

## サンプルソースコード (2値化アプリケーション)

以下は、2値化処理を行うアプリケーションです。

2値化処理固有のコードを書き換えるだけで、任意の画像処理アプリケーションを作成できます。

```

#include "IALIB.h"

// 画面サイズ
#define IMGXSIZE 320
#define IMGYSIZE 240

// 画面ID
static int g_imgID_bin;

// 制御パラメータ
static IA_BinarizeCtl g_binCtl;

// 関数宣言
int Exec();

/***** メイン関数 *****/
int WINAPI WinMain( HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow )
{
    MSG msg;
    IA_Size imgSize;

    IA_Open(); // IALIB・有効化
    IA_Eval_Enable_MultiPort( 1, 1 ); // IALIB・Eval有効化

    IA_Eval_RegisterExec( Exec ); // 評価関数登録

    // 画面確保
    imgSize.xSize = IMGXSIZE;
    imgSize.ySize = IMGYSIZE;
    g_imgID_bin = IA_AllocImg( IA_IMG_CHAR, &imgSize, "2値化画像" );

    // 制御パラメータ登録
    g_binCtl.bw = 0;
    g_binCtl.thr = 128;
    IA_Eval_RegisterParam( &g_binCtl.bw, "2値化", "明暗(g_binCtl.bw)", 1, 1, 0, 1, "0:閾値以上を輝度\n1:閾値未満を輝度" );
    IA_Eval_RegisterParam( &g_binCtl.thr, "2値化", "閾値(g_binCtl.thr)", 1, 1, 0, 255, "0~255:閾値" );

    IA_Eval_LoadIniFile( "sample_binarize.ini" ); // iniファイル読み込み

    IA_Eval_CreateWindow( IA_EVAL_VIEWSIZE_320x240, IA_EVAL_VIEWNUM_2x2, 0 ); // メインウィンドウ表示

    while( GetMessage( &msg, NULL, 0, 0 ) ){ // メッセージループ (メインウィンドウを閉じるとWM_QUITが発生し、ループ終了)
        TranslateMessage( &msg );
        DispatchMessage( &msg );
    }

    IA_Eval_SaveIniFile( "sample_binarize.ini" ); // iniファイル書き込み

    IA_Eval_Disable(); // IALIB・Eval無効化
    IA_Close(); // IALIB・無効化

    return 0;
}

/***** 評価関数 *****/
int Exec()
{
    int imgID_src = IA_Eval_GetSrcImgID(); // 入力画像を取得

    IA_Binarize( imgID_src, g_imgID_bin, &g_binCtl ); // 2値化処理

    return 0;
}

```

2値化  
固有処理

2値化  
固有処理

## 制御パラメータ設定

関数 IA\_Eval\_RegisterParam により、ソースコードの変数を GUI より編集することができます。(下図)  
 また、これら変数の値は ini ファイルに保存することができます。

**IA\_View.exe**

メニュー: ファイル(E) 領域取得(A) 表示(Y) データ出力形式(D) ヘルプ(H)

(1) メニュー「領域取得」  
 (2) マウスにて入力

**「ファイル名を選択」ダイアログ**

ファイル名(N): wave000000.bmp

**トラックバー**

25 S  
 59.000000 S

**制御パラメータ設定ダイアログ**

パラメータ名	値	単位	パラメータ名	値	単位
int型パラメータ	0		ファイル名属性パラメータ	123.bmp	F
float型パラメータ	10.000000		点属性パラメータX	20	1
char[]型パラメータ	abcde		点属性パラメータY	21	1
四角形属性パラメータAX	40	4	点列属性パラメータ		N
四角形属性パラメータAY	41	4	矩形属性パラメータSX	30	R
四角形属性パラメータBX	42	4	矩形属性パラメータSY	31	R
四角形属性パラメータBY	43	4	矩形属性パラメータEX	32	R
四角形属性パラメータCX	44	4	矩形属性パラメータEY	33	R
四角形属性パラメータCY	45	4	色属性パラメータX	50	C
四角形属性パラメータDX	46	4	色属性パラメータY	51	C
四角形属性パラメータDY	47	4	色属性パラメータZ	52	C

OK CLOSE SET INIT LOAD SAVE

**設定方法は3種類**

- ・エディットボックス
- ・トラックバー
- ・IA\_View よりマウス入力(座標など)

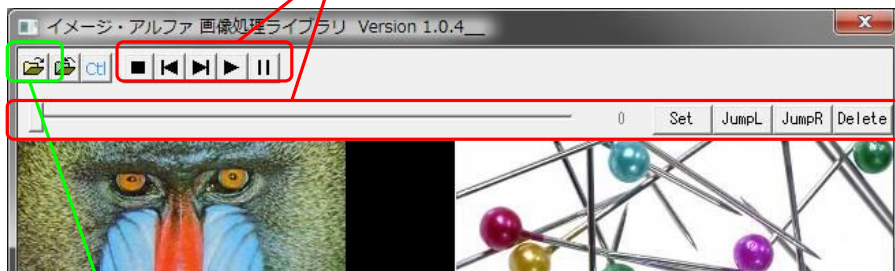
**イメージ・アルファ 画像処理ライブラリ Version 0.9.2b**

20 Set JumpL JumpR Delete

## 画像入力設定、制御

- ・専用ダイアログより IALIB 対応の全画像デバイスを選択。(AVI、MPEG、BMP、PPM、JPEG、DirectShow 対応カメラ、ネットワークカメラ)
- ・ステレオ画像処理にも対応
- ・ツールバー、トラックバーにより、ファイル逆再生、指定位置ジャンプなどの制御も可能。

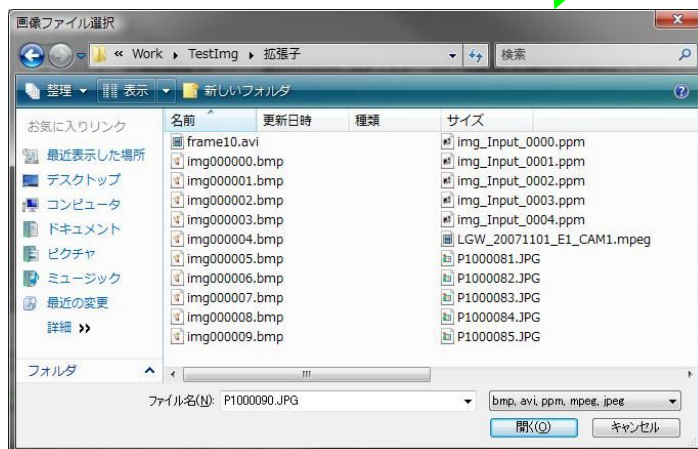
画像再生制御。1フレーム読み込むたびに、ユーザー定義の関数が呼び出される。



複数デバイス入力時は、デバイス毎にタブが表示。  
(関数 IA\_Eval\_EnableMultiPort の引数を2以上にすると、複数デバイス状態になる)



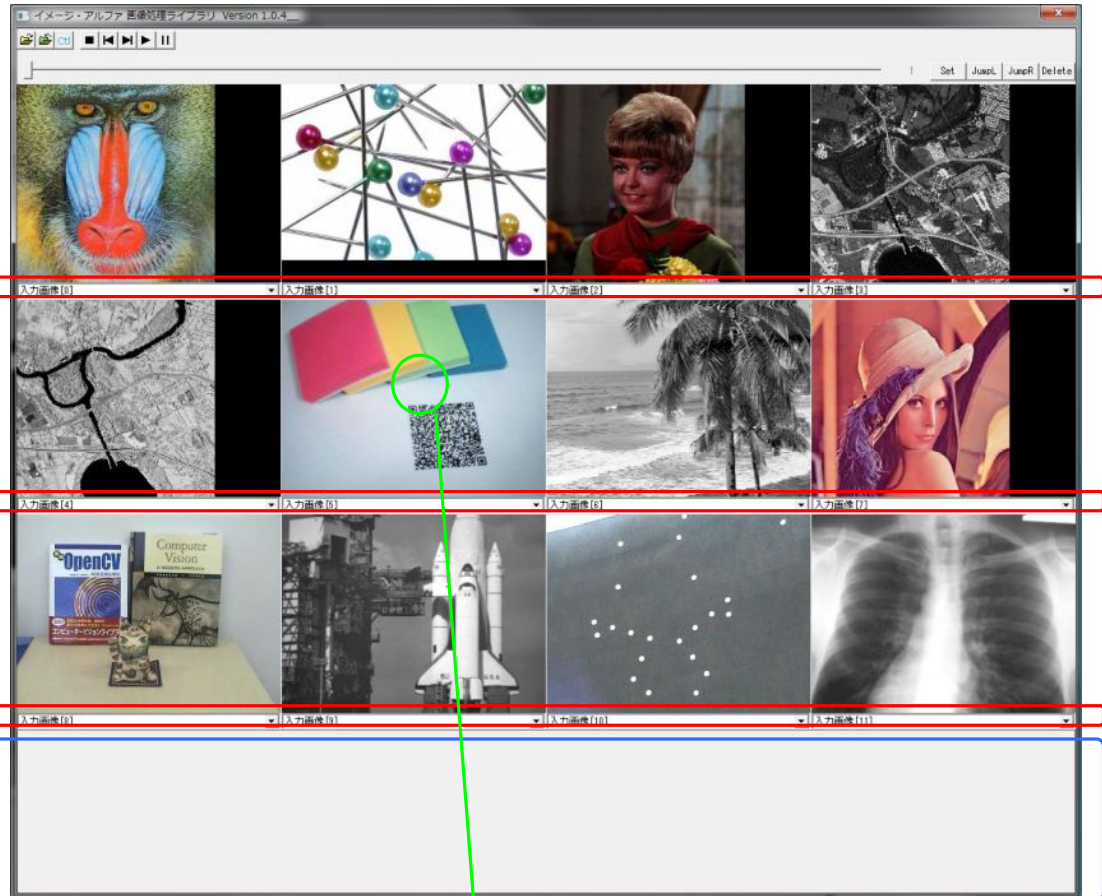
カメラも選択可能



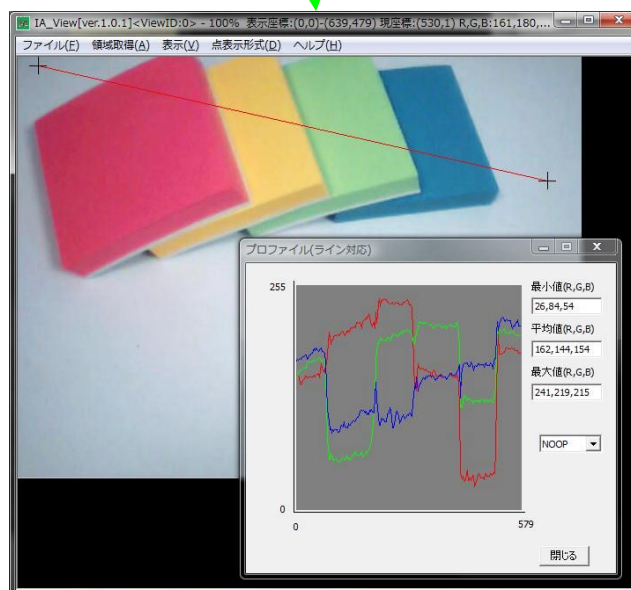
AVI、MPEG なら、処理したいファイルを選択。  
BMP、PPM、JPEG なら、処理したいファイルの先頭を選択。

## 任意画像表示

- ・メインウィンドウにて複数処理画像を表示。(最大 4x3 面表示)
- ・表示させたい画像は画像下にあるコンボボックスより選択。
- ・表示する画像枚数、濃度値も設定可能。



表示画像を右クリックすれば、IA\_View 起動。  
拡大表示、ヒストグラム表示も可能。



## OpenCV との連携

- IALIB の画面と同じフォーマットを持つ OpenCV の画面については、IALIB の処理対象とすることができます。  
例えば、画像処理は OpenCV、GUI(画像入力、制御パラメータ設定など)は IALIB、という組み合わせが可能になります。
- そのためには、OpenCV の画面を関数 IA\_AllocDummyImg により IALIB に登録する必要があります。
- 下記コードは、OpenCV の画面を確保し、IALIB に登録する関数です。  
(サンプルプログラムとして用意しましたので、そのままお使いいただけます)  
第一引数と第二引数で画面の種類を設定し、第三引数に確保された画面に関する情報が格納されます。

```

// 画面情報
typedef struct {
    IplImage *pCVImg; // OpenCV
    int IAID; // IALIB(画面ID)
} Img;

// OpenCVの画面を確保し、IALIBに登録する関数
int AllocCVImg( IA_Size *pImgSize, IA_ImgType imgType, char imgName[], Img *pImg )
{
    CvSize cvSize;
    int depth;
    int channels;

    // OpenCV画面確保
    cvSize.width = pImgSize->xSize;
    cvSize.height = pImgSize->ySize;
    if( imgType == IA_IMG_CHAR ){
        depth = IPL_DEPTH_8U;
        channels = 1;
    }else if( imgType == IA_IMG_RGB ){
        depth = IPL_DEPTH_8U;
        channels = 3;
    }else if( imgType == IA_IMG_SHORT ){
        depth = IPL_DEPTH_16S;
        channels = 1;
    }else if( imgType == IA_IMG_FLOAT ){
        depth = IPL_DEPTH_32F;
        channels = 1;
    }else{
        return -1; // 現在IALIB非対応
    }
    pImg->pCVImg = cvCreateImage( cvSize, depth, channels );

    // IALIBに登録
    pImg->IAID = IA_AllocDummyImg( pImg->pCVImg->imageData, pImgSize, imgType, pImg->pCVImg->widthStep, imgName );

    return 0;
}

```

項目	条件
画像サイズ	アライメント(align)が0となるような画像幅
ビットデプス、チャンネル数	以下のいずれかの組み合わせ ( [] は対応する IALIB 画面データタイプ ) <ul style="list-style-type: none"> <li>• IPL_DEPTH_8U、1 チャンネル [IA_IMG_CHAR]</li> <li>• IPL_DEPTH_8U、3 チャンネル [IA_IMG_RGB]</li> <li>• IPL_DEPTH_16S、1 チャンネル [IA_IMG_SHORT]</li> <li>• IPL_DEPTH_32F、1 チャンネル [IA_IMG_FLOAT]</li> </ul>

使用可能な OpenCV の画像、および対応する IALIB 画面データタイプ