

イメージ・アルファ画像処理ライブラリ QRGuide マニュアル

第 1.00 版

株式会社 イメージ・アルファ

はじめに

本ドキュメントは、イメージ・アルファ画像処理ライブラリ IALIB のQRコード位置計測機能(QRGuide)について解説しています。

QRコード復号/符号処理については、QRコード処理マニュアルを参照してください。

表 0-1 本ドキュメントの適用 IALIB バージョン

ソフト名	バージョン番号
IALIB	Version 1.2.2

表 0-2 QRGuide・修正履歴

バージョン番号	分類	修正内容
version 1.2.2	-	新規

目次

1. 機能.....	4
2. 使用方法.....	5
3. 関数.....	10
4. 列挙体、構造体.....	19
5. 補足.....	22

1. 機能

IALIB では、QRコードの3次元位置を計測する関数を用意しています (QRGuide)。

QRコードの物理サイズ、画像を作成したカメラの内部パラメータは既知とし、QRコードは平面上にあるものとして処理します。
歪んだ面にあるQRコードは、復号はできますが位置計測は失敗します。

位置計測の前処理に行われる復号処理については、QRコード処理マニュアルを参照してください。
復号だけ行う (つまり位置計測は行わない) 場合、QRコード処理マニュアルにあるQRコード符号関数だけで高速に行えます。

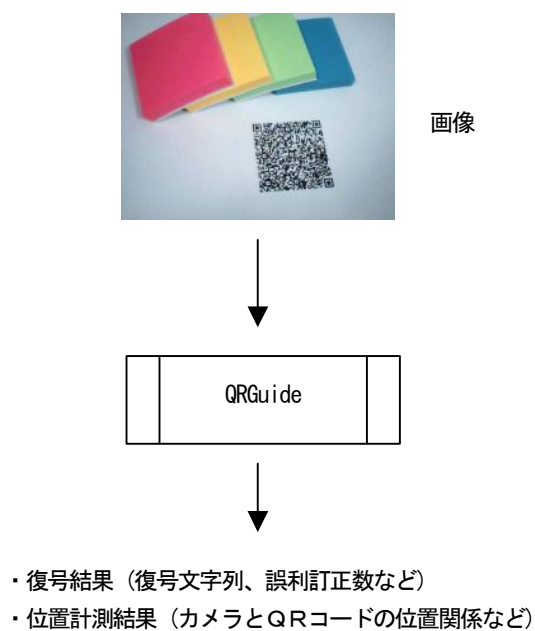


図 1-1 QRGuide 機能

2. 使用方法

下のサンプルコードは、BMP ファイルから画像を読み込み、その画像にあるQRコードを復号、位置計測し、メッセージボックスで結果を表示するアプリケーションです。

```
int APIENTRY WinMain( HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow )
{
    int i;
    int imgID;
    IA_Size imgSize;
    int qrNum;
    IA_QR_Decode_CtlExt ctID;
    IA_QRGuide_Ctl ctIG;
    IA_QR_Decode_Rslt rsltD;
    IA_QRGuide_Rslt rsltG;
    char message[100];
    char caption[100];

    // IALIB・有効化
    IA_Open(); → 2.1.1節

    // IALIB・QRGuide・有効化
    IA_QRGuide_Open(); → 2.1.1節

    // 処理対象画面確保
    imgSize.xSize = 640;
    imgSize.ySize = 480;
    imgID = IA_AllocImg( IA_IMG_CHAR, &imgSize );

    // BMPファイルを読み込み処理対象画面へ格納
    IA_BMPFile_Load( imgID, "QRImg.bmp" );

```

⋮

▪
▪
▪

```
// 復号制御パラメータ設定
IA_QR_GetInitCtl( &ctID );           // ctIDを初期化
ctID.fMicroQR = 1;                   // マイクロQRコードも復号
ctID.maxNum = IA_QR_MAXQRNUM;       // 複数のQRコードを復号
IA_QR_SetCtl( &ctID );              // 設定 → 2.1.2節
```

```
// 位置計測制御パラメータ設定
IA_QRGuide_GetInitCtl( &ctIG );     // ctIGを初期化
ctIG.moduleSizeX = 3.517000F;       // QRコードのモジュールサイズ[mm]
ctIG.moduleSizeY = 3.517000F;
ctIG.camParam.imgSize.xSize = 640;  // カメラ内部パラメータ
ctIG.camParam.imgSize.ySize = 480;
ctIG.camParam.trCtoI[0] = 778.210094F;
ctIG.camParam.trCtoI[1] = 4.657677F;
ctIG.camParam.trCtoI[2] = 322.376346F;
ctIG.camParam.trCtoI[3] = 0;
ctIG.camParam.trCtoI[4] = 782.128798F;
ctIG.camParam.trCtoI[5] = 225.025061F;
ctIG.camParam.trCtoI[6] = 0;
ctIG.camParam.trCtoI[7] = 0;
ctIG.camParam.trCtoI[8] = 1;
ctIG.camParam.lensDistCf.k1 = 0;
ctIG.camParam.lensDistCf.k2 = 0;
ctIG.camParam.lensDistCf.p1 = 0;
ctIG.camParam.lensDistCf.p2 = 0;
IA_QRGuide_SetCtl( &ctIG );        // 設定 → 2.1.3節
```

```
// 復号、位置計測
qrNum = IA_QRGuide_Exec( imgID );   // → 2.1.4節
```

▪
▪
▪

⋮

```

// 結果をメッセージボックスで表示
if( qrNum > 0 ){
    for( i=0; i<qrNum; i++){
        // 復号文字列
        IA_QR_GetRslt( i, &rsltD ); // 復号結果取得 → 2.1.5節
        sprintf( caption, "%d番目復号文字列", i );
        MessageBox( NULL, rsltD.str, caption, MB_OK );
        // QRコードとカメラの距離
        IA_QRGuide_GetRslt( i, &rsltG ); // 位置計測結果取得 → 2.1.6節
        sprintf( caption, "%d番目距離", i );
        sprintf( message, "%g", rsltG.lngQRtoCM );
        MessageBox( NULL, message, caption, MB_OK );
    }
}
else if( qrNum == 0 ){
    MessageBox( NULL, "復号失敗(検知数)", "復号失敗(検知数)", MB_OK );
}
else{
    MessageBox( NULL, "エラー", "エラー", MB_OK );
}

// IALIB・QRGuide・無効化
IA_QRGuide_Close(); → 2.1.1節

// IALIB・無効化
IA_Close(); → 2.1.1節

return 0;
}

```

2.1.1. 各種有効化

IALIB全体の有効化、QRGuideの有効化を最初に行います。無効化されるまで機能が使用できます。
IA_QRGuide_Open (3. 1. 1節) を呼び出した場合、IA_QR_Openの呼び出しは不要です。

有効時には何回でも位置計測が行えます。位置計測のたびに有効化をする必要はありません。

2.1.2. 復号用制御パラメータの設定

構造体 IA_QR_Decode_CtlExt の変数を初期化して、環境に合わせてカスタマイズし、
関数 IA_QR_SetCt の引数とします。

復号だけを行う場合と同様の手順となります。詳細はQRコード処理マニュアルを参照してください。

2.1.3. 位置計測用処理パラメータの設定

構造体 IA_QRGuide_Ctl (4. 1 節) の変数を初期化して、環境に合わせてカスタマイズし、
関数 IA_QRGuide_SetCtl (3. 2. 1 節) の引数とします。

2.1.4. 復号、位置計測

関数 IA_QRGuide_Exec (3. 3. 2 節) により、これまでの制御パラメータ設定値に基づいた復号、位置計測を行います。

このサンプルでは画像上のQRコードすべてを復号、位置計測しますが、
例えば以下手順により、復号文字列の先頭が ABC のQRコードだけを位置計測することもできます。

- ・関数 IA_QR_Exec で復号のみ行う
- ・復号できた全QRコードから、復号文字列の先頭が ABC のものを探す
- ・このQRコードだけを関数 IA_QRGuide_Calc3D (3. 3. 1 節) で位置計測する

2.1.5. 復号結果の取得

関数 `IA_QR_GetResult` により QRコード 1 つ分の復号結果を取得できます。

QRコードの指定には QRコード ID を使用します。

QRコード ID は検知した QRコード に割り振られた番号で、0 以上検知数未満の整数値です。

復号だけを行う場合と同様の手順となります。詳細は QRコード処理マニュアルを参照してください。

2.1.6. 位置計測結果の取得

関数 `IA_QRGuide_GetResult` (3.4.1 節) により QRコード 1 つ分の位置計測結果を取得できます。

QRコードの指定には QRコード ID を使用します。

QRコード ID は検知した QRコード に割り振られた番号で、0 以上検知数未満の整数値です。

3. 関数

本章では関数インターフェースを記しています。
 関連する列挙体、構造体は4章を参照してください。

区分	節番号	関数名	機能
有効/無効	3.1.1 節	IA_QRGuide_Open	有効化
	3.1.2 節	IA_QRGuide_Close	無効化
制御パラメータ	3.2.1 節	IA_QRGuide_SetCtl	制御パラメータの設定
	3.2.2 節	IA_QRGuide_GetCtl	制御パラメータの取得
	3.2.3 節	IA_QRGuide_GetInitCtl	制御パラメータデフォルト値の取得
位置計測	3.3.1 節	IA_QRGuide_Calc3D	位置計測
	3.3.2 節	IA_QRGuide_Exec	復号および位置計測
処理結果	3.4.1 節	IA_QRGuide_GetRslt	処理結果の取得

表 3-1 QRコード処理一覧

3.1. 有効/無効

3.1.1. 有効化 [IA_QRGuide_Open]

インタフェース

```
int IA_QRGuide_Open ();
```

リターン値

基本エラーコード

機能

QRコード位置計測機能(QRGuide)を有効にします。

本関数の呼び出しは、IALIB を有効化した後(つまり関数 IA_Open を呼んだ後)に行ってください。

本関数を呼び出した場合、QRコード復号/符号機能も有効となります。関数 IA_QR_Open は呼び出し不要です。

3.1.2. 無効化 [IA_QRGuide_Close]

インタフェース

```
int IA_QRGuide_Close ();
```

リターン値

基本エラーコード

機能

QRコード位置計測機能(QRGuide)を無効にします。

IALIB 内部にて確保したメモリの開放などの後処理を行いますので、有効にした場合には本関数を必ず呼び出してください。

本関数の呼び出しは、IALIB を無効化する前(つまり関数 IA_Close を呼ぶ前)に行ってください。

3.2. 制御パラメータ

3.2.1. 制御パラメータの設定 [IA_QRGuide_SetCtl]

インタフェース

```
int IA_QRGuide_SetCtl (
    IA_QRGuide_Ctl *pCtl
);
```

パラメータ

名称	タイプ	入/出	許容値	意味
pCtl	IA_QRGuide_Ctl* (4.1節)	入力		制御パラメータ

リターン値

基本エラーコード

機能

制御パラメータを設定します。この設定値で位置計測(3.3節)が行われます。

本関数を呼び出さずに復号する場合はデフォルト値で動作します。

3.2.2. 制御パラメータの取得 [IA_QRGuide_GetCtl]

インタフェース

```
int IA_QRGuide_GetCtl (
    IA_QRGuide_Ctl *pCtl
);
```

パラメータ

名称	タイプ	入/出	許容値	意味
pCtl	IA_QRGuide_Ctl* (4.1節)	出力		制御パラメータ

リターン値

基本エラーコード

機能

現在設定されている制御パラメータを取得します。

3.2.3. 制御パラメータデフォルト値の取得 [IA_QRGuide_GetInitCtl]

インタフェース

```
int IA_QRGuide_GetInitCtl (
    IA_QRGuide_Ctl *pCtl
);
```

パラメータ

名称	タイプ	入/出	許容値	意味
pCtl	IA_QRGuide_Ctl* (4.1 節)	出力		制御パラメータデフォルト値

リターン値

基本エラーコード

機能

制御パラメータのデフォルト値を取得します。
主に構造体の初期化に使用します。

3.3. 位置計測

3.3.1. 位置計測 [IA_QRGuide_Calc3D]

インタフェース

```
int IA_QRGuide_Calc3D (
    );
int IA_QRGuide_Calc3D (
    int qrID
    );
```

パラメータ

名称	タイプ	入/出	許容値	意味
qrID	int	入力	0~検知数-1	QRコード ID

リターン値

基本エラーコード

機能

QRコードの復号結果から、位置計測を行います。

本関数では復号は行いません。画像を入力する代わりに、直前の復号結果が使用されます。

QRコード ID は検知したQRコードに割り振られた番号で、0 以上検知数未満の整数値です。

引数でQRコード ID を指定しなかった場合、全てのQRコードにて位置計測を実行します。

3.3.2. 復号、位置計測 [IA_QRGuide_Exec]

インタフェース

```
int IA_QRGuide_Exec (
    int imgID
);
```

パラメータ

名称	タイプ	入/出	許容値	意味
imgID	int	入力		処理対象画像

リターン値

0 以上 : 検知QRコード数
 0 未満 : 基本エラーコード

機能

画像より、QRコード検知、復号、位置計測を行います。

全てのQRコードにて位置計測を実行します。

3.4. 処理結果

3.4.1. 処理結果の取得 [IA_QRGuide_GetRslt]

インタフェース

```
int IA_QRGuide_GetRslt (
    int qrID, IA_QRGuide_Rslt *pRslt
);
```

パラメータ

名称	タイプ	入/出	許容値	意味
qrID	int	入力	0~検知数-1	QRコード ID
pRslt	IA_QRGuide_Rslt* (4.2節)	出力		処理結果

リターン値

基本エラーコード

機能

位置計測結果を取得します。

QRコード ID は検知したQRコードに割り振られた番号で、0以上検知数未満の整数値です。

4. 列挙体、構造体

4.1. 制御パラメータ

```
typedef struct {
    int mrgn; // 周辺mrgn モジュールに誤りがないモジュールを位置計測に使用。
             // 0 なら誤りでなければ周辺に関係なく使用。

    float alpha; // 孤立モジュールに対する、その他モジュールの位置計測時の重み係数。
                // 0 なら使用しない。1 なら同等の扱い。

    float grad_tvecMag; // 最適化時の並進ベクトルの回転ベクトル成分に対する倍率。0 なら自動計算
    float grad_loopNum; // 最適化時のループ回数
    float moduleSizeX; // モジュールの横サイズ[mm]。複数の復号 QR コードに共通の値なので注意！
    float moduleSizeY; // モジュールの縦サイズ[mm]。複数の復号 QR コードに共通の値なので注意！
    IA_CameraParam camParam; // カメラパラメータ。
                             // 内部パラメータ(下記)のみ使用され、外部パラメータには設定不要。
                             // ・imgSize : 解像度
                             // ・trCtoI [9] : 内部パラメータ行列
                             // ・lensDistCf : レンズ歪み係数。
                             // 現在レンズ歪みは非対応。0を設定する。
                             // 処理対象画像には事前にレンズ歪みを除去する必要有り。
} IA_QRGuide_Ctl;
```

QRコード位置計測の制御パラメータを表します。

moduleSizeX、moduleSizeY、camParamは環境に応じて変更し、それ以外はデフォルト値を推奨します。
関数 IA_QRGuide_GetInitCtl (3.2.3 節)によりデフォルト値で初期化できます。(2.1.3 節)

現在、QRGuide 自体はレンズ歪みには対応していません。camParam.lensDistCf はすべて 0 を設定してください。
レンズ歪みのある画像に対して位置計測を行うには、事前に画像からレンズ歪みを除去する必要があります。(図
4-1)

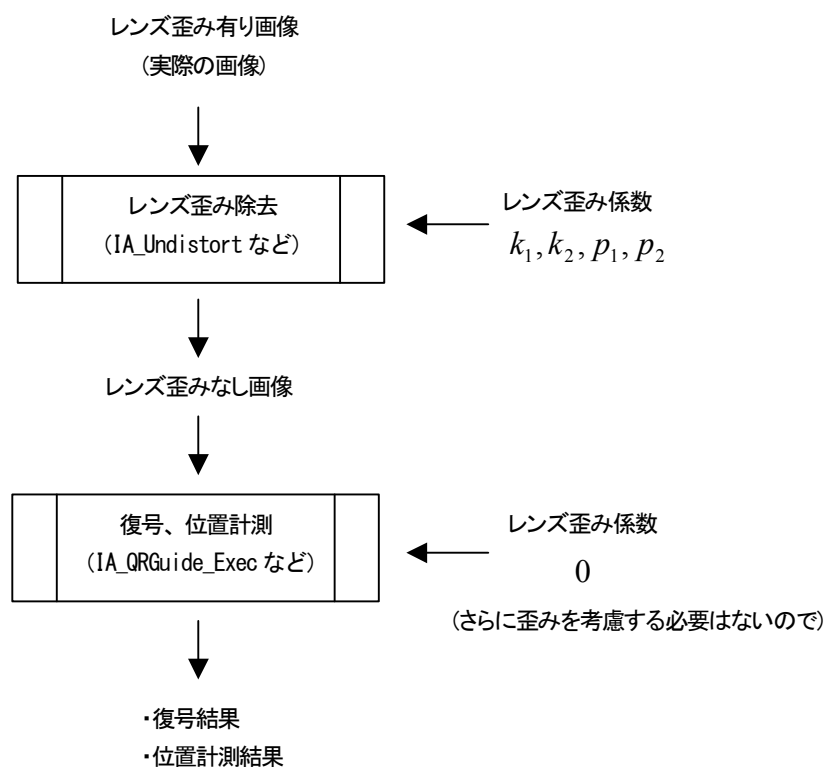


図 4-1 レンズ歪みのある画像に対する位置計測の手順

4.2. 位置計測結果

```
typedef struct {
    int          fCalib;    // 1:位置計測成功、:位置計測失敗(復号は成功)
    float        trMtoI[9]; // モジュール座標→画像座標の変換係数
    float        trItoM[9]; // 画像座標→モジュール座標の変換係数
    IA_PointF    vertexI[4]; // QRコード4隅の画像座標。[0]:左上、[1]:右上、[2]:左下、[3]:右下
    IA_CameraParam camParam; // カメラパラメータ。内部パラメータは制御パラメータの値がコピーされる。
    IA_AffTrCf3DF trCtoW;    // カメラ座標→世界座標の変換係数
    IA_AffTrCf3DF trWtoC;    // 世界座標→カメラ座標の変換係数
    float        rollDeg;    // ロール角[° ]
    float        pitchDeg;   // ピッチ角[° ]
    float        yawDeg;     // ヨー角[° ]
    float        height;     // 世界座標 z=0 平面とカメラの距離[mm]
    float        lngQRtoCM;  // QRコード中心とカメラの距離[mm]
    float        optimalValue; // 最適値。位置計測の正確さを表し、0に近いほど正確

    // 世界座標の定義
    //   ・QRコードのある面をXY平面とする
    //   ・原点は右下モジュールの中心
    //   ・X軸はQRコード左→右方向、Y軸は下→上方向
} IA_QRGuide_Rslt;
```

検出したQRコード1つの位置計測結果を格納します。

trMtoI、trItoMはモジュール座標と画像座標の間の3x3射影変換行列です。(5.2節)

モジュール座標、世界座標、カメラ座標の定義については、5.1節を参照してください。

5. 補足

5.1. 座標系

QRGuide では4つの座標を使用します。

5.1.1. モジュール座標

モジュールのQRコードにおける位置を表すための2次元座標です。

- ・原点 : 左上のモジュール(の中心)
- ・X方向 : 左→右の方向
- ・Y方向 : 上→下の方向
- ・スケール(単位) : 1つのモジュールの幅を1

ここでの上下左右は画像上での上下左右ではなく、QRコードが持つ絶対的な方向です。

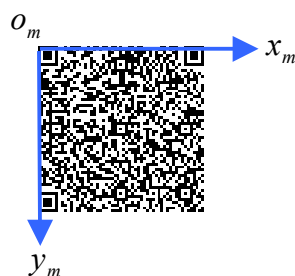


図 5-1 モジュール座標

5.1.2. 画像座標

IALIBにて共通に使われる画像座標と同じです。

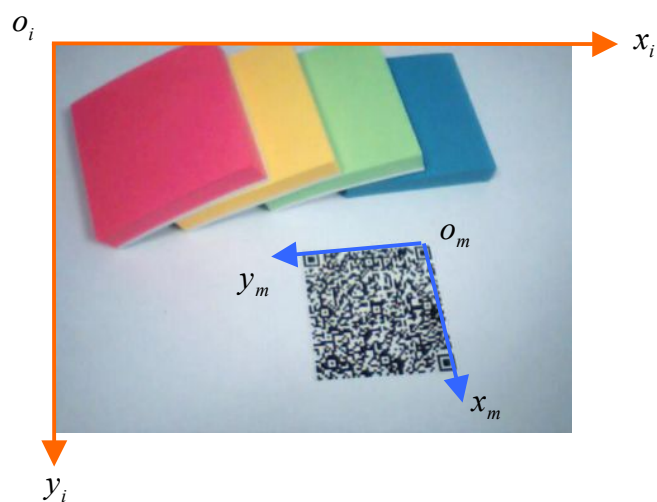


図 5-2 画像座標とモジュール座標

5.1.3. 世界座標

QRコードを基準とした3次元座標です。

- ・原点 : 右下のモジュール(の中心)
- ・X方向 : QRコード左→右の方向
- ・Y方向 : QRコード下→上の方向
- ・Z方向 : QRコードの向いている方向
- ・スケール(単位) : mm

ここでの上下左右は画像上や空間内での上下左右ではなく、QRコードが持つ絶対的な方向です。

5.1.4. カメラ座標

カメラを基準とした3次元座標です。

- ・原点 : カメラ位置
- ・X方向 : カメラ左→右の方向
- ・Y方向 : カメラ上→下の方向
- ・Z方向 : カメラ光軸方向
- ・スケール(単位) : mm

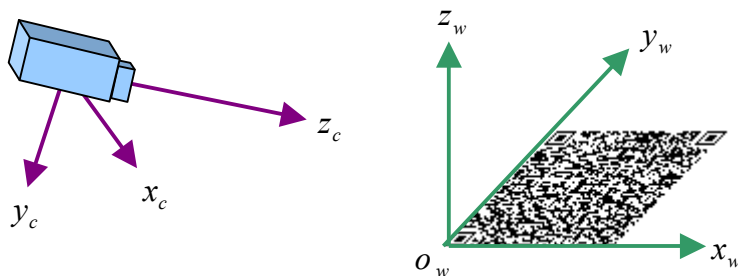


図 5-3 カメラ座標と世界座標

5.2. 射影変換

IALIB では 3x3 の射影変換行列を float 型の配列で表します。

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} trMtoI[0] & trMtoI[1] & trMtoI[2] \\ trMtoI[3] & trMtoI[4] & trMtoI[5] \\ trMtoI[6] & trMtoI[7] & trMtoI[8] \end{bmatrix} \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix}$$

```
float trMtoI[9];
```