
-150	FUNC_FAILED_HOMOGRAPHY	ホモグラフィ行列計算失敗

-200	FUNC_FEW_PTN_SMPL_NUM	パターンのドット点数が足りない
-201	FUNC_CANNOT_READ_MARK	パターン大ドット点を読めない
-202	FUNC_CANNOT_READ_PTN	パターン・ドット点の実座標を読めない
-210	FUNC_FEW_LSR_SMPL_NUM	レーザー線の参照点数が足りない
-211	FUNC_CANNOT_READ_LSR	レーザー線の実座標を読めない
-212	FUNC_FAILED_LSR_LINE	レーザー線の直線近似失敗
-220	FUNC_CANNOT_FIND_PLATE	プレートが見つからない
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項

本関数は、[CalibOpen](#) 関数でのオープン時に限り使用可能となります。

CalibCameraParam

機能 カメラキャリブレーションの実行

クラス EyemSharpLsm.eyemLsm

形式 using EyemSharp;
using EyemSharpLsm;

```
int CalibCameraParam(IntPtr vpCalib, ref double dpRms)
```

解説 パターン・ドット点およびパターン上レーザー線の座標データ群から、カメラのキャリブレーションを行います。

引数

入出力	vpCalib	キャリブレーション・ディスクリプタ CalibOpen 関数の出力値を指定します。
入力	dpRms	バンドル調整評価関数値(画素) 残差(再投影誤差)の重み付き平均二乗誤差の平方根です。

戻り値

値	定数	意味
0以上		(正常終了) バンドル調整反復回数
-1	FUNC_NOT_ENOUGH_MEM	ワークメモリ不足
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適當
-100	FUNC_CANNOT_CALC	計算不可
-150	FUNC_FAILED_HOMOGRAPHY	ホモグラフィ行列計算失敗
-151	FUNC_FAILED_CAM_PRM	カメラパラメータ計算失敗
-152	FUNC_FAILED_BUNDLE_ADJ	バンドル調整失敗
-211	FUNC_CANNOT_READ_LSR	レーザーの実座標が読めない
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項 本関数は、[CalibOpen](#)関数でのオープン時に限り使用可能となります。

Measure

機能 レーザー線画像座標から3次元座標への変換

クラス EyemSharpLsm.eyemLsm

形式 using EyemSharp;
using EyemSharpLsm;

```
int Measure(IntPtr vpCalib, Bitmap tImage, int nAuto, ref EyemRect tpRoi,  
ref EyemLsmLsr tpParam)
```

解説 レーザー線の画像座標を取得し、3次元座標へ変換します。

引数

入出力	vpCalib	キャリブレーション・ディスクリプタ LoadCalibDataF 関数の出力値を指定します。								
入力	tImage	計測対象の画像メモリ先頭アドレス								
入力	nAuto	-1のとき、tpRoiを自動設定								
入力	tpRoi	処理範囲の矩形領域(左上座標およびサイズ) <table border="1"><tr><td>iX</td><td>左上X座標(画素)</td></tr><tr><td>iY</td><td>左上Y座標(画素)</td></tr><tr><td>iWidth</td><td>領域のX方向サイズ(画素)</td></tr><tr><td>iHeight</td><td>領域のY方向サイズ(画素)</td></tr></table>	iX	左上X座標(画素)	iY	左上Y座標(画素)	iWidth	領域のX方向サイズ(画素)	iHeight	領域のY方向サイズ(画素)
iX	左上X座標(画素)									
iY	左上Y座標(画素)									
iWidth	領域のX方向サイズ(画素)									
iHeight	領域のY方向サイズ(画素)									
入力	tpParam	レーザー点群の選別パラメータ。 <table border="1"><tr><td>dLevelMul</td><td>2値化レベルに対する倍率(標準値:1.0) これは、濃度下限値の指定を意味します。</td></tr><tr><td>dSigmaCoef</td><td>線幅σ値の係数(標準値:3.0) これは、線幅のレンジの指定を意味します。</td></tr></table>	dLevelMul	2値化レベルに対する倍率(標準値:1.0) これは、濃度下限値の指定を意味します。	dSigmaCoef	線幅 σ 値の係数(標準値:3.0) これは、線幅のレンジの指定を意味します。				
dLevelMul	2値化レベルに対する倍率(標準値:1.0) これは、濃度下限値の指定を意味します。									
dSigmaCoef	線幅 σ 値の係数(標準値:3.0) これは、線幅のレンジの指定を意味します。									

戻り値

値	定数	意味
0	FUNC_OK	正常終了
-1	FUNC_NOT_ENOUGH_MEM	ワークメモリ不足
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適当
-100	FUNC_CANNOT_CALC	計算不可
-153	FUNC_FAILED_UNDISTORT	歪み補正失敗

-999	FUNC_CANNOT_USE	ライブラリ使用不可
------	-----------------	-----------

留意事項

本関数は、[LoadCalibDataF](#) 関数でのオープン時に限り使用可能となります。

GetMeasureResult

機能 レーザー線の3次元座標の取得

クラス EyemSharpLsm.eyemLsm

形式
using EyemSharp;
using EyemSharpLsm;

```
int GetMeasureResult(IntPtr vpCalib, int iDataNo, ref EyemOcsDXY tpImgPt,  
ref EyemOcsDXYZ tpPt3D)
```

解説 [Measure](#) 関数による、レーザー線の画像座標値およびその3次元座標値への変換結果を取得します。

引数

入力	vpCalib	キャリブレーション・ディスクリプタ LoadCalibDataF 関数の出力値を指定します。						
入力	iDataNo	座標データの番号。0番からの通し番号を指定します。						
出力	tpImgPt	画像座標 <table border="1"><tr><td>dX</td><td>X座標(画素)</td></tr><tr><td>dY</td><td>Y座標(画素)</td></tr></table>	dX	X座標(画素)	dY	Y座標(画素)		
dX	X座標(画素)							
dY	Y座標(画素)							
出力	tpPt3D	3次元座標 <table border="1"><tr><td>dX</td><td>X座標(mm)</td></tr><tr><td>dY</td><td>Y座標(mm)</td></tr><tr><td>dZ</td><td>Z座標(mm)</td></tr></table>	dX	X座標(mm)	dY	Y座標(mm)	dZ	Z座標(mm)
dX	X座標(mm)							
dY	Y座標(mm)							
dZ	Z座標(mm)							

戻り値

値	定数	意味
0	FUNC_OK	正常終了
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適當
-100	FUNC_CANNOT_CALC	計算不可
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項 本関数は、[LoadCalibDataF](#) 関数でのオープン時に限り使用可能となります。

ChangeLsr2Dto3D

機能 画像座標点から3次元座標点への変換

クラス EyemSharpLsm.eyemLsm

形式 using EyemSharp;
using EyemSharpLsm;

```
int ChangeLsr2Dto3D(IntPtr vpCalib, ref EyemOcsDXY tpImgPt, ref EyemOcsDXYZ tpPt3D)
```

解説 画像座標点を、ワールド座標系内のレーザー平面上における3次元座標点へ変換します。
なお、ワールド座標系がレーザー平面座標系の場合は、3次元座標のz成分は常にゼロとなります。

引数

入力	vpCalib	キャリブレーション・ディスクリプタ CalibOpen 関数または LoadCalibDataF 関数の出力値を指定します。						
入力	tpImgPt	画像座標 <table border="1"><tr><td>dX</td><td>X座標(画素)</td></tr><tr><td>dY</td><td>Y座標(画素)</td></tr></table>	dX	X座標(画素)	dY	Y座標(画素)		
dX	X座標(画素)							
dY	Y座標(画素)							
出力	tpPt3D	3次元座標 <table border="1"><tr><td>dX</td><td>X座標(mm)</td></tr><tr><td>dY</td><td>Y座標(mm)</td></tr><tr><td>dZ</td><td>Z座標(mm)</td></tr></table>	dX	X座標(mm)	dY	Y座標(mm)	dZ	Z座標(mm)
dX	X座標(mm)							
dY	Y座標(mm)							
dZ	Z座標(mm)							

戻り値

値	定数	意味
0	FUNC_OK	正常終了
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適當
-153	FUNC_FAILED_UNDISTORT	歪み補正失敗
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項 特にありません。

Change3Dto2D

機能 3次元座標点から画像座標点への変換

クラス EyemSharpLsm.eyemLsm

形式 using EyemSharp;
using EyemSharpLsm;

```
int Change3Dto2D( IntPtr vpCalib, ref EyemOcsDXYZ tpPt3D, ref EyemOcsDXYZ tpImgPt )
```

解説 ワールド座標系における3次元座標点から、画像座標点への変換を行います。

引数

入力	vpCalib	キャリブレーション・ディスクリプタ CalibOpen 関数または LoadCalibDataF 関数の出力値を指定します。						
入力	tpPt3D	3次元座標 <table border="1"><tr><td>dX</td><td>X座標 (mm)</td></tr><tr><td>dY</td><td>Y座標 (mm)</td></tr><tr><td>dZ</td><td>Z座標 (mm)</td></tr></table>	dX	X座標 (mm)	dY	Y座標 (mm)	dZ	Z座標 (mm)
dX	X座標 (mm)							
dY	Y座標 (mm)							
dZ	Z座標 (mm)							
出力	tpImgPt	画像座標 <table border="1"><tr><td>dX</td><td>X座標 (画素)</td></tr><tr><td>dY</td><td>Y座標 (画素)</td></tr></table>	dX	X座標 (画素)	dY	Y座標 (画素)		
dX	X座標 (画素)							
dY	Y座標 (画素)							

戻り値

値	定数	意味
0	FUNC_OK	正常終了
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適當
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項 特にありません。

GetHomography

機能 ホモグラフィ行列の取得

クラス EyemSharpLsm.eyemLsm

形式 using EyemSharp;
using EyemSharpLsm;

```
int GetHomography( IntPtr vpCalib, ref EyemCalibHom tpHom )
```

解説 ホモグラフィ行列を取得します。この行列は、レーザー平面と画像平面との間の射影変換行列 **H** です。すなわち、レーザー平面3次元座標を (X, Y, 0)、およびレンズ歪みなしの画像座標を (x, y) とするとき、 λ を実数として、

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathbf{H} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

となります。

引数

入力	vpCalib	キャリブレーション・ディスクリプタ CalibOpen 関数または LoadCalibDataF 関数の出力値を指定します。				
出力	tpHom	ホモグラフィ行列 <table border="1"><tr><td>daH[3][3]</td><td>ホモグラフィ行列H</td></tr><tr><td>daInvH[3][3]</td><td>行列Hの逆行列H⁻¹</td></tr></table>	daH[3][3]	ホモグラフィ行列 H	daInvH[3][3]	行列 H の逆行列 H ⁻¹
daH[3][3]	ホモグラフィ行列 H					
daInvH[3][3]	行列 H の逆行列 H ⁻¹					

戻り値

値	定数	意味
0	FUNC_OK	正常終了
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適當
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項 特にありません。

GetCamIntParam

機能 カメラ内部パラメータの取得

クラス EyemSharpLsm.eyemLsm

形式 using EyemSharp;
using EyemSharpLsm;

```
int GetCamIntParam( IntPtr vpCalib, ref EyemCalibInt tpInt )
```

解説 カメラの内部パラメータを取得します。内部パラメータは、 x 方向、 y 方向それぞれの「焦点距離×スケール」(f_x および f_y)、「焦点距離×せん断係数」(f_s)、「画像中心」(u_0, v_0)および「半径方向の歪み(ラジアル歪み)係数」が得られます。これらを用いて、内部パラメータ行列 A は、次の 3×3 行列で表されます。

$$A = \begin{pmatrix} f_x & f_s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}.$$

引数

入力	vpCalib	キャリブレーション・ディスクリプタ CalibOpen 関数または LoadCalibDataF 関数の出力値を指定します。												
出力	tpInt	カメラの内部パラメータ <table border="1"><tr><td>daA[3][3]</td><td>内部パラメータ行列A</td></tr><tr><td>dFx、dFy</td><td>焦点距離×スケール(f_xおよびf_y) (画素)</td></tr><tr><td>dFs</td><td>焦点距離×せん断係数(f_s) (画素)</td></tr><tr><td>dUo、dVo</td><td>画像中心座標((u_0, v_0)) (画素)</td></tr><tr><td>dK1、dK2</td><td>ラジアル歪み係数</td></tr><tr><td>dTheta</td><td>画像面傾斜角(CCD面x軸周りの角度) (rad) (シャインプルーフ構成カメラの場合に使用可能)</td></tr></table>	daA[3][3]	内部パラメータ行列A	dFx、dFy	焦点距離×スケール(f_x および f_y) (画素)	dFs	焦点距離×せん断係数(f_s) (画素)	dUo、dVo	画像中心座標((u_0, v_0)) (画素)	dK1、dK2	ラジアル歪み係数	dTheta	画像面傾斜角(CCD面x軸周りの角度) (rad) (シャインプルーフ構成カメラの場合に使用可能)
daA[3][3]	内部パラメータ行列A													
dFx、dFy	焦点距離×スケール(f_x および f_y) (画素)													
dFs	焦点距離×せん断係数(f_s) (画素)													
dUo、dVo	画像中心座標((u_0, v_0)) (画素)													
dK1、dK2	ラジアル歪み係数													
dTheta	画像面傾斜角(CCD面x軸周りの角度) (rad) (シャインプルーフ構成カメラの場合に使用可能)													

戻り値

値	定数	意味
0	FUNC_OK	正常終了
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適當
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項 特にありません。

GetCamExtParam

機能 カメラ外部パラメータの取得

クラス EyemSharpLsm.eyemLsm

形式 using EyemSharp;
using EyemSharpLsm;

```
int GetCamExtParam( IntPtr vpCalib, int iPtnNo, ref EyemCalibExt tpExt )
```

解説 カメラの外部パラメータを取得します。外部パラメータは、各パターンをxy平面とする右手座標系の3次元座標をカメラ座標へ変換する「回転」および「平行移動」です。

引数

入力	vpCalib	キャリブレーション・ディスクリプタ CalibOpen 関数または LoadCalibDataF 関数の出力値を指定します。						
入力	iPtnNo	パターンの番号 0番からの通し番号を入力します。画像セット番号に対応しています。						
出力	tpExt	カメラ外部パラメータ <table border="1"><tr><td>daR[3][3]</td><td>回転行列</td></tr><tr><td>daT[3]</td><td>平行移動ベクトル</td></tr><tr><td>daRdr[3]</td><td>回転ベクトル(ロドリゲスの表現)</td></tr></table>	daR[3][3]	回転行列	daT[3]	平行移動ベクトル	daRdr[3]	回転ベクトル(ロドリゲスの表現)
daR[3][3]	回転行列							
daT[3]	平行移動ベクトル							
daRdr[3]	回転ベクトル(ロドリゲスの表現)							

戻り値

値	定数	意味
0	FUNC_OK	正常終了
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適當
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項 特にありません。

GetPtnLsrNum

機能 パターン・ドット点およびパターン上レーザー線の画像座標データ個数の取得

クラス EyemSharpLsm.eyemLsm

形式 using EyemSharp;
using EyemSharpLsm;

```
int GetPtnLsrNum( IntPtr vpCalib, int iDataNo, ref int ipPtnNum, ref int ipLsrNum )
```

解説 キャリブレーションで使用したパターン・ドット点およびパターン上レーザー線の画像座標データの個数を取得します。

引数

入力	vpCalib	キャリブレーション・ディスクリプタ CalibOpen 関数の出力値を指定します。
入力	iDataNo	座標データの番号 0番からの通し番号を入力します。画像セット番号に対応しています。
出力	ipPtnNum	パターン・ドット点の画像座標データの個数
出力	ipLsrNum	パターン上レーザー線の画像座標データの個数

戻り値

値	定数	意味
0	FUNC_OK	正常終了
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適當
-100	FUNC_CANNOT_CALC	計算不可(データなし)
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項 本関数は、[CalibOpen](#)関数でのオープン時に限り使用可能となります。

GetPattern

機能 パターン・ドット点の画像座標データの取得

クラス EyemSharpLsm.eyemLsm

形式 using EyemSharp;
using EyemSharpLsm;

```
int GetPattern( IntPtr vpCalib, int iDataNo, ref EyemOcsDXY taPtn )
```

解説 キャリブレーションで使ったパターン・ドット点の画像座標データを取得します。予め、[GetPtrLsrNum](#)関数で取得した個数分の配列 taPtn[]を確保してください。

引数

入力	vpCalib	キャリブレーション・ディスクリプタ CalibOpen 関数の出力値を指定します。				
入力	iDataNo	座標データの番号 0番からの通し番号を入力します。画像セット番号に対応しています。				
出力	taPtn[]	パターン・ドット点の画像座標データ。 <table border="1"><tr><td>dX</td><td>X座標(画素)</td></tr><tr><td>dY</td><td>Y座標(画素)</td></tr></table>	dX	X座標(画素)	dY	Y座標(画素)
dX	X座標(画素)					
dY	Y座標(画素)					

戻り値

値	定数	意味
0	FUNC_OK	正常終了
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適當
-100	FUNC_CANNOT_CALC	計算不可(データなし)
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項 本関数は、[CalibOpen](#)関数でのオープン時に限り使用可能となります。

GetLaser

機能 パターン上レーザー線の画像座標データの取得

クラス EyemSharpLsm.eyemLsm

形式 using EyemSharp;
using EyemSharpLsm;

```
int GetLaser( IntPtr vpCalib, int iDataNo, ref EyemOcsDXY taLsr )
```

解説 キャリブレーションで使ったパターン上のレーザー線の画像座標データを取得します。予め、[GetPtrLsrNum](#)関数で取得した個数分の配列 taLsr[] を確保してください。

引数

入力	vpCalib	キャリブレーション・ディスクリプタ CalibOpen 関数の出力値を指定します。				
入力	iDataNo	座標データの番号 0番からの通し番号を入力します。画像セット番号に対応しています。				
出力	taLsr[]	パターン上レーザー線の画像座標データ <table border="1"><tr><td>dX</td><td>X座標(画素)</td></tr><tr><td>dY</td><td>Y座標(画素)</td></tr></table>	dX	X座標(画素)	dY	Y座標(画素)
dX	X座標(画素)					
dY	Y座標(画素)					

戻り値

値	定数	意味
0	FUNC_OK	正常終了
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適當
-100	FUNC_CANNOT_CALC	計算不可(データなし)
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項 本関数は、[CalibOpen](#)関数でのオープン時に限り使用可能となります。

ShiftLsr3D

機能 レーザー線画像点列の平行移動

クラス EyemSharpLsm.eyemLsm

形式 using EyemSharp;
using EyemSharpLsm;

```
int ShiftLsr3D( IntPtr vpCalib, ref EyemOcsDXYZ tpShift, int iNum, ref  
EyemOcsDXY taSrcPt, ref EyemOcsDXY taDstPt )
```

解説 レーザー線の画像点列を、3次元空間内(ワールド座標系内)にて平行移動し、それを再び画像面へ投影したときのレーザー線画像点列を取得します。なお、ここで用いるワールド座標系は、キャリブレーション時に指定したものです。

引数

入力	vpCalib	キャリブレーション・ディスクリプタ CalibOpen 関数の出力値を指定します。						
入力	tpShift	3次元空間内の平行移動量(mm) <table border="1"><tr><td>dX</td><td>X方向移動量(mm)</td></tr><tr><td>dY</td><td>Y方向移動量(mm)</td></tr><tr><td>dZ</td><td>Z方向移動量(mm)</td></tr></table>	dX	X方向移動量(mm)	dY	Y方向移動量(mm)	dZ	Z方向移動量(mm)
dX	X方向移動量(mm)							
dY	Y方向移動量(mm)							
dZ	Z方向移動量(mm)							
入力	iNum	レーザー線画像点のデータ数。配列taSrcPt[]の要素数です。						
入力	taSrcPt[]	移動元のレーザー線画像点配列 <table border="1"><tr><td>dX</td><td>X座標(画素)</td></tr><tr><td>dY</td><td>Y座標(画素)</td></tr></table>	dX	X座標(画素)	dY	Y座標(画素)		
dX	X座標(画素)							
dY	Y座標(画素)							
出力	taDstPt[]	移動先のレーザー線画像点配列 taSrcPt[]と同じでも構いません。 <table border="1"><tr><td>dX</td><td>X座標(画素)</td></tr><tr><td>dY</td><td>Y座標(画素)</td></tr></table>	dX	X座標(画素)	dY	Y座標(画素)		
dX	X座標(画素)							
dY	Y座標(画素)							

戻り値

値	定数	意味
0	FUNC_OK	正常終了
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適当
-153	FUNC_FAILED_UNDISTORT	歪み補正失敗

-999	FUNC_CANNOT_USE	ライブラリ使用不可
------	-----------------	-----------

留意事項 特にありません。

CalcEquiPitchProfile

機能 レーザー線の等ピッチ 3次元座標の計算

クラス EyemSharpLsm.eyemLsm

形式 using EyemSharp;
using EyemSharpLsm;

```
int CalcEquiPitchProfile( IntPtr vpCalib, Bitmap tImage, int nAuto, ref  
EyemRect tpRoi, int nAuto, ref EyemLsmLsr tpParam, double dPitch )
```

解説 ≪ワールド座標系:第1パターン座標系 限定≫基準線上において等ピッチとなるように、レーザー線の3次元座標を計算します。ここで、基準線とは、レーザー平面と第1パターン座標系 xy 平面との交線です。また、等ピッチとは、隣り合うレーザー点から基準線へ下した垂線の足の間隔が、指定されたピッチであることを意味します。なお、レーザー線の情報がない部分は、線型補間により仮想点を計算します。

引数

入出力	vpCalib	キャリブレーション・ディスクリプタ CalibOpen 関数の出力値を指定します。								
入力	tImage	計測対象の画像メモリ								
入力	nAuto	-1のとき、tpRoiを自動設定								
入力	tpRoi	処理範囲の矩形領域(左上座標およびサイズ) <table border="1"><tr><td>iX</td><td>左上X座標(画素)</td></tr><tr><td>iY</td><td>左上Y座標(画素)</td></tr><tr><td>iWidth</td><td>領域のX方向サイズ(画素)</td></tr><tr><td>iHeight</td><td>領域のY方向サイズ(画素)</td></tr></table>	iX	左上X座標(画素)	iY	左上Y座標(画素)	iWidth	領域のX方向サイズ(画素)	iHeight	領域のY方向サイズ(画素)
iX	左上X座標(画素)									
iY	左上Y座標(画素)									
iWidth	領域のX方向サイズ(画素)									
iHeight	領域のY方向サイズ(画素)									
入力	tpParam	レーザー点群の選別パラメータ <table border="1"><tr><td>dLevelMul</td><td>2値化レベルに対する倍率(標準値:1.0) これは、濃度下限値の指定を意味します。</td></tr><tr><td>dSigmaCoef</td><td>線幅σ値の係数(標準値:3.0) これは、線幅のレンジの指定を意味します。</td></tr></table>	dLevelMul	2値化レベルに対する倍率(標準値:1.0) これは、濃度下限値の指定を意味します。	dSigmaCoef	線幅 σ 値の係数(標準値:3.0) これは、線幅のレンジの指定を意味します。				
dLevelMul	2値化レベルに対する倍率(標準値:1.0) これは、濃度下限値の指定を意味します。									
dSigmaCoef	線幅 σ 値の係数(標準値:3.0) これは、線幅のレンジの指定を意味します。									
入力	dPitch	ピッチ(mm) 正の数を指定します。								

戻り値

値	定数	意味
0以上		(正常終了)座標個数
-1	FUNC_NOT_ENOUGH_MEM	ワークメモリ不足
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適當
-100	FUNC_CANNOT_CALC	計算不可
-153	FUNC_FAILED_UNDISTORT	歪み補正失敗
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項

本関数は、[LoadCalibDataF](#) 関数でのオープン時、かつワールド座標系が第1パターン座標系の場合に限り使用可能となります。

GetEquiPitchProfile

機能 レーザー線の等ピッチ 3次元座標の取得

クラス EyemSharpLsm.eyemLsm

形式 using EyemSharp;
using EyemSharpLsm;

```
int GetEquiPitchProfile( IntPtr vpCalib, int iDataNo, ref EyemOcsDXYZ  
tpPt3D )
```

解説 <<ワールド座標系:第1パターン座標系 限定>>[CalcEquiPitchProfile](#)関数で求めたレーザー線の等ピッチ3次元座標値を取得します。

引数

入力	vpCalib	キャリブレーション・ディスクリプタ CalibOpen 関数の出力値を指定します。						
入力	iDataNo	座標データの番号 0番からの通し番号を指定します。						
出力	tpPt3D	3次元座標 <table border="1"><tr><td>dX</td><td>X 座標 (mm)</td></tr><tr><td>dY</td><td>Y 座標 (mm)</td></tr><tr><td>dZ</td><td>Z 座標 (mm)</td></tr></table>	dX	X 座標 (mm)	dY	Y 座標 (mm)	dZ	Z 座標 (mm)
dX	X 座標 (mm)							
dY	Y 座標 (mm)							
dZ	Z 座標 (mm)							

戻り値

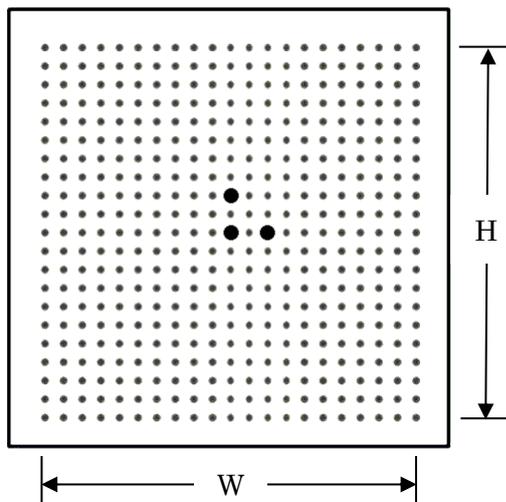
値	定数	意味
0	FUNC_OK	正常終了
-2	FUNC_ILLEGAL_ARGUMENT	引数が不適當
-100	FUNC_CANNOT_CALC	計算不可(データなし)
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項 本関数は、[LoadCalibDataF](#)関数でのオープン時、かつワールド座標系が第1パターン座標系の場合に限り使用可能となります。

Appendix1. キャリブレーション治具情報構造体 EyemLsmJig の設定

本ライブラリでは、下図に示すような大小の円形ドットパターンで構成されたキャリブレーション治具を想定しています。すなわち、格子点に小ドットを配置し、中央部分に座標軸を認識するための大ドットを配置します。大ドットについては、まず座標原点としてプレート中央に1つ配置し、さらにx軸上およびy軸上にそれぞれ1つずつ、合わせて3つを配置します。

このキャリブレーション治具の情報を設定する EyemLsmJig 構造体の具体的内容は以下のとおりです。



dPatternSizeX	パターン横方向(x方向)サイズ(mm)。上図の W サイズです。
dPatternSizeY	パターン縦方向(y方向)サイズ(mm)。上図の H サイズです。
dDotSizeSmall	小ドットサイズ(直径)(mm)
dDotSizeLarge	大ドットサイズ(直径)(mm)
dDotPitchSmallX	小ドット横方向(x方向)間隔(mm)
dDotPitchSmallY	小ドット縦方向(y方向)間隔(mm)
dDotPitchLargeX	大ドット横方向(x方向)間隔(mm)
dDotPitchLargeY	大ドット縦方向(y方向)間隔(mm)
iLargeDotNum	大ドット個数。現状は3を指定します。
iDotColor	ドットの色 2値化におけるドットの色です。EYEM_BIN_COLOR 列挙型の BLACK (黒)または WHITE(白)のいずれかを指定します。
iDotAreaThrs	ドットの面積下限値(画素) 2値化におけるノイズ除去のための値です。指定値以上の面積をもつ2値ブロップが処理対象となります。
dDotEdgeStrThrs	ドットのエッジ強度下限値 ピントぼけたドットを除去するための値です。0 以上の値を指定しま

	す。(標準値:0)
iMinPtnPtNum	ドット最小数 認識すべきドット個数の最小値を $\max(4, iLargeDotNum)$ 以上の値で指定します。

以下は、パターン上に照射したレーザーに関する設定です。

iLsrSmplStep	レーザー線のサンプリングステップ(画素) レーザー線をサンプリングする間隔を1以上の値で指定します。(標準値:10)
iMinLsrPtNum	レーザー線の参照点最小数 レーザー線のサンプリング個数の最小値を指定します。(標準値:30)

Appendix2. ワールド座標系

本ライブラリで使用できるワールド座標系は以下の3種類で、すべて右手座標系です。

(1) 第1パターン座標系

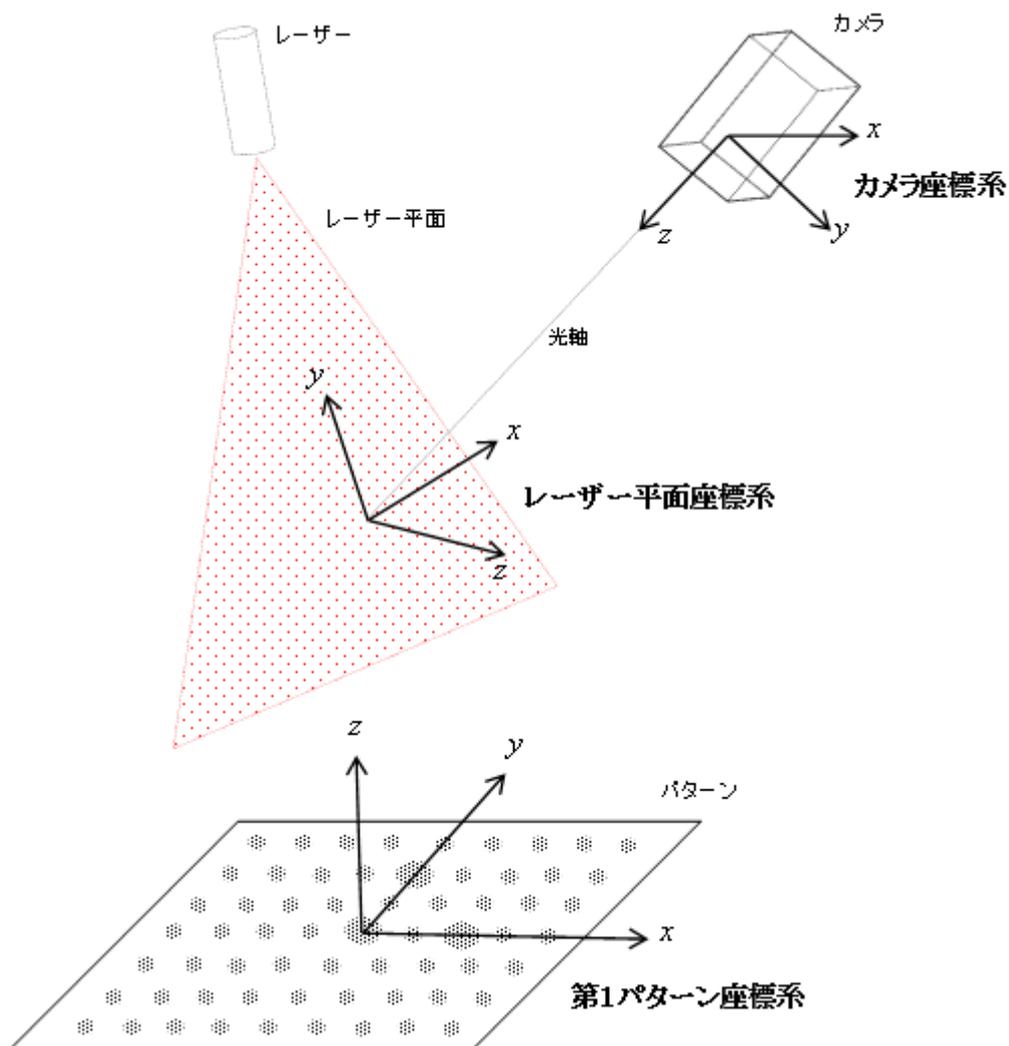
キャリブレーション・パターン画像群における1枚目の画像のパターン位置をワールド座標のxy面とする座標系です。

(2) カメラ座標系

レンズ中心を原点とし、光軸をz軸とした座標系です。

(3) レーザー平面座標系

レンズ光軸とレーザー平面の交点を原点とし、レーザー平面をxy面とした座標系です。この座標系では、レーザー線の3次元座標のz成分は必ずゼロとなります。



Appendix3. キャリブレーションの方法

キャリブレーションは、レーザーをパターン上に照射し、その画像を読み込むことで行います。具体的な手順は以下のとおりです。

- ① パターンをカメラに正対するように向け、照射されたレーザー線がパターンの中心付近を通るように設置します。(これが、第1パターン・ワールド座標系のxy面となります。)
- ② パターンを固定したまま、レーザーを照射した場合の画像と、照射しない場合の画像を1セットとして読み込みます。
- ③ パターンとカメラの距離およびパターンの向きを変えながら②の手順を繰り返し、画像セットを3セット以上取得します。その際、照射されたレーザー線がパターンの中心付近を通るようにします。
- ④ 本キャリブレーションライブラリを実行します。

留意事項

- (1) 画像内に映るパターンが画像内をまんべんなく、さらに、どの2枚も平行とならないようにパターンの距離と向きを調整してください。
- (2) レーザー線(レーザー平面)は、カメラ座標系x軸またはy軸に平行であることを想定しています。