

Appendix:eyemLib 共通の構造体

(Ver.1.4)

2019 年 8 月

株式会社 アイディール

■ 座標系関連定義

```
typedef struct{
    double      dX;          // X 座標
    double      dY;          // Y 座標
} EyemOcsDXY;
```



```
typedef struct{
    double      dX;          // X 座標
    double      dY;          // Y 座標
    double      dZ;          // Z 座標
} EyemOcsDXYZ;
```



```
typedef struct{
    double      dX;          // X 座標
    double      dY;          // Y 座標
    double      dQ;          //  $\theta$ 
} EyemOcsDXYQ;
```



```
typedef struct{
    double      dX;          // 中心の X 座標
    double      dY;          // 中心の Y 座標
    double      dR;          // 半径
} EyemOcsDXYR;          // 円の表現に使用
```



```
typedef struct{
    double      dA;          // a
    double      dB;          // b
    double      dC;          // c
} EyemOcsDABC;          // 直線 ( $ax + by + c = 0$ ) の表現に使用
```



```
typedef struct{
    double      dX;          // X 座標
    double      dY;          // Y 座標
    double      dQ;          // 回転角  $\theta$  (単位:rad)
    double      dS;          // スケール(相似比)
} EyemOcsDXYQS;          // マッチングに使用
```

```

typedef      struct{
    double      dA;          //  $a$ 
    double      dB;          //  $b$ 
    double      dC;          //  $c$ 
    double      dD;          //  $d$ 
}      EyemOcsDABCD;        // 平面( $ax + by + cz + d = 0$ )の表現に使用

typedef  struct{
    double      dXo;          // 中心の  $X$  座標
    double      dYo;          // 中心の  $Y$  座標
    double      dL;          // 長軸半径
    double      dS;          // 短軸半径
    double      dQ;          // 長軸の傾き(単位:rad)
}      EyemOcsDXYLSQ;      // 楕円の表現に使用

typedef  struct{
    EyeqOcsDXYZ  tP;          // 直線上の1点の座標
    EyeqOcsDXYZ  tV;          // 直線の方角ベクトル
}      EyemOcsDPV;          // 3次元空間内の直線の表現に使用

typedef  struct{
    EyeqOcsDXYZ  tC;          // 中心の座標
    EyeqOcsDXYZ  tR;          // 軸の半径(dX:長軸, dY:中軸, dZ:短軸)
    double      dU;          // 長軸を  $xy$  平面へ射影した直線の  $x$  軸からの角  $u$ 
                                (単位:rad,  $0 \leq u \leq \pi$ )
    double      dV;          // 長軸の  $xy$  平面からの角  $v$  (単位:rad,  $-\pi/2 \leq v \leq \pi/2$ )
    double      dW;          // 長軸まわりの回転角  $w$  (単位:rad,  $-\pi/2 \leq w \leq \pi/2$ )
}      EyemOcsDCRUVW;      // 楕円体の表現に使用

```

■ 画像情報

```

typedef  struct{
    unsigned char  *ucpImage; // 先頭アドレス
    int            iWidth;     // 画像メモリ  $X$  方向サイズ
    int            iHeight;    // 画像メモリ  $Y$  方向サイズ
}      EyemImage;

```

■ 矩形定義

```
typedef struct{
    int          iXs;          // 始点(左上) X 座標
    int          iYs;          // 始点(左上) Y 座標
    int          iWidth;       // x 方向サイズ(幅)
    int          iHeight;      // y 方向サイズ(高さ)
} EyemRect;
```

```
typedef struct{
    int          iXs;          // 始点(左上) X 座標
    int          iYs;          // 始点(左上) Y 座標
    int          iXe;          // 終点(右下) X 座標
    int          iYe;          // 終点(右下) Y 座標
} EyemRect2;
```

■ 2値ブロップ解析

```
typedef struct {
    int          iArea;        // 面積
    double       dCenterX;     // 重心 x 座標
    double       dCenterY;     // 重心 y 座標
    int          iXs, iYs, iXe, iYe; // 外接長方形(始点, 終点)
    int          iWidth, iHeight; // 外接長方形(x 方向サイズ(幅), y 方向サイズ(高さ))
    double       dTheta;       // 主軸傾斜角(rad)
} EyemBinBlob;                // ブロップ解析の結果格納に使用
```

■ カメラキャリブレーション

```
typedef struct {
    int          iPtnRefN;     // パターン参照点の個数
    EyemOcsDXY   *tpObjPt;     // パターン参照点の実座標(単位:mm)
    EyemOcsDXY   *tpImgPt;     // パターン参照点の画像座標(単位:画素)
} EyemCalibPtn;               // キャリブレーション用パターンデータの表現に使用
```

```

typedef struct {
    double      daA[3][3];      // 内部パラメータ行列
    double      dFx, dFy;       // 焦点距離×スケール(単位:画素)
    double      dFs;            // 焦点距離×せん断係数(単位:画素)
    double      dUo, dVo;       // 画像中心座標(単位:画素)
    double      dTheta;         // (未使用)画像面傾斜角(単位:rad)
    double      dK1, dK2;       // ラジアル歪み係数
    double      dK3, dK4, dK5;   // ラジアル歪み係数
    double      dP1, dP2;       // 偏心歪み係数
    double      daInvA[3][3];    // 内部パラメータ行列の逆行列
    double      daRsv[5];       // 予備領域
} EyemCalibInt;               // カメラの内部パラメータの表現に使用

typedef struct {
    double      daR[3][3];      // 回転行列
    double      daT[3];         // 並進ベクトル
    double      daRdr[3];       // 回転ベクトル(ロドリゲスの表現)
} EyemCalibExt;               // カメラの外部パラメータの表現に使用

typedef struct {
    double      daH[3][3];      // ホモグラフィ行列 H
    double      daInvH[3][3];   // 行列 H の逆行列
} EyemCalibHom;               // ホモグラフィ行列の表現に使用

```

改訂履歴

Version No.	内 容
1.0	• 新規発行
1.1	• EyemOcsDCRUVW 構造体の追加
1.2	• 矩形構造体の変更および EyemBinBlob 構造体の追加
1.3	• EyemCalibInt 構造体へのせん断係数項の追加.
1.4	• EyemCalibInt 構造体への歪み係数の追加