

ISC ソフトウェアライブラリ

説明書 C

【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

Copyright 2023 ITD Lab 株式会社

本マニュアルで使用されている各会社名、各製品名は各社の商標あるいは登録商標です

目次

1. 概要.....	3
1. 1 機能.....	4
1. 2 関数の使用手順.....	5
1. 3 指定された座標のデータ取得.....	6
1. 4 指定された矩形のデータ取得.....	7
1. 5 IscMeasurement クラス関数.....	8
Initialize().....	9
Terminate().....	9
GetPositionDepth.....	10
GetPosition3D.....	10
GetAreaStatistics.....	11
改版履歴.....	13

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“距離計算・モジュール”について述べたものです。

① 距離計算・モジュール

指定された座標または矩形の距離を計算します

モジュールは、以下のクラス名で提供されています。

処理ライブラリ名	クラス名	DLL 名
距離計算・モジュール	IscMeasurement	IscDplMainControl.dll

1. 1 機能

(1) 指定位置の距離取得

指定された基準画像上の座標における、視差と距離(m)を取得します

(2) 指定位置の 3D 情報の取得

指定された基準画像上の座標における、距離(m)と画像中心からの X/Y 距離 (m) を取得します

(2) 指定矩形の統計データ取得

指定された基準画像上の矩形における以下のデータを取得します

- ・ 指定矩形内の視差の最大値、最小値、平均値、中央値、最頻値、分散
- ・ 指定矩形内の距離の最大値、最小値、平均値、中央値、最頻値、分散
- ・ 指定矩形の幅(m)、高さ(m)、距離(m) ※距離は最頻値を使用

1. 2 関数の使用手順

距離計算・モジュールを使用する場合は、開始時と終了時に次の関数を呼び出します。

(1) 開始時に呼び出す関数

- ・ Initialize()

(2) 終了時に呼び出す関数

- ・ Terminate()

1. 3 指定された座標のデータ取得

視差は指定された座標のデータを取得します。

指定された座標に視差が無い場合には、周辺 4x4 を探索し、視差があればその値を採用します。

距離は、取得した視差より下記にて計算します。

計算する距離を $Z(m)$ とし、取得した視差を D とします。

カメラの固有パラメータを以下とします。

- ・基線長 B
- ・固有パラメータ $D_{INF} \cdot BF$ (カメラより取得します)

距離 Z は、以下となります。

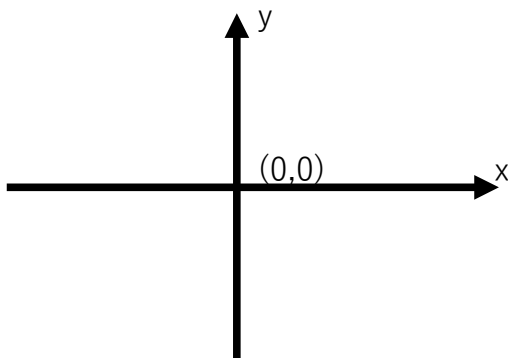
$$Z = \frac{BF}{(D - D_{INF})}$$

座標 (X, Y) の画像中心 (X_c, Y_c) からの距離 (X_z, Y_z) は以下とります。

$$X_z = \frac{B \times (X - X_c)}{(D - D_{INF})}$$

$$Y_z = \frac{B \times (Y_c - Y)}{(D - D_{INF})}$$

この時の座標系は、以下です。



1. 4 指定された矩形のデータ取得

指定された矩形のデータを取得します。

(1) 視差データ

- ①最大値：矩形内の最大値です
- ②最小値：矩形内の最小値です
- ③平均値：矩形内の算術平均です
- ④中央値：データを昇順に並べた中央の値です
- ⑤最頻値：視差のヒストグラムを計算し、最も頻度の高い値です。

カメラ固有パラメータ D_{INF} より小さい値は無効データとして無視します

- ⑥標準偏差：一般的な標準偏差です

ただし、カメラ固有パラメータ D_{INF} より小さい値は無効データとして無視します

(2) 距離データ

視差より距離を計算し、視差データと同様にデータを求めます。

(3) 矩形までの距離(m)及び、矩形の幅(m)、高さ(m)

代表値として視差データの最頻値を使い、矩形までの距離と、矩形の幅(m)、高さ(m)を求めます。

矩形の幅 $W(m)$ 、高さ $H(m)$ は以下のように求めます。

矩形の左上を (X_1, Y_1) 、右下を (X_2, Y_2) とします。

$$W = \frac{B \times |(X_1 - X_2)|}{(D - D_{INF})}$$
$$H = \frac{B \times |(Y_1 - Y_2)|}{(D - D_{INF})}$$

1. 5 IscMeasurement クラス関数

一覧

関数	概要
Initialize	初期化を行います
Terminate	終了処理を行います
GetPositionDepth	指定された座標の視差と距離(m)を取得します
GetPosition3D	指定された座標の、距離(m)と画面中心からの距離(m)を取得します
GetAreaStatistics	指定された矩形の情報を取得します ・ 視差の最大値、最小値、平均、中央値、最頻値、標準偏差 ・ 距離(m)の最大値、最小値、平均、中央値、最頻値、標準偏差 ・ 矩形までの距離(m)及び矩形の幅(m)、高さ(m)

Initialize()

IscMeasurement の初期化を行います

```
int Initialize(const int max_width, const int max_height);
```

引数

max_width 視差データ幅

max_height 視差データ高さ(IN)

補足説明

基準画像が視差データより大きい場合は、基準画像のサイズを入力します

Terminate()

IscMeasurement の終了処理を行います

```
int Terminate();
```

引数

補足説明

GetPositionDepth

指定された座標の視差と距離を取得します

```
iint GetPositionDepth(const int x, const int y, const IscImageInfo* isc_image_info,  
float* disparity, float* depth);
```

引数

x 指定座標 (X)

y 指定座標 (Y)

isc_image_info 入力データ・GetCameraData()又は GetDataProcModuleData()

で取得する

disparity 取得した視差

depth 取得した距離(m)

補足説明

指定位置に視差が無い場合又は、カメラ固有パラメータ D_{INF} より小さい値は無効として 0 となります

GetPosition3D

指定された座標の、距離(m)と画面中心からの距離(m)を取得します

```
int GetPosition3D(const int x, const int y, const IscImageInfo* isc_image_info, float*  
x_d, float* y_d, float* z_d);
```

引数

x 指定座標 (X)

y 指定座標 (Y)

isc_image_info 入力データ・GetCameraData()又は GetDataProcModuleData()

で取得する

x_d 取得した画面中心からの距離 X(m)

y_d 取得した画面中心からの距離 Y(m)

z_d 取得した画面中心からの距離(m)

補足説明

指定位置に視差が無い場合又は、視差がカメラ固有パラメータ D_{INF} より小さい値は無効として 0 となります

GetAreaStatistics

指定された矩形の情報を取得します

```
int GetAreaStatistics(const int x, const int y, const int width, const int height, const
IscImageInfo* isc_image_info, IscAreaDataStatistics* isc_data_statistics)
```

引数

x	指定矩形左上 (X)
y	指定矩形左上 (Y)
width	指定矩形幅
height	指定矩形高さ
isc_image_info	入力データ・GetCameraData()又は GetDataProcModuleData()

で取得する

isc_data_statistics	取得した矩形のデータ
---------------------	------------

補足説明

```
struct IscAreaDataStatistics {
    struct Statistics {
        float max_value;           /**< 最大値 */
        float min_value;           /**< 最小値 */
        float std_dev;             /**< 標準偏差 */
        float average;             /**< 平均 */
        float median;              /**< 中央値 */
        float mode;                /**< 最頻値 */
    };

    struct Roi3D {
        float width;               /**< 幅(m) */
        float height;              /**< 高さ(m) */
        float distance;            /**< 平均距離(m) */
    };

    int x;                        /**< 左上座標X */
    int y;                        /**< 左上座標Y */
    int width;                    /**< 矩形幅 */
    int height;                   /**< 矩形高さ */

    float min_distance;           /**< 有効距離 最小値 */
    float max_distance;           /**< 有効距離 最大値 */

    Statistics statistics_depth;   /**< 視差データ */
    Statistics statistics_distance; /**< 距離データ */
    Roi3D roi_3d;                 /**< 矩形データ */
};
```

※有効距離 最小値、最大値

距離のデータを集計する場合の範囲をします。

最大値、最小値、平均などは、この範囲内のデータのみで計算します。

通常は、表示している距離の範囲を指定します。

改版履歴

Rev	Date	Content
0.0.1	2023/12/28	初版発行

End of Document