ISC ソフトウェアライブラリ

説明書 B

【ご注意】

1．本マニュアルの内容の一部または全部を無断転載することは禁止されています

2．本マニュアルの内容に関しては将来予告なしに変更することがあります

3．本マニュアルの内容について万全を期して作成しております、万一ご不審な点や誤り、記載漏れなどお気づきのことがございましたら、ご連絡ください

4．運用した結果の影響に関しては、3．項にかかわらず責任を負いかねますのでご了承ください

目次

1．概要

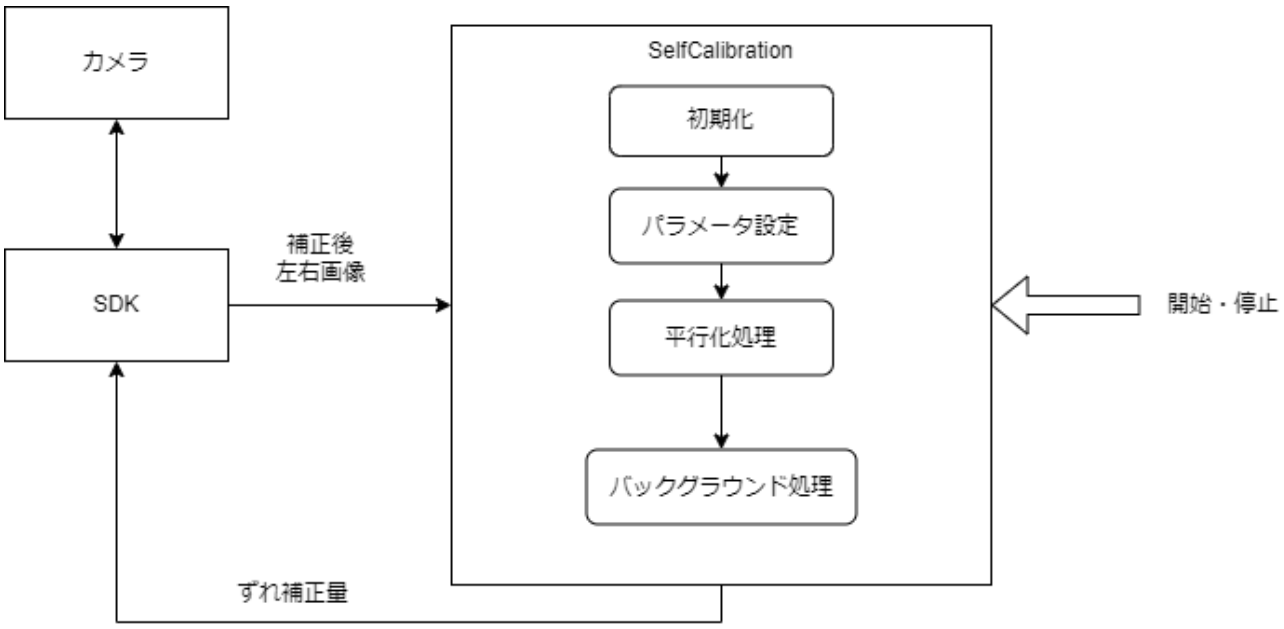
本ドキュメントは、ISC ソフトウェアライブラリの中のソフトウェアキャリブレーション・モジュールについて述べたものです。

ソフトウェアキャリブレーション・モジュールは、以下のモジュール名で提供されています。

処理ライブラリ名	モジュール名	DLL 名
SelftCalibration	IscSelfCalibration	IscSelfCalibration.dll

１．１　処理の流れ

下図に処理の流れを示します。



（1）SelftCalibration を初期化し、開始します
（2）SDK より左右の補正後画像を取得し、SelftCalibration へ与えます
（3）SelftCalibration は平行化を行い、カメラのパラメータを更新します

１．２　関数の使用手順

基本的な操作順を以下に示す。

操作	呼び出し関数	内容
初期化	initialize()	ソフトウェアキャリブレーションを初期化する
パラメータ設定	setMeshParameter() setMeshThreshold() setOperationMode() setAveragingParameter() setCriteria()	メッシュパラメータを設定する メッシュ閾値を設定する 動作モードを設定する ズレ量平均パラメータを設定する 平均ズレ量の判定基準を設定する
平行化処理	start() parallelize() stop()	ステレオ平行化処理を開始する 左右画像を平行にする ステレオ平行化処理を停止する
終了	finalize()	終了処理

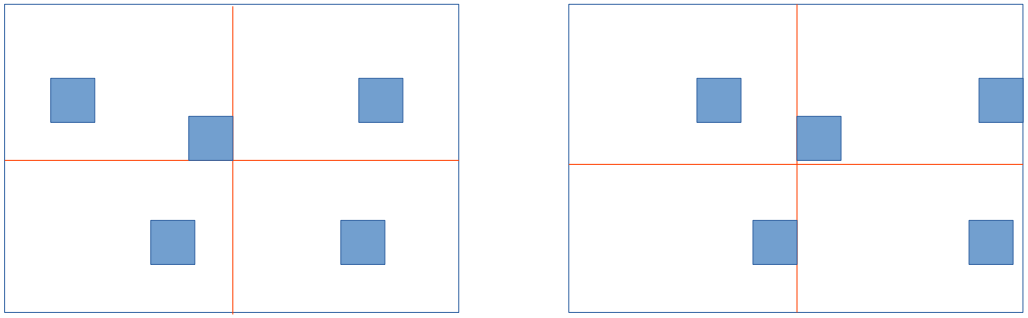
初期化 initialize() は、最初に一度だけ呼び出します。
（終了時は、必ず finalize()を呼び出します）
パラメータ設定用に、４つの関数があります。
パラメータ設定は最低限一度行う必要があります。また必要に応じて任意に設定可能です。
平行化処理では、初めに関数 start()を呼び出し、関数 parallelize() で左右の補正画像を受け付け可能とします。
その後、カメラより左右の補正画像を取得し、関数 parallelize()で平行化処理を繰り返し行います。
平行化処理では、複数フレームの画像からズレ量を計測し、ズレの補正量を求めます。必要に応じて結果をカメラへ書き込んで、平行へ戻します。
この平行化処理は、バックグラウンドで行われます。
バックグラウンドの処理中に、関数 parallelize()が呼び出された場合は、渡された画像に対しての処理は行われません。
また、処理を停止したい場合は、関数 stop()の呼び出し、関数 parallelize()が補正画像を受け付けないようにします。
再開には、再び関数 start()を呼び出します。

2．ステレオ平行化処理

2．1　ステレオ平行化について

ステレオカメラは、左右のカメラに写る同じ物体が、左右の画像で同じ高さ（同じ行座標）に見えるようにしています。
これをステレオ平行化と呼びます。
このため、左右の画像上の横方向のみの探索で視差を求めることができます。
これがステレオマッチングです。
次のイメージ図は、左右の画像が平行な状態にあることを示します。
右（基準）画像の 5 つのポイントと、それに対応する左（比較）画像のそれぞれのポイントが同じ高さ（同じ行座標）にあります。

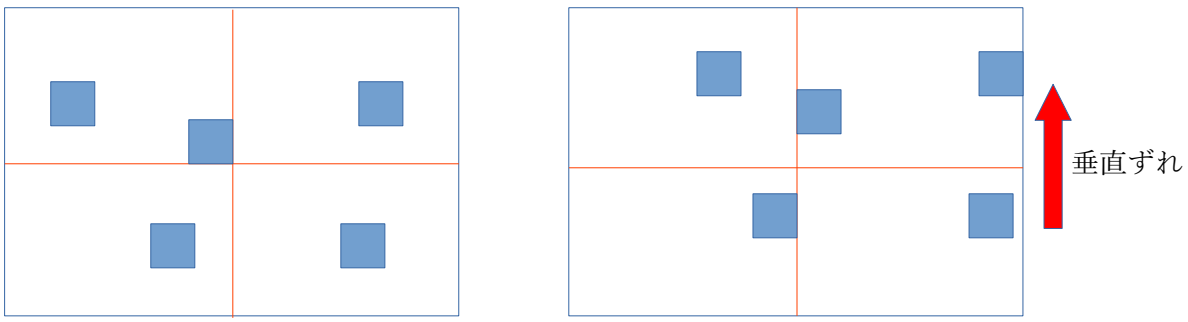
右（基準）画像　　左（比較）画像



次に平行でない状態のイメージ図を示します。
左画像が上へずれた場合（垂直ズレ）と、時計回りに回転した場合（回転ズレ）です。

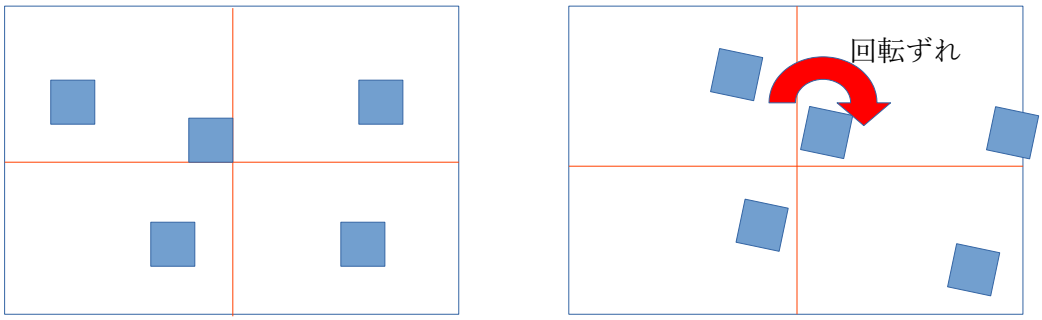
・ずれパターン 1

右（基準）画像　　左（比較）画像



・ずれパターン２

右（基準）画像　　左（比較）画像



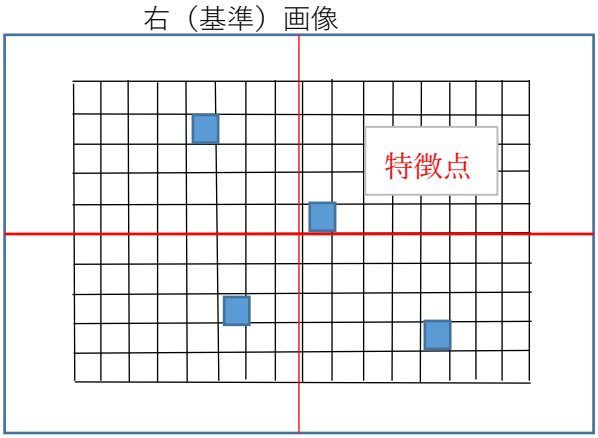
画像がずれると、横方向のみの探索では、対応点がみつかりません。
この状態では、ステレオマッチングできず視差を求めることができません。
ソフトウェアキャリブレーションでは、このズレを補正し平行に戻すことを行います。
補足）

・ズレの補正には、カメラの機能を使用します。
カメラには、補正画像を上下シフト、回転させる機能があり、それを使用します。
上下シフトの精度は 0.0625 画素幅です。
回転の精度は VM が 0.00017 ラジアン XC が 0.00024 ラジアンです。

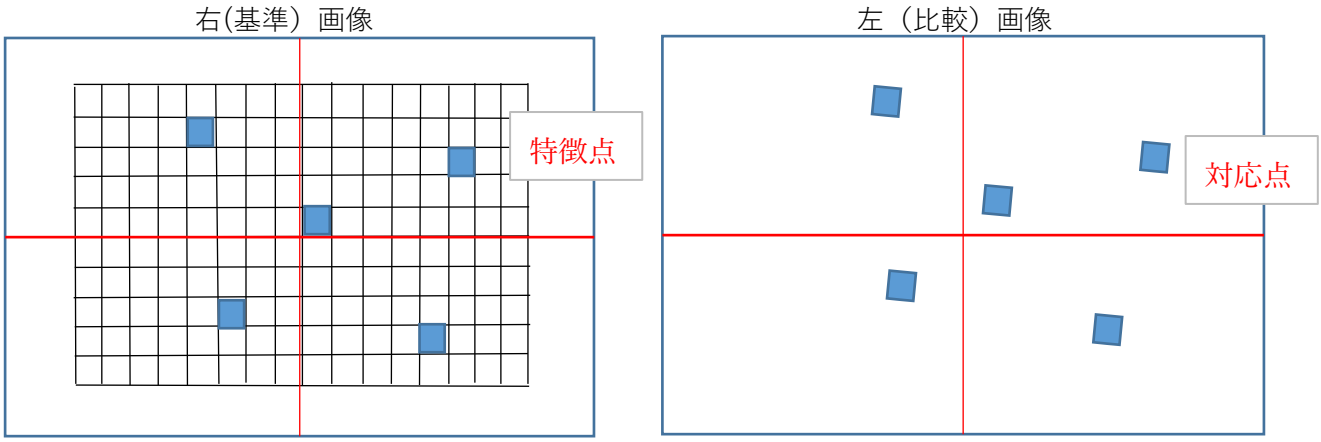
２．２　ステレオ平行化処理

ソフトウェアキャリブレーションのステレオ平行化処理を説明します。

（１）右（基準）画像から特徴点を決定
特徴点は、ズレ量を計測するための小領域です。
図中では、青色の四角に当たります。
画像を小さな領域に分割（メッシュ分割）し、その中からパターン強度の高いものを抽出します。
パターン強度が弱い場合、左（比較）画像より対応する領域を検出することが難しいためです。



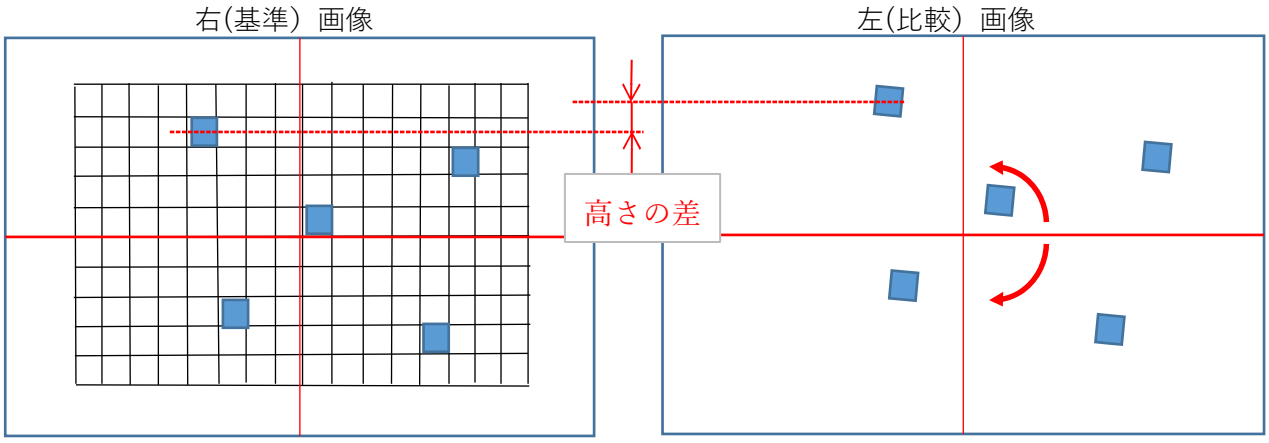
（２） 左（比較）画像から特徴点の対応点を探索
特徴点の領域のパターンと一致する領域を、左（比較）画像で探索します。
パターンマッチングを行い、一致率の低いものは、ズレ量の計測対象としません。



（3） 現在のズレ量を取得

右（基準）画像のそれぞれの特徴点と左（比較）画像のその対応点と高さ（行座標）の差に対して、画像の中心を軸に対応点を回転させて、高さの差を求めます。

すべての高さの差が同じなる回転角が回転のズレ量であり、そのときの高さの差が垂直ズレ量です。



（4） 補正量を取得

フレームごとに求めたズレ量を平均化します。

直近の複数フレームでバラツキがなければ、正しいズレ量と判断します。

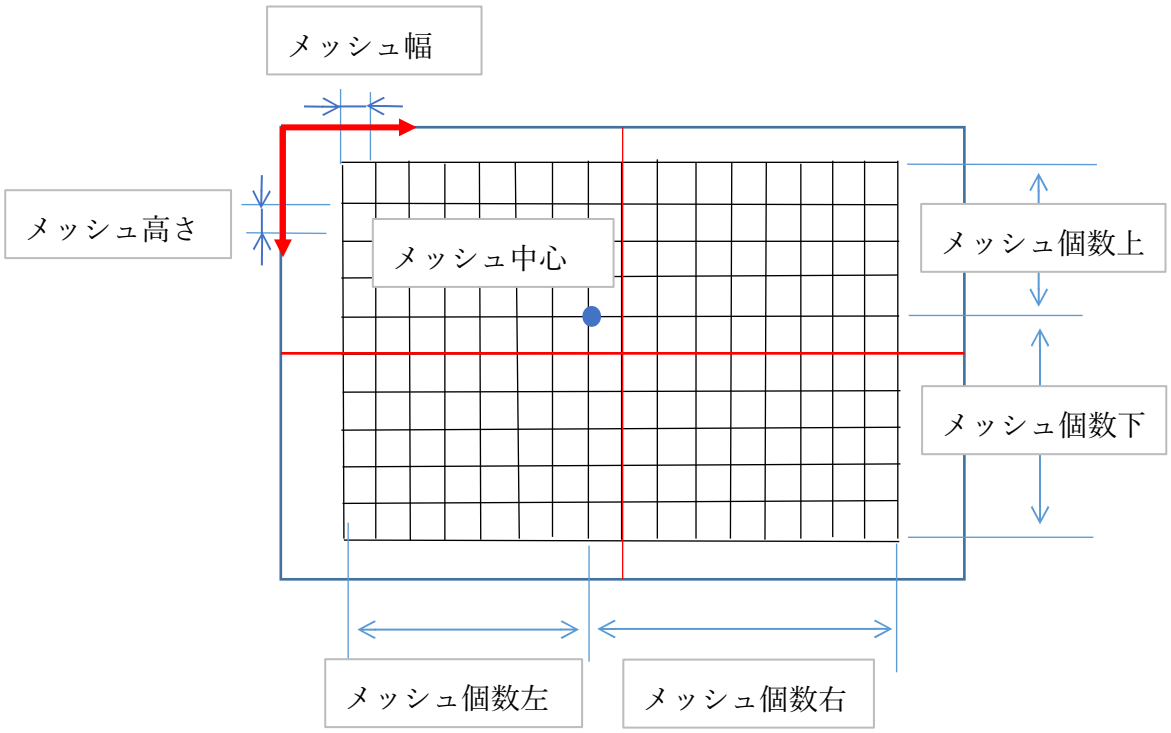
カメラの現在の補正量を取得し、ズレ量を加算します。これを新たな補正量として、カメラに保存します。

２．３　パラメータ

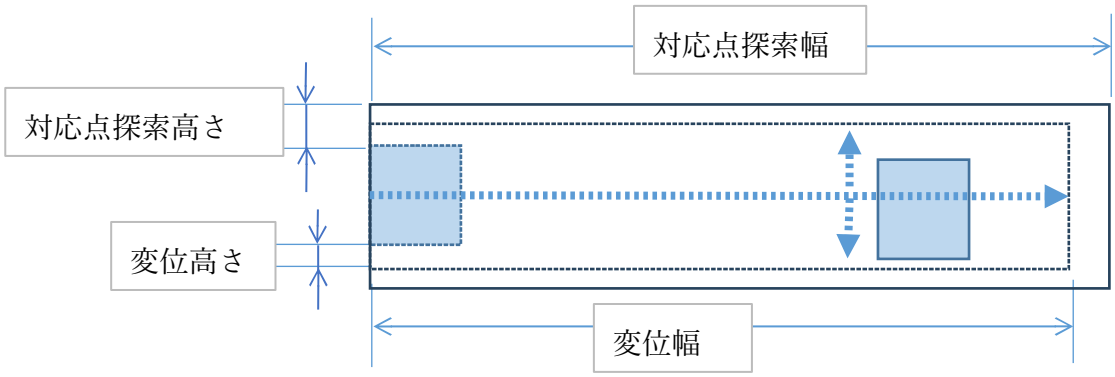
パラメータについて説明します。

（１）特徴点のサイズと対応点の探索範囲
以下のパラメータがあります。

・メッシュサイズ（高さ、幅）
・メッシュ中心座標（x 座標、y 座標）
・メッシュ個数（右、左、上、下）
・メッシュ展開領域座標(上、下、左、右)
・対応点探索領域（高さ、幅）



特徴点のサイズは、画像を小さな領域に分割（メッシュ分割）したときのメッシュのサイズです。
初めに、メッシュのサイズを指定します。適度な大きさであり、左（比較）画像から一対一で、高い一致率で見つけやすいサイズとします。
次にメッシュの中心位置を指定し、中心から上下、左右にメッシュの個数を指定します。
また、メッシュの展開領域を指定し、その領域をはみ出すメッシュを除外します。
対応点の探索は、左（比較）画像上に対して、特徴点のメッシュの座標を基準に指定された範囲で、上下と右方向に行います。
対応点は、特定の範囲内に存在し、範囲を広げ過ぎても効果がありません。



これらのパラメータは、カメラから出力される画像データの先頭を左上の座標原点の位置にします。
＊ISC100VM、ISC100XC では、カメラを逆さまにしているため、倒立画像が出力されます。

設定関数とパラメータ推奨値
setMeshParameter()

パラメータ	VM(480x752)	XC(720x1280)
メッシュサイズ（高さ、幅）	20、20	25、25
メッシュ中心座標（x 座標、y 座標）	376、240	480、360
メッシュ個数（右、左、上、下）	20、20、20、20	20、20、20、20
メッシュ展開領域座標（上、下、左、右）	16、460、16、640	16、703、16、1024
対応点探索領域（高さ、幅）	10、100	10、180

呼び出し例）
setMeshParameter(720, 1280, 25, 25, 480, 360, 20, 20, 20, 20, 16, 703, 16, 1024, 10, 180)

（2）特徴点と対応点の判定閾値
メッシュの中より特徴点の候補として、パターン強度の高いものを選別します。
パターン強度として、判定し易い明るさであること、鮮鋭であること、ランダムであることを評価します。
評価の項目は以下となります。

・平均輝度
・コントラスト
・エッジ比率

エッジ比率は、上下、斜めに隣り合う画素の輝度差が 3 以上ある画素の比率です。
パターンの変化を評価します。
対応点の評価では、初めに探索位置を評価します。
探索範囲はマージンを持ち、その内側で一致する領域が存在することを前提としています。
変位の高さ、幅で、ズレを許容できる範囲を指定します。
最後に、パターンの一致率が指定以上のものが対応点となり、特徴点が決まります。
以下が、閾値パラメータとなります。

・最小、最大平均輝度
・最小コントラスト
・最小エッジ比率
・最大変位（高さ、幅)
・最小一致率

設定関数とパラメータ推奨値
setMeshThreshold()

パラメータ	VM(480x752)	XC(720x1280)
最小、最大平均輝度	20、200	40、160
最小コントラスト	1000	1000
最小エッジ比率	50.0	30.0
最大変位（高さ、幅)	5、90	5、160
最小一致率	96.0	97.0

呼び出し例）
setMeshThreshold(40, 160, 1000, 30.0, 5, 160, 97.0)

階調補正された画像を使用する場合は、関数 setOperationMode()で指定します。
この関数は、設定された閾値を、階調補正に合わせて調整します。

（3）ズレ量のフレーム平均

初めに、そのフレームで求めたズレ量が妥当かどうか判断します。

ズレ量は、複数のフレームの妥当なズレ量を平均して求めます。

以下を満たしている場合に妥当と判断し、平均の対象します。

・そのフレームに十分な数の特徴点があること

・それぞれの特徴点と対応点の高さの差のバラツキが十分に小さいこと

直近の複数のフレームのズレ量で平均を取るため、使用するフレーム数を決めます。

パラメータは、次以下となります

・最小特徴点数

・最大垂直ズレ標準偏差

・ズレ量平均フレーム数

設定関数とパラメータ推奨値

setAveragingParameter()

パラメータ	VM(480x752)	XC(720x1280)
最小特徴点数	10	10
最大垂直ズレ標準偏差	0.5	1.0
ズレ量平均フレーム数	20	50

呼び出し例）

setAveragingParameter(10, 1.0, 50)

（4）平均ズレ量の判定

初めに、平均ズレ量を評価する前に、十分な回数だけ平均の計算を行っているか評価します。
十分な回数で算出できていれば、安定したズレ量が求められていると判断します。
次に、垂直ズレ量、回転ズレ量の判定基準を決めます。
求めた平均の垂直ズレ量、回転ズレ量が、それを超えた場合に補正を行います。
ただし、フレームごとの垂直ズレ量のバラツキが大きい場合は、補正をしないようにします。
ズレ量の判定パラメータは、以下です。

- 平均ズレ量算出回数
- 垂直ズレ量の判定基準
- 回転ズレ量の判定基準
- 標準偏差の判定基準

また、回転ズレ量を補正しない設定を設けます。

設定関数とパラメータ推奨値
setCriteria()

パラメータ	VM(480x752)	XC(720x1280)
平均ズレ量算出回数	100	100
垂直ズレ量の判定基準	0.1	0.1
回転ズレ量の判定基準	0.001	0.001
標準偏差の判定基準	0.25	0.25
回転ズレ補正	0	0
補正量の自動保存	1	1

呼び出し例）
setCriteria(100, 0.1, 0.001, 0.25, 0, 1)

5．SelftCalibration　Functions

一覧

関数	概要
initialize()	ソフトウェアキャリブレーションを初期化する
finalize()	ソフトウェアキャリブレーションを終了する
setMeshParameter()	メッシュパラメータを設定する
setMeshThreshold()	メッシュ閾値を設定する
setOperationMode()	動作モードを設定する
setAveragingParameter()	ズレ量平均パラメータを設定する
setCriteria()	平均ズレ量の判定基準を設定する
getMeshCoordinate()	メッシュ座標を取得する
GetMeshTextureStrength()	メッシュのパターン強度を取得する
getMatchSubpixelCoordinate()	対応点領域サブピクセル座標を取得する
clearCurrentMeshDifference()	現在のフレームの平行化処理結果をクリアする
getCurrentMeshDifference()	現在のフレームの平行化処理結果を返す
getMeshDifference()	最新フレームの左右画像のメッシュのズレ量を取得する
getAverageDifference()	左右画像のフレーム平均ズレ量を取得する
getCurrentCorrectValue()	現在の左右画像のズレ補正量を取得する
saveLatestCorrectValue()	最新のズレ補正量をカメラに保存する
parallelize()	左右画像を平行にする
start()	ステレオ平行化処理を開始する
stop()	ステレオ平行化処理を停止する

initialize()

ソフトウェアキャリブレーションを初期化する

static void SelfCalibration::initialize(int imghgt, int imgwdt);

引数

imghgt　補正画像の高さ(IN)

imgwdt　補正画像の幅(IN)

補足説明

指定されたサイズで処理バッファが確保される。

平行化処理をするスレッドが生成される。

終了時に、必ず 関数 finalize()を呼び出すこと

finalize()

ソフトウェアキャリブレーションを終了する

static void SelfCalibration::finalize();

setMeshParameter()

メッシュパラメータを設定する

static void SelfCalibration::setMeshParameter(int imghgt, int imgwdt,
　　int mshhgt, int mshwdt, int mshcntx, int mshcnty,
　　int mshnrgt, int mshnlft, int mshnupr, int mshnlwr,
　　int rgntop, int rgmbtm, int rgnlft, int rgnrgt, int srchhgt, int srchwdt);

引数

imghgt	補正画像の高さ(IN)
imgwdt	補正画像の幅(IN)
mshhgt	メッシュサイズ高さ(IN)
mshwdt	メッシュサイズ幅(IN)
mshcntx	メッシュ中心 x 座標(IN)
mshcnty	メッシュ中心 y 座標(IN)
mshnrgt	メッシュ右個数(IN)
mshnlft	メッシュ左個数(IN)
mshnupr	メッシュ上個数(IN)
mshnlwr	メッシュ下個数(IN)
rgntop	領域座標 Top(IN)
rgmbtm	領域座標 Bottom(IN)
rgnlft	領域座標 Left(IN)
rgnrgt	領域座標 Right(IN)
srchhgt	対応点探索領域の高さ(IN)
srchwdt	対応点探索領域の幅(IN)

setMeshThreshold()

メッシュ閾値を設定する

static void SelfCalibration::setMeshThreshold(int minbrgt, int maxbrgt, int mincrst, double minedgrt, int maxdphgt, int maxdpwdt, double minmtcrt);

引数

minbrgt	最小平均輝度(IN)
maxbrgt	最大平均輝度(IN)
mincrst	最小コントラスト(IN)
minedgrt	最小エッジ比率(IN)
maxdphgt	最大変位高さ IN)
maxdpwdt	最大変位幅(IN)
minmtcrt	最小一致率(IN)

setOperationMode()

動作モードを設定する

static void SelfCalibration::setOperationMode(int grdcrct);

引数

grdcrct	階調補正モードステータス 0:オフ 1:オン(IN)

setAveragingParameter()

ズレ量平均パラメータを設定する

static void SelfCalibration::setAveragingParameter(int minmshn, double maxdifdv, int avefrmn);

引数

minmshn	最小メッシュ特徴点数(IN)
maxdifstd	最大垂直ズレ標準偏差(IN)
avefrmn	ズレ量平均フレーム数(IN)

setCriteria()

平均ズレ量の判定基準を設定する

static void SelfCalibration::setCriteria(int frmcnt, double crtrdiff, double crtrrot, double crtrdev, int crctrot, , int crctsv);

引数

calccnt　ズレ量算出回数(IN)
crtrdiff　垂直ズレ量の判定基準(IN)
crtrrot回転ズレ量の判定基準(IN)
crtrdev　標準偏差の判定基準(IN)
crctrot　回転ズレ補正 0:しない 1:する(IN)
crctsv 補正量の自動保存 0:しない 1:する(IN)

getMeshCoordinate()

メッシュ座標を取得する

static int SelfCalibration::getMeshCoordinate(int **mshrgnx, int **mshrgny, int **srchrgnx, int **srchrgny);

引数

mshrgnx　メッシュ x 座標の配列ポインタ(IN)
mshrgny　メッシュ y 座標の配列ポインタ(IN)
srchrgnx　メッシュ探索領域 x 座標の配列ポインタ(IN)
srchrgny　メッシュ探索領域 y 座標の配列ポインタ(IN)

戻り値

メッシュの数を返す

getMeshTextureStrength()

メッシュのパターン強度を取得する

static void SelfCalibration::getMeshTextureStrength(bool **mshstrgt, int **mshbrgt, int **mshcrst,
　　double **mshedgx, double **mshedgd);

引数

mshstrgt	パターン強度判定(OUT)
mshbrgt	メッシュ輝度の配列ポインタ(OUT)
mshcrst	メッシュコントラストの配列ポインタ(OUT)
mshedgx	メッシュ輝度エッジ比率（縦横）の配列ポインタ(OUT)
mshedgd	メッシュ輝度エッジ比率（斜め）の配列ポインタ(OUT)

getMatchSubpixelCoordinate()

対応点領域サブピクセル座標を取得する

static int SelfCalibration::getMatchSubpixelCoordinate(bool **mshmtc, double **mtcrt, double **mtcsubx, double **mtcsuby);

引数

mshmtch	メッシュパターン一致判定の配列ポインタ(OUT)
mtcrt	パターン一致率の配列ポインタ(OUT)
mtcsubx	一致領域サブピクセル x 座標の配列ポインタ(OUT)
mtcsuby	一致領域サブピクセル y 座標の配列ポインタ(OUT)

戻り値

　　パターン一致したメッシュの数を返す

clearCurrentMeshDifference()

現在のフレームの平行化処理結果をクリアする

補足説明

　　それまでの平行化処理結果の記録がクリアされる。

getCurrentMeshDifference()

現在のフレームの平行化処理結果を返す

static int SelfCalibration::getCurrentMeshDifference(double *difvrt, double *difrot, double *difvstd);

引数

difvrt　左右画像の縦方向差分(OUT)

difrot　左右画像の回転差分(OUT)

difvstd　　左右画像の縦方向差分の標準偏差(OUT)

戻り値

パターン一致したメッシュの数を返す

getMeshDifference()

最新フレームの左右画像のメッシュのズレ量を取得する

static void SelfCalibration::getMeshDifference(double **mshdif, double *difvrt, double *difrot, double *difvstd);

引数

mshdif　　メッシュごとの縦方向差分の配列ポインタ(OUT)

difvrt　左右画像の縦方向差分(OUT)

difrot　左右画像の回転差分(OUT)

difvstd　　左右画像の縦方向差分の標準偏差(OUT)

補足説明

この関数は、ステレオ平行化処理中の場合、現在のフレームの処理が完了するまでブロックされる。

getAverageDifference()

左右画像のフレーム平均ズレ量を取得する

static int SelfCalibration::getAverageDifference(double *avedvrt, double *avedrot, double *avedvstd);

引数

avedvrt　上下ズレ差（画素）(OUT)
avedrot　回転ズレ差（ラジアン）(OUT)
avedvstd　上下ズレ差標準偏差（画素）(OUT)

引数

平均算出トータルフレームカウントを返す

getCurrentCorrectValue()

現在の左右画像のズレ補正量を取得する

static void SelfCalibration::getCurrentCorrectValue(double *vrtdif, int *vrtval, double *rotdif, int *rotval);

引数

vrtdif 上下ズレ補正量（画素）(OUT)
vrtval 上下並進レジスタ値（1/16 画素）(OUT)
rotdif 回転ズレ補正量（ラジアン）(OUT)
rotval 回転レジスタ値（1/(256 or 376 * 16)勾配画素）(OUT)

saveLatestCorrectValue()

最新のズレ補正量をカメラに保存する

static void SelfCalibration::saveLatestCorrectValue();

parallelize()

左右画像を平行にする

static void SelfCalibration::parallelize(unsigned char *pimgref, unsigned char *pimgcmp);

引数

pimgref　基準補正画像(IN)
pimgcmp　比較補正画像(IN)

start()

ステレオ平行化処理を開始する
static void SelfCalibration::start();

stop()

ステレオ平行化処理を停止する
static void SelfCalibration::stop();

改版履歴

Rev	Date	Content
0.0.1	2023/10/30	初版発行
0.0.2	2023/11/6	語句変更

End of Document