

カメラキャリブレーション

(Ver.1.4)

2015 年 6 月

株式会社 アイディール

目次

eyemCalibCalcHomography.....	1
機 能 2平面間の射影変換 (ホモグラフィ)	1
eyemCalibCameraParam	3
機 能 カメラの内部・外部パラメータ	3
eyemCalibRodrigues	5
機 能 ロドリゲス変換	5
eyemCalibNormalizeData	6
機 能 座標データの正規化	6
eyemCalibUnDistortPoint	8
機 能 レンズ歪みの補正	8
eyemCalibDistortPoint	9
機 能 レンズ歪みの付加	9
eyemCalibCameraMatrix	10
機 能 カメラ行列	10
eyemCalibChange3Dto2D	11
機 能 3次元座標点の画像平面上への投影	11

eyemCalibCalcHomography

機 能 2平面間の射影変換(ホモグラフィー)

形 式 `#include "eyemLib.h"`
`int eyemCalibCalcHomography (int iNum, void *vpSrcPt, void *vpDstPt,`
`double dRobustCoef, void *vpHom);`

解 説 二つの平面間の射影変換行列(ホモグラフィー行列)を求めます。
すなわち, 平面 α の同次座標点 $\mathbf{x}_\alpha = (x_\alpha, y_\alpha, 1)$ を平面 β の同次座標点 $\mathbf{x}_\beta = (x_\beta, y_\beta, 1)$ へ射影変換するときの, 2次元射影変換行列(ホモグラフィー行列) \mathbf{H} を求めます。

$$\lambda \mathbf{x}_\beta = \mathbf{H} \mathbf{x}_\alpha \ (\lambda: \text{実数}).$$

なお, \mathbf{x}_α は誤差なし, \mathbf{x}_β は誤差ありを想定して, 再投影誤差の最小化(バンドル調整)を行っています。

引 数

iNum	座標データ数(4以上)です。
*vpSrcPt	平面 α の座標点です。 データは double型 の配列であり, その要素は $0 \leq k < \text{iNum}$ とするとき, $[\text{k}][0]$: x 座標, $[\text{k}][1]$: y 座標, または $[\text{k}*2]$: x 座標, $[\text{k}*2+1]$: y 座標のいずれかとします。
*vpDstPt	平面 β の座標点です。 データは double型 の配列であり, その要素は $0 \leq k < \text{iNum}$ とするとき, $[\text{k}][0]$: x 座標, $[\text{k}][1]$: y 座標, または $[\text{k}*2]$: x 座標, $[\text{k}*2+1]$: y 座標のいずれかとします。
dRobustCoef	ロバスト推定用の座標点群のばらつき(σ)係数です。 バンドル調整のロバスト推定に用います。負数を入力すると, ロバスト推定を行わず, 最小二乗法によるバンドル調整となります。平面 β の座標点群が, 真の位置から標準偏差 σ に対して $k\sigma$ ばらつきを持つと推定されるとき, その k を入力します。推奨値は 2.7 です。
*vpHom	ホモグラフィー行列が格納されます。 double[9] または double[3][3] の配列です。Hom[2][2] で割って正規化したものを出力とします。

戻り値 エラー報告です。

0以上	正常終了(バンドル調整反復回数)
FUNC_NOT_ENOUGH_MEM	メモリ不足
FUNC_ILLEGAL_ARGUMENT	パラメータが不正
FUNC_FAILED_EIGEN	固有値計算失敗

FUNC_FAILED_BUNDLE_ADJ	バンドル調整失敗
------------------------	----------

留意事項

特にありません。

eyemCalibCameraParam

機 能 カメラの内部・外部パラメータ

形 式 `#include "eyemLib.h"`
`int eyebCalibCameraParam (int iPtnNum, EyemCalibPtn taPtn[],`
`double dRobustCoef, EyemCalibInt *tpPrmA,`
`EyemCalibExt taRT[], double *dpRms, int iFlag);`

解 説 Zhang の方法に従い、カメラの内部および外部パラメータを、再投影誤差の最小化(バンドル調整)により求めます。

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)、 (u_0, v_0) および「半径方向の歪み(ラジアル歪み)係数」を求めます。これらを用いて、内部パラメータ行列 \mathbf{A} は、次の 3×3 行列で表されます。

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

また、外部パラメータは、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」を求めます。

なお、Zhangの方法は、次の論文に従っています。

Z. Zhang, "Flexible Camera Calibration by Viewing a Plane from Unknown Orientations.",
In Proc. 7th Int'l Conf. on Computer Vision, Vol. 1, pp666-673,1999.

引 数 `iPtnNum` パターン数(3以上)です。
`taPtn[]` パターンの画像座標および実座標です。
`dRobustCoef` ロバスト推定用の座標点群のばらつき(σ)係数です。
 バンドル調整のロバスト推定に用います。負数を入力すると、ロバスト推定を行わず、最小二乗法によるバンドル調整となります。画像座標点群が、真の位置から標準偏差 σ に対して $k\sigma$ ばらつきを持つと推定される
 とき、その k を入力します。推奨値は 2.7 です。
`*dpRms` バンドル調整評価関数値(再投影誤差, 単位:画素)が格納されます。
 具体的には、残差の重み付き平均二乗誤差の平方根(RMS)です。なお、
 ここでの残差とは、パターンの画像座標点と、対応する実座標をカメラパ
 ラメータにより画像面へ投影した点、との距離です。また、重みとは、ロ
 バスト推定で用いた重みのことです。
`*tpPrmA` カメラの内部パラメータが格納されます。

taRT[] カメラの外部パラメータが、パターンごとに格納されます.
iFlag カメラの種別です. 現状では0を入力して下さい.

戻り値

エラー報告です.

0以上	正常終了(バンドル調整反復回数)
FUNC_NOT_ENOUGH_MEM	メモリ不足
FUNC_ILLEGAL_ARGUMENT	パラメータが不正
FUNC_FAILED_HOMOGRAPHY	ホモグラフィー行列計算失敗
FUNC_FAILED_CAM_PRM	カメラパラメータ計算失敗
FUNC_FAILED_BUNDLE_ADJ	バンドル調整失敗

留意事項

特にありません.

eyemCalibRodrigues

機 能 ロドリゲス変換

形 式 `#include "eyemLib.h"`
`int eyedCalibRodrigues (int iMode, double daV[3], double daR[3][3]);`

解 説 ロドリゲス変換 (3次元回転ベクトル⇒3×3回転行列) またはその逆変換を行います.
 回転ベクトルは, 方向を回転軸方向にとり, 大きさを回転角としたベクトルです.

引 数 iMode 処理選択のフラグです.
 回転ベクトル⇒回転行列の場合は**0**, 回転行列⇒回転ベクトルの場合は
 1 を入力して下さい.
 daV[3] 回転ベクトルです.
 daR[3][3] 回転行列です.

戻り値 エラー報告です.

FUNC_OK	正常終了
FUNC_ILLEGAL_ARGUMENT	パラメータが不正

留意事項 特にありません.

eyemCalibNormalizeData

機 能 座標データの正規化

形 式

```
#include "eyemLib.h"

int      eyemCalibNormalizeData ( int iDim, int iNum, void *vpSrc, void *vpDst,
                                   void *vpT, void *vpInvT );
```

解 説 2次元および3次元の座標データを正規化します。
正規化とは、次の変換を行うことです。

- (1) データの重心を原点とする。
- (2) 原点から各点までの距離の平均値が $\sqrt{2}$ となるようにスケーリングする。

引 数

iDim 座標データの次元 (2または3) です。

iNum 座標データ数 (1以上) です。

*vpSrc 座標データです。
データは double型 の配列であり、その要素は $0 \leq k < iNum$ とするとき、
[k][0]: x 座標, [k][1]: y 座標, [k][2]: z 座標, または [iDim*k]: x 座標,
[iDim*k+1]: y 座標, [iDim*k+2]: z 座標のいずれかとします。

*vpDst 正規化した座標データが格納されます。
データは double型 の配列であり、その要素は $0 \leq k < iNum$ とするとき、
[k][0]: x 座標, [k][1]: y 座標, [k][2]: z 座標, または [iDim*k]: x 座標,
[iDim*k+1]: y 座標, [iDim*k+2]: z 座標のいずれかとします。
vpSrc と同じでも構いません。

*vpT 正規化の変換行列が格納されます。必要がないときは NULL を入力して下さい。
iDim == 2 のとき, double[9] またはdouble[3][3] の配列です。
iDim == 3 のとき, double[16] またはdouble[4][4] の配列です。
なお、この行列には座標データを同次座標で用います。

*vpInvT 正規化の変換行列の逆行列が格納されます。必要がないときは NULL を入力して下さい。
iDim == 2 のとき, double[9] またはdouble[3][3] の配列です。
iDim == 3 のとき, double[16] またはdouble[4][4] の配列です。
なお、この行列には座標データを同次座標で用います。

戻り値 エラー報告です。

FUNC_OK	正常終了
FUNC_ILLEGAL_ARGUMENT	パラメータが不正

留意事項

特にありません。

eyemCalibUnDistortPoint

機 能 レンズ歪みの補正

形 式 `#include "eyemLib.h"`
`int eyedCalibUnDistortPoint (EyemCalibInt *tpPrmA, double dUo, double dVo,`
`double *dpU, double *dpV);`

解 説 画像座標点に対して、レンズ歪みの補正(歪みの除去)を反復計算にて行います。なお、考慮する歪みは、半径方向の歪み(ラジアル歪み)のみとしています。

引 数 `*tpPrmA` 関数 `eyemCalibCameraParam` で得られたカメラ内部パラメータです。
 (`tpPrmA->dTheta` は ゼロ でなければいけません。)

`dUo` 画像座標の x 座標 です。

`dVo` 画像座標の y 座標 です。

`*dpU` 歪み補正した座標の x 座標が格納されます。
 `dUo`と同じでも構いません。

`*dpV` 歪み補正した座標の y 座標が格納されます。
 `dVo`と同じでも構いません。

戻り値 エラー報告です。

0以上	正常終了(反復回数)
<code>FUNC_ILLEGAL_ARGUMENT</code>	パラメータが不正
<code>FUNC_FAILED_UNDISTORT</code>	歪み補正失敗(反復回数オーバー)

留意事項 特にありません。

eyemCalibDistortPoint

機 能 レンズ歪みの付加

形 式 `#include "eyemLib.h"`
`int eyedCalibDistortPoint (EyemCalibInt *tpPrmA, double dU, double dV,`
`double *dpU, double *dpV);`

解 説 歪みのない画像座標点に対して, レンズ歪みを施した画像座標を求めます, 考慮する歪みは, 半径方向の歪み(ラジアル歪み)のみとしています.

引 数 `*tpPrmA` 関数 `eyemCalibCameraParam` で得られたカメラ内部パラメータです.
 (`tpPrmA->dTheta` は ゼロ でなければいけません.)

`dU` 歪みのない画像座標の x 座標です.
`dV` 歪みのない画像座標の y 座標です.

`*dpU` 歪みを施した画像座標の x 座標が格納されます.
 `dU`と同じでも構いません.

`*dpV` 歪みを施した画像座標の y 座標が格納されます.
 `dV`と同じでも構いません.

戻り値 エラー報告です.

<code>FUNC_OK</code>	正常終了
<code>FUNC_ILLEGAL_ARGUMENT</code>	パラメータが不正

留意事項 特にありません.

eyemCalibCameraMatrix

機 能 カメラ行列

形 式 `#include "eyemLib.h"`
`int eyedCalibCameraMatrix(EyemCalibInt *tpPrmA, EyemCalibExt *tpRT,`
`double daP[3][4], int iFlag);`

解 説 カメラ行列 **P** を求めます。
行列 **P** (3×4行列) は, カメラ内部パラメータ行列 **A** (3×3行列) およびカメラ外部パラメータ (**R** : 3×3回転行列, **t** : 3次元平行移動ベクトル) を用いて, **P** = **A**[**R** **t**] と表されます。

引 数 `*tpPrmA` 関数 `eyemCalibCameraParam` で得られたカメラ内部パラメータです。
`*tpRT` 関数 `eyemCalibCameraParam` で得られたカメラ外部パラメータです。
`daP[3][4]` カメラ行列 **P** が格納されます。3行4列の配列を指定してください。
 `daP[2][3]` で割って正規化したものを出力とします。
`iFlag` カメラの種別です。現状では0を入力して下さい。

戻り値 エラー報告です。

FUNC_OK	正常終了
FUNC_ILLEGAL_ARGUMENT	パラメータが不正

留意事項 特にありません。

eyemCalibChange3Dto2D

機 能 3次元座標点の画像平面上への投影

形 式

```
#include "eyemLib.h"
in      eyedCalibChange3Dto2D( EyemOcsDXYZ *tpPt3D, EyemCalibInt *tpPrmA,
                                EyemCalibExt *tpRT, EyemOcsDXY *tpPt2D, int iFlag );
```

解 説 指定された3次元座標点を画像平面上へ投影し、その2次元座標(画像座標)を求めます。

引 数

*tpPt3D	3次元座標点です。
*tpPrmA	関数 eyedCalibCameraParam で得られたカメラ内部パラメータです。
*tpRT	関数 eyedCalibCameraParam で得られたカメラ外部パラメータです。
*tpPt2D	2次元座標(画像座標)が格納されます。
iFlag	カメラの種別です。現状では0を入力して下さい。

戻り値 エラー報告です。

FUNC_OK	正常終了
FUNC_ILLEGAL_ARGUMENT	パラメータが不正

留意事項 特にありません。

改訂履歴

Version No.	内 容
1.0	<ul style="list-style-type: none"> 新規発行
1.1	<ul style="list-style-type: none"> 説明の追記および語句の修正
1.2	<ul style="list-style-type: none"> カメラ内部・外部パラメータ関数に, セン断係数の追記 カメラ行列関数, および3次元座標点の画像平面上への投影関数を追加
1.3	<ul style="list-style-type: none"> カメラ内部・外部パラメータ関数に, 内部パラメータの説明を追記
1.4	<ul style="list-style-type: none"> カメラ内部・外部パラメータ関数に, バンドル調整評価関数値の追記