

レンズ歪み補正ライブラリ

(Ver.1.0)

2018年7月

株式会社 アイディール

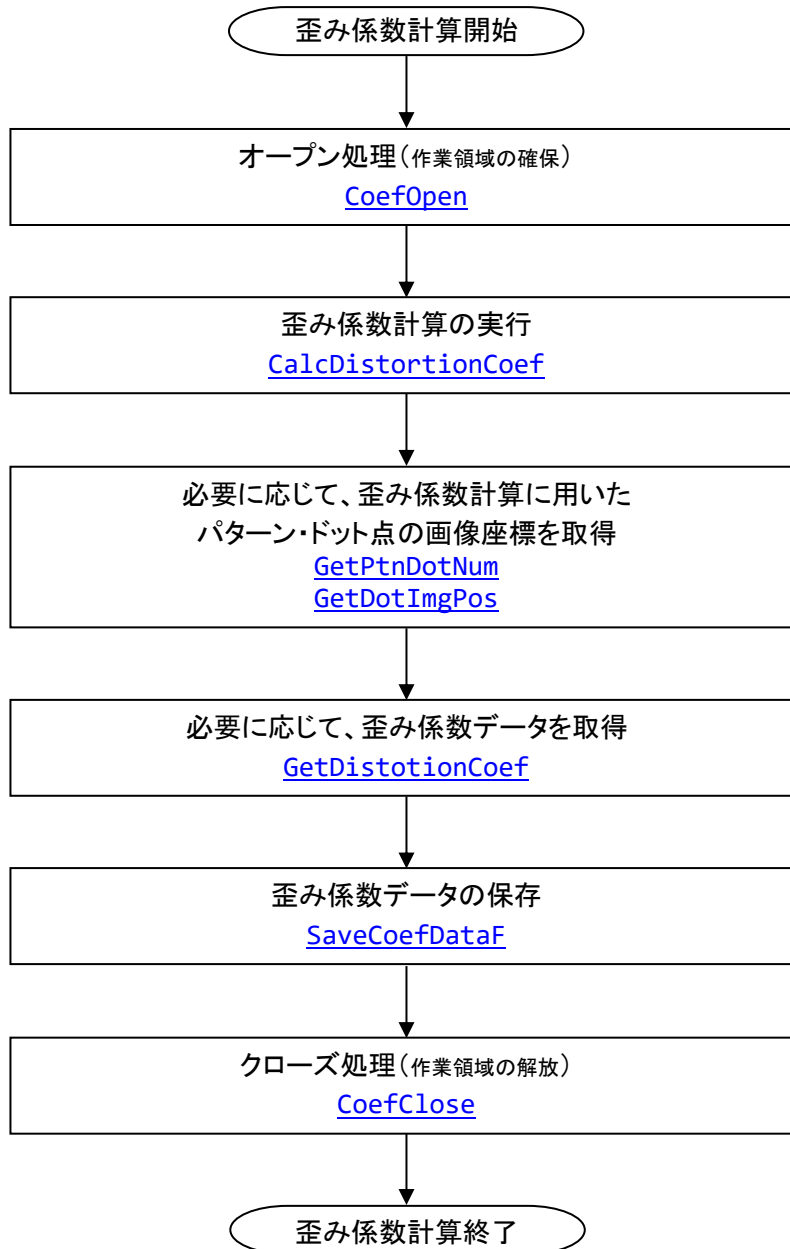
目次

目次.....	i
1. レンズ歪み補正ライブラリの使用手順.....	1
(1) 歪み係数計算フェーズ.....	1
(2) 歪み補正フェーズ.....	2
2. ライブラリの詳細説明.....	2
CoefOpen	3
機能 歪み係数計算のオープン処理.....	3
SaveCoefDataF	4
機能 歪み係数データのファイル保存.....	4
LoadCoefDataF	5
機能 歪み係数データのファイル読み込み.....	5
CoefClose	6
機能 クローズ処理.....	6
CalcDistortionCoef	7
機能 レンズ歪み係数の計算.....	7
GetPtnDotNum	8
機能 パターン・ドットの画像座標データ個数の取得.....	8
GetDotImgPos	9
機能 パターン・ドットの画像座標の取得.....	9
GetDistotionCoef	10
機能 歪み係数データの取得.....	10
UnDistImgSize	11
機能 歪み補正画像サイズの取得.....	11
UnDistortImage	13
機能 歪み補正画像の作成.....	13
UnDistortPoint	14
機能 原画像点から原画像サイズの歪み補正画像点への変換.....	14
UnDistortPoint2	15
機能 原画像点から指定サイズの歪み補正画像点への変換.....	15
DistortPoint	16
機能 原画像サイズの歪み補正画像点から原画像点への変換.....	16
DistortPoint2	17
機能 指定サイズの歪み補正画像点から原画像点への変換.....	17
Appendix1. ドット・パターン治具情報構造体 EyemLensJig の設定	18
Appendix2. ドット・パターン治具ドット・パターン画像取得の留意事項.....	19
改訂履歴.....	20

1. レンズ歪み補正ライブラリの使用手順

レンズ歪み補正ライブラリでは、歪み係数計算フェーズと歪み補正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) 歪み係数計算フェーズ



(2) 歪み補正フェーズ



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。なお、ライブラリ内に出てくる定数および構造体の具体的内容は「ライブラリ共通事項.pdf」を参照してください。

CoefOpen

機能 歪み係数計算のオープン処理

クラス EyemSharpLens.eyemLens

形式 using EyemSharp;
using EyemSharpLens;

```
int CoefOpen( EyemLensJig *tpJig, EyemLensCam *tpCam, void **vpCoef );
```

解説 作業領域の確保を行い、歪み係数ディスクリプタを生成します。

引数

入力	*tpJig	ドット・パターン治具情報 詳細は Appendix1 を参照してください。				
入力	*tpCam	カメラ情報(画像サイズ) <table border="1"><tr><td>iWidth</td><td>カメラの画像メモリX方向サイズ(画素)</td></tr><tr><td>iHeight</td><td>カメラの画像メモリY方向サイズ(画素)</td></tr></table>	iWidth	カメラの画像メモリX方向サイズ(画素)	iHeight	カメラの画像メモリY方向サイズ(画素)
iWidth	カメラの画像メモリX方向サイズ(画素)					
iHeight	カメラの画像メモリY方向サイズ(画素)					
出力	**vpCoef	歪み係数ディスクリプタ(データ領域の先頭アドレス) なお、*vpCoefは必ずNULLに初期設定してください。				

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、歪み係数計算終了時に[CoefClose](#)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。

SaveCoefDataF

機能 歪み係数データのファイル保存

クラス EyemSharpLens.eyemLens

形式 using EyemSharp;
using EyemSharpLens;

```
int SaveCoefDataF( const char *cpFilePath, void *vpCoef );
```

解説 歪み係数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	*vpCoef	歪み係数ディスクリプタ。 CoefOpen 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
-100	FUNC_CANNOT_CALC	計算不可(ファイル保存失敗)
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項 特になし。

LoadCoefDataF

機能 歪み係数データのファイル読み込み

クラス EyemSharpLens.eyemLens

形式 using EyemSharp;
using EyemSharpLens;

```
int LoadCoefDataF( const char *cpFilePath, void **vpCoef );
```

解説 保存してあるファイルから歪み係数データを読み込み、歪み係数ディスクリプタの生成および作業領域の確保を行います。

引数

入力	*cpFilePath	保存してある歪み係数データのファイルパス
出力	**vpCoef	歪み係数ディスクリプタ(データ領域の先頭アドレス) なお、*vpCoefは必ずNULLに初期設定してください。

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(ファイル読み込み失敗)
-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、歪み補正終了時に [CoefClose](#) 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。

CoefClose

機 能 クローズ処理

クラス EyemSharpLens.eyemLens

形 式

```
using EyemSharp;  
using EyemSharpLens;
```

```
void CoefClose( void **vpCoef );
```

解 説 歪み係数計算または歪み補正で使用したワークメモリの解放を行います。

引 数

入出力	**vpCoef	歪み係数ディスクリプタ CoefOpen 関数または LoadCoefDataF 関数の出力値を指定します。 *vpCoefはNULLに設定して戻します。
-----	----------	---

戻り値 ありません。

留意事項 特にありません。

CalcDistortionCoef

機能 レンズ歪み係数の計算

クラス EyemSharpLens.eyemLens

形式
using EyemSharp;
using EyemSharpLens;

```
int CalcDistortionCoef( void *vpCoef, unsigned char *ucpPtnImg );
```

解説 ドット・パターン画像から、レンズ歪み係数を求めます。なお、歪み係数は、ラジアル歪み(半径方向歪み)のみを求めます。また、パターンの設置方法については、[Appendix2](#)を参照してください。

引数

入出力	*vpCoef	歪み係数ディスクリプタ CoefOpen 関数の出力値を指定します。
入力	*ucpPtnImg g	ドット・パターン画像の先頭アドレス

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
-200	FUNC_FEW_PTN_SMPL_NUM	パターンのドット点数が足りない
-202	FUNC_CANNOT_READ_PTN	パターン・ドット点の実座標が読めない
-220	FUNC_CANNOT_FIND_PLAT	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、[CoefOpen](#)関数でのオープン時に限り使用可能となります。

GetPtnDotNum

機能 パターン・ドットの画像座標データ個数の取得

クラス EyemSharpLens.eyemLens

形式 using EyemSharp;
 using EyemSharpLens;

```
int GetPtnDotNum( void *vpCoef, int *ipDotNum );
```

解説 有効なパターン・ドットの画像座標データの個数を取得します。

引数

入力	*vpCoef	歪み係数ディスクリプタ CoefOpen 関数の出力値を指定します。
出力	*ipDotNum	パターン・ドット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(データがない)
-999	FUNC_CANNOT_USE	ライブラリ使用不可

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

GetDotImgPos

機能 パターン・ドットの画像座標の取得

クラス EyemSharpLens.eyemLens

形式 using EyemSharp;
 using EyemSharpLens;

```
int GetDotImgPos( void *vpCoef, EyemOcsDXY taDotImgPos[] );
```

解説 歪み係数計算に使用したパターン・ドットの画像座標を取得します。予め、[GetPtnDotNum](#)関数で取得した個数分の配列 taDotImgPos[] を確保してください。

引数

入力	*vpCoef	歪み係数ディスクリプタ CoefOpen 関数の出力値を指定します。
出力	taDotImgPos[]	パターン・ドットの画像座標データ

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(データがない)
-999	FUNC_CANNOT_USE	ライブラリ使用不可

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

GetDistotionCoef

機能 歪み係数データの取得

クラス EyemSharpLens.eyemLens

形式 using EyemSharp;
using EyemSharpLens;

```
int GetDistotionCoef( void *vpCoef, EyemLensCoef *tpCoef );
```

解説 [CalcDistortionCoef](#)関数で得られた歪み係数データを取得します。取得できるデータは、ラジアル歪み係数 k_1 、 k_2 、歪み中心座標 (c_x, c_y) およびスケール s です。これらは、歪みのない理想画像座標 (x, y) および歪みのある観測画像座標 (\tilde{x}, \tilde{y}) に対する、次の関係式の係数です。

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + (k_1 r^2 + k_2 r^4) \begin{pmatrix} x - c_x \\ y - c_y \end{pmatrix},$$
$$r = s \sqrt{(x - c_x)^2 + (y - c_y)^2}.$$

引数

入力	*vpCoef	歪み係数ディスクリプタ CoefOpen 関数または LoadCoefDataF 関数の出力値を指定します。										
出力	*tpCoef	歪み係数データ <table border="1"><tr><td>dK1</td><td>ラジアル歪み係数 k_1.</td></tr><tr><td>dK2</td><td>ラジアル歪み係数 k_2.</td></tr><tr><td>dCx</td><td>歪み中心 x座標 c_x.</td></tr><tr><td>dCy</td><td>歪み中心 y座標 c_y.</td></tr><tr><td>dScale</td><td>スケール s.</td></tr></table>	dK1	ラジアル歪み係数 k_1 .	dK2	ラジアル歪み係数 k_2 .	dCx	歪み中心 x 座標 c_x .	dCy	歪み中心 y 座標 c_y .	dScale	スケール s .
dK1	ラジアル歪み係数 k_1 .											
dK2	ラジアル歪み係数 k_2 .											
dCx	歪み中心 x 座標 c_x .											
dCy	歪み中心 y 座標 c_y .											
dScale	スケール s .											

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
-999	FUNC_CANNOT_USE	ライブラリ使用不可

留意事項 特にありません。

UnDistImgSize

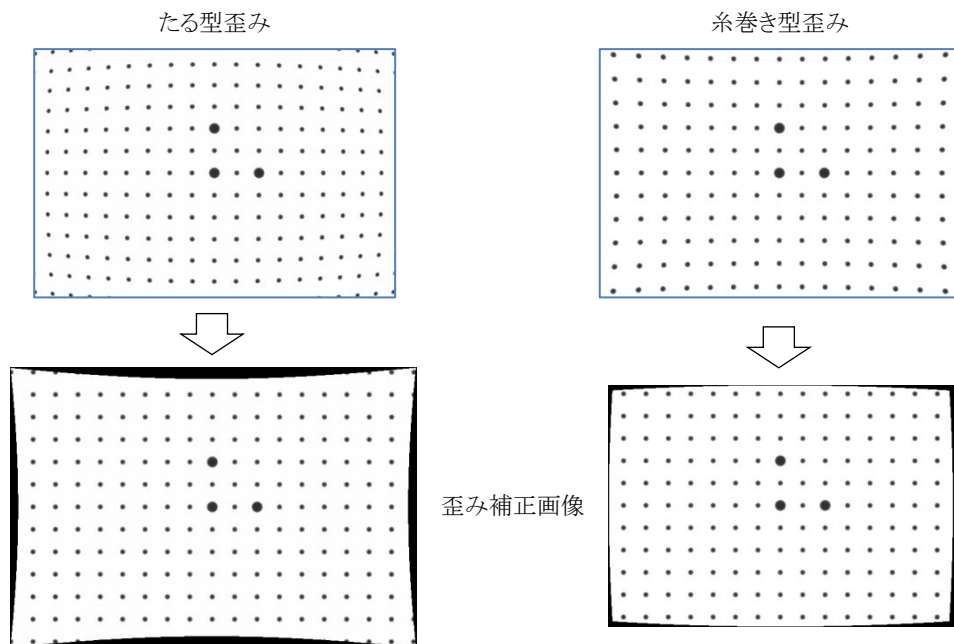
機能 歪み補正画像サイズの取得

クラス EyemSharpLens.eyemLens

形式
using EyemSharp;
using EyemSharpLens;

```
int UnDistImgSize( void *vpCoef, int *ipWidth, int *ipHeight );
```

解説 歪み補正画像の収まる画像サイズを取得します。歪み補正画像は、元画像のサイズをはみ出したり(たる型歪みの場合)、縮んだり(糸巻き型歪みの場合)します。本関数では、その補正画像のサイズを取得できます。



引数

入力	*vpCoef	歪み係数ディスクリプタ CoefOpen 関数または LoadCoefDataF 関数の出力値を指定します。
出力	*ipWidth	画像の幅(x方向サイズ)
出力	*ipHeight	画像の高さ(y方向サイズ)

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。

UnDistortImage

機能 歪み補正画像の作成

クラス EyemSharpLens.eyemLens

形式 using EyemSharp;
using EyemSharpLens;

```
int UnDistortImage( void *vpCoef, unsigned char *ucpImgSrc, int iDstW, int iDstH, unsigned char *ucpImgDst );
```

解説 歪み補正した画像を作成します。

引数

入力	*vpCoef	歪み係数ディスクリプタ CoefOpen 関数または LoadCoefDataF 関数の出力値を指定します。
入力	*ucpImgSrc	原画像メモリの先頭アドレス CoefOpen 関数で指定した画像サイズとします。
入力	iDstW	歪み補正画像を格納する画像メモリの幅(x方向サイズ)
入力	iDstH	歪み補正画像を格納する画像メモリの高さ(y方向サイズ)
出力	*ucpImgDst	歪み補正画像を格納する画像メモリの先頭アドレス

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。

UnDistortPoint

機能 原画像点から原画像サイズの歪み補正画像点への変換

クラス EyemSharpLens.eyemLens

形式 using EyemSharp;
using EyemSharpLens;

```
int UnDistortPoint( void *vpCoef, EyemOcsDXY *tpPtSrc,  
EyemOcsDXY *tpPtDst );
```

解説 原画像上の座標点を、歪み補正画像上の座標点に変換します。すなわち、座標点の歪みを除去します。ただし、歪み補正画像は、[UndistortImage](#) 関数において原画像と同サイズで作成された画像とします。

引数

入力	*vpCoef	歪み係数ディスクリプタ CoefOpen 関数または LoadCoefDataF 関数の出力値を指定します。				
入力	*tpPtSrc	原画像上の座標点 <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	*tpPtDst	歪み補正画像上の座標点 tpPtSrc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。

UnDistortPoint2

機能 原画像点から指定サイズの歪み補正画像点への変換

クラス EyemSharpLens.eyemLens

形式 using EyemSharp;
using EyemSharpLens;

```
int UnDistortPoint2( void *vpCoef, EyemOcsDXY *tpPtSrc, int iWidth,int  
iHeight, EyemOcsDXY *tpPtDst );
```

解説 原画像上の座標点を、歪み補正画像上の座標点に変換します。すなわち、座標点の歪みを除去します。ただし、歪み補正画像は、[UndistortImage](#)関数で作成された画像とします。

引数

入力	*vpCoef	歪み係数ディスクリプタ CoefOpen 関数または LoadCoefDataF 関数の出力値を指定します。				
入力	*tpPtSrc	原画像上の座標点 <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)					
入力	iWidth	歪み補正画像メモリの幅(x方向サイズ) UndistortImage 関数での指定値です。				
入力	iHeight	歪み補正画像メモリの高さ(y方向サイズ) UndistortImage 関数での指定値です。				
出力	*tpPtDst	歪み補正画像上の座標点 tpPtSrc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。

DistortPoint

機能 原画像サイズの歪み補正画像点から原画像点への変換

クラス EyemSharpLens.eyemLens

形式 using EyemSharp;
using EyemSharpLens;

```
int DistortPoint( void *vpCoef, EyemOcsDXY *tpPtSrc, EyemOcsDXY *tpPtDst );
```

解説 歪み補正画像上の座標点を、原画像上の座標点に変換します。すなわち、座標点に歪みを加えます。ただし、歪み補正画像は、[UndistortImage](#) 関数において原画像と同サイズで作成された画像とします。

引数

入力	*vpCoef	歪み係数ディスクリプタ CoefOpen 関数または LoadCoefDataF 関数の出力値を指定します。				
入力	*tpPtSrc	歪み補正画像上の座標点 <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	*tpPtDst	原画像上の座標点 tpPtSrc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。

DistortPoint2

機能 指定サイズの歪み補正画像点から原画像点への変換

クラス EyemSharpLens.eyemLens

形式 using EyemSharp;
using EyemSharpLens;

```
int DistortPoint2( void *vpCoef, EyemOcsDXY *tpPtSrc, int iWidth,  
int iHeight, EyemOcsDXY *tpPtDst );
```

解説 歪み補正画像上の座標点を、原画像上の座標点に変換します。すなわち、座標点に歪みを加えます。ただし、歪み補正画像は、[UndistortImage](#) 関数で作成された画像とします。

引数

入力	*vpCoef	歪み係数ディスクリプタ CoefOpen 関数または LoadCoefDataF 関数の出力値を指定します。				
入力	*tpPtSrc	歪み補正画像上の座標点 <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)					
入力	iWidth	歪み補正画像メモリの幅(x方向サイズ) UndistortImage 関数での指定値です。				
入力	iHeight	歪み補正画像メモリの高さ(y方向サイズ) UndistortImage 関数での指定値です。				
出力	*tpPtDst	原画像上の座標点 tpPtSrc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)					

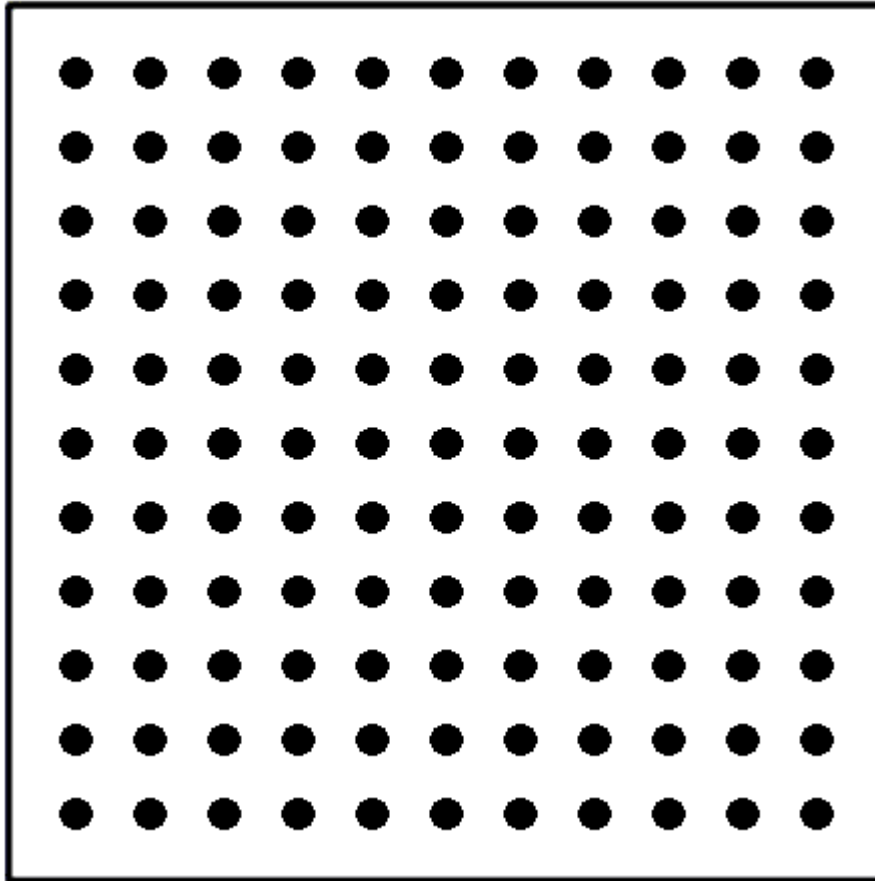
戻り値

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。

Appendix1. ドット・パターン治具情報構造体 EyemLensJig の設定

本ライブラリでは、下図に示すような円形ドット・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、正方等間隔(行および列のドット数が同じ、かつ等間隔)に円形ドットを配置した治具です。



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

`iLineDotNum`: 1ラインのドット個数

行および列における1ラインのドット個数です。

`iDotColor` : ドットの色

2値化におけるドットの色です。EYEM_BIN_BLACK(黒)または EYEM_BIN_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。

Appendix2. ドット・パターン治具ドット・パターン画像取得の留意事項

レンズ歪み係数の計算([CalcDistortionCoef](#)関数)に用いるドット・パターン画像は、以下の事項に留意して取得してください。

留意事項

- (1) 計算に用いる画像は、1枚だけです。
- (2) 画面内には、パターンがなるべく全面に映るようにしてください。パターンが映っている範囲が歪み補正の有効範囲と考えてください。
- (3) パターンは「回転」や「傾き」があっても構いませんが、なるべくレンズに正対に設置して、ドットがボケないようにしてください。

改訂履歴

Version No.	日付	内容
1.0	2018.07.16	・新規発行

以上