

OpenVGR

座標系変換ツール説明書

クロスマーカール検出プログラムの使い方

V e r . 0.9.0

2011 年 12 月 21 日

独立行政法人 産業技術総合研究所

本文書の取り扱いについて

- 本書に掲載する情報は、使用者に有用なものであるように万全を期していますが、内容の正確性、最新性、その他一切の事項について保証をするものではありません。
- 本書の著者および著作権者は、使用者がこの文書から得たまたは得られなかった情報から生じる損害に対しても、一切責任を負いません。
- 本書は使用者への事前の予告なしに変更、削除、公開の中止を行うことがあります。

【連絡先】

(独) 産業技術総合研究所 知能システム研究部門 タスクビジョン研究グループ
〒305-8568 茨城県つくば市梅園 1-1-1 中央第二

E-Mail: openvgr-contact@m.aist.go.jp

1. プログラムの実行環境

本プログラムは以下の環境で動作します。プログラムの導入はこのファイルセットに同梱されている文書「はじめにお読みください」を参照して行ってください。

動作環境：

PC：CPU Core2Duo 以上，メモリ 1GB 以上，ハードディスク 10GB 以上

OS：Ubuntu 10.04 LTS Desktop 日本語 Remix（32bit 環境）

開発環境：

GNU Compiler Collection 4.4.3

OpenCV 2.0

2. クロスマーカー検出の目的

カメラキャリブレーションで得られた座標系と、カメラの観測結果を適用したい座標系が異なる場合、座標変換（剛体変換）行列を用いて目的の座標データやベクトルを得ます。ステレオカメラシステムで認識した物体をマニピュレータでつかむときはその一例です。ステレオカメラ座標系で得られた認識物体の位置姿勢をマニピュレータの座標系で表してマニピュレータへの動作を指令します。

この座標変換行列は、変換したい座標系（目的空間）であらかじめ場所がわかっている点の 3 次元位置データと、その点をカメラ座標系で計測した 3 次元位置データを組にして求め、この座標対応関係を 3 つ以上用いて計算します。

クロスマーカー検出プログラムは、正方形に対角線を引いて分けられた領域を白と黒の交互に塗り分けた、クロスパターンを目印（マーカー）として検出します。マーカーが空間に置かれた画像をプログラムに入力すると、マーカー中央の画素位置を出力します。画素位置データとステレオカメラの個々のカメラのキャリブレーションデータを用いて、マーカー中央点の空間内 3 次元座標を計算することができます。

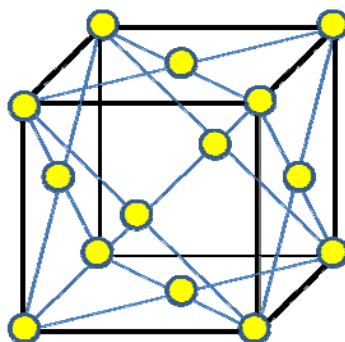


クロスマーカー

3. 計測対象画像の撮影

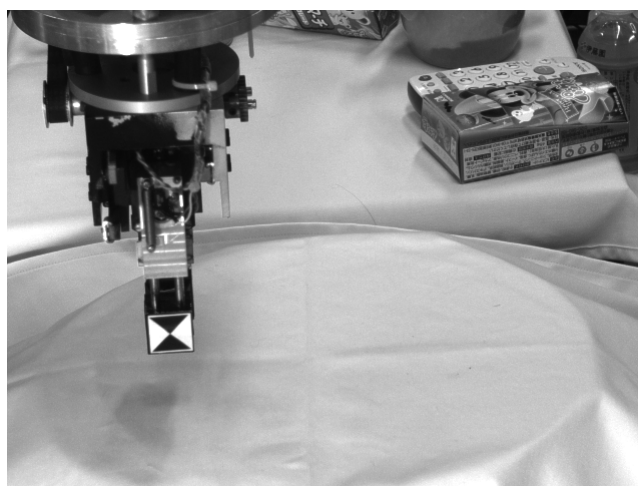
クロスマーカーをひとつだけ、目的空間内の既知の座標に置いてステレオカメラで撮影し、マーカーの位置を別の既知座標に変えて再び撮影した画像を 10 セットくらい用意します。画像は 24 ビット RGB か 8 ビットモノクロで作ります。マーカーを置く位置は、空間

内で必要な領域を直方体として決め、その頂点や頂点中間点にします。一例として、サイコロの 5 の目状にマーカーを配置する面で立方体を定めた図を下に示します。全ての画像中のマーカー位置をひとつにまとめたものです（全 14 点）。どの画像でもマーカーが完全に写っているように、定義する直方体の大きさを調整してください。



目的空間内のマーカー全配置例（黄色の丸がマーカー位置）

クロスマーカーは十分な大きさで、なるべく正対するように写す必要があります。ある程度斜めから写してもよいですが、検索対象として指定するテンプレート画像は矩形になるので、テンプレートとして切り出しても十分な大きさが確保できるようにします。例えば元画像の大きさが VGA の場合、テンプレートの大きさが 20×20 ピクセルよりも大きくないと安定して検出することができません。画像中での大きさを考慮してマーカーの作成をしてください。



マニピュレータを利用したマーカーの撮影例

画像が明るく、マーカーの白と黒のコントラストがはっきりしているほうが検出精度が上がりますが、マーカー表面が反射しない程度に明るさを抑えて撮影してください。

ステレオカメラによっては、各カメラの画像を連結した形で出力するものがあります。本プログラムは個々のカメラ画像を処理することを想定していますので、入力には連結を解除した画像を作成してください。

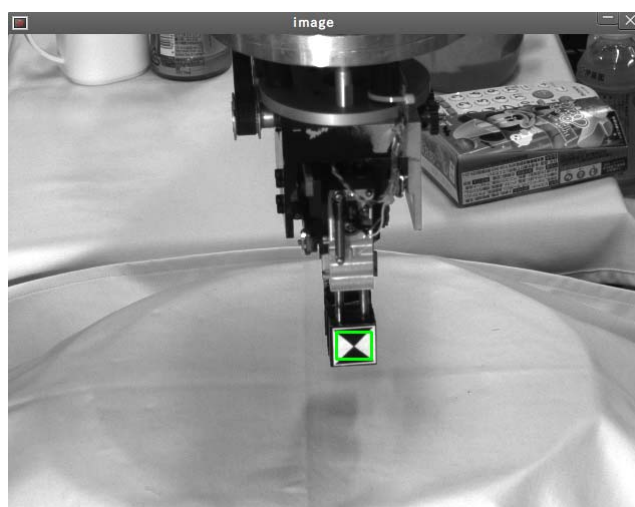
4. テンプレートの作成

テンプレートは入力画像内に写っている探索対象を指定するために使用します。計測のために用意した画像のひとつを選んで、そこからクロスマーカーが写っている部分を手動で切り出してテンプレートを作ります。まず、画像セットの中でマーカーが最も小さく写っているものを選んで元画像とします。次に **cutroi** プログラムを使ってテンプレート画像を切り出します。**cutroi** プログラムは以下のように起動します。

`./cutroi -i 元画像ファイル名 -o テンプレートファイル名`

プログラムを起動すると元画像が表示されます。この画像上でマウス左ボタンを押してドラッグすると画像上に緑色の矩形枠が現れます。これが選択した領域になります。ここで **c** ボタンを押すと選択領域がテンプレートファイルとして出力されます。領域を変更したいときは **r** ボタンを押すと、今まで選んでいた領域がキャンセルされ、新たに領域を選択することができます。プログラムを終了させる場合は **ESC** ボタンを押します。

テンプレートにするクロスマーカーの領域はなるべく大きくとれることが望ましいですが、マーカーの実際の境界はエッジのノイズが出やすいので、領域の内側を指定するほうがよいです。また、マーカーが正対していない場合もマーカーの内側を矩形でとるように指定します。



領域選択例



作成したテンプレート例

入力する元画像ファイルや出力するテンプレートファイルの画像フォーマットは、OpenCV の画像入出力関数が対応するものが使用できます。このフォーマットはプログラム実行時に与えるファイル名の拡張子によって指定するので、ファイル名には必ず拡張子をつけてください。ただし、元画像がモノクロであっても、テンプレートファイルは RGB カラーフォーマットで出力されます。従って、テンプレートファイルに .pgm のようなモノクロ画像としてのファイル名を与えても、内容は PPM ファイルとして出力されます。

5. マーカー検出の実行

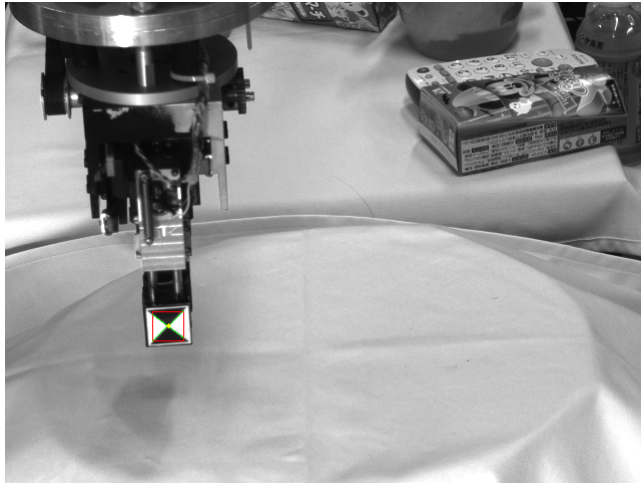
各画像でのクロスマーカー検出は **findx** プログラムを使って行います。**findx** プログラムは計測対象画像（被検索画像）とテンプレート画像を以下のように入力して実行します（実際は改行を含まず一行）。

```
.findx -i 被検索画像ファイル名 -t テンプレート画像ファイル名  
      [-o 出力画像ファイル名] [-disp]
```

入力や出力に使用される画像ファイルのフォーマットは前項の **cutroi** と同様に OpenCV の画像入出力関数が対応できるものです。フォーマットの指定に拡張子を用いるので、ファイル名に必ずつけてください。

プログラムを実行し検出ができると、結果としてマーカー中心の画素座標が標準出力に表示されます。画素座標は画像左上を原点としています。

出力画像ファイル名を指定すると、マーカー中央点が黄・マーカーの対角線が緑・テンプレート検出枠が赤でオーバーレイ表示された、検出結果画像を出力します。検出結果のマーカー中心座標は必ず正しいわけではないので、この画像を見て実行結果を確認する必要があります。なお、マーカー中央点のサブピクセル位置補正が行われている場合（デフォルト）は、緑色線交点と黄色の点の位置が一致しないことがあります。



出力画像例

-disp オプションは検出結果画像をプログラム実行時にポップアップ表示するために使
用します。画像表示している状態で何かキーを押すとプログラムが終了します。

また、**-h** オプションで以下のような簡単な使用法を標準エラーに出力します。

Usage : findx -i sourceImage -t templateImage [-o outputImage]

[-bthr #] [-sedg #] [-wedg #]

[-lmin #] [-gmax #] [-smin #] [-smax #] [-nosp]

[-disp] [-xout] [-eout] [-lout] [-sout] [-xelo] [-xels]

Default value

bthr:50 sedg:100 wedg:20 lmin:5 gmax:3 smin:25 smax:65

入力するファイル名だけでプログラムを実行すると、デフォルトの設定で処理を行いま
すが、検出を制御するパラメータとして以下のオプションがあります。

[-nosp] デフォルト=指定なし

このオプションを指定するとクロス線交点の画素座標をそのまま検出結果として出力しま
す。指定しない場合（デフォルト）は検出サブピクセル位置の補正が行われます。

[-bthr 2値化閾値] デフォルト=50

-xout オプションで出力した2値化画像または **-eout** オプションで出力したエッジ画像を
確認して、エッジがきれいに出ていないとき調整します。0 ～ 255 の値を指定します。
明るい画像のときは値を大きく、暗い画像のときは小さくします。

[-sedg 強いエッジの閾値] デフォルト=100

2 値化閾値の調整でエッジ検出が改善しない場合 0 ～ 255 の範囲で調整します。
弱いエッジの閾値より大きくなるようにしてください。

[-wedg 弱いエッジの閾値] デフォルト=20

2 値化閾値の調整でエッジ検出が改善しない場合 0 ～ 255 の範囲で調整します。
強いエッジの閾値より小さくなるようにしてください。

[-lmin 検出対角線の最短長 (ピクセル)] デフォルト=5

エッジ検出はよくできているが、**-lout** オプションで出力した画像で確認するとマーカーの対角線が検出できていないとき、この値を小さくすると結果が改善することがあります。また、対角線が間違って検出されているとき、値を大きくすると結果が改善することがあります。

[-gmax 検出対角線のとぎれ長さ最大値 (ピクセル)] デフォルト=3

エッジ検出はよくできているが、**-lout** オプションで出力した画像で確認するとマーカーの対角線が検出できていないとき、この値を大きくすると結果が改善することがあります。また、対角線が間違って検出されているとき、値を小さくすると結果が改善することがあります。

[-smin 検出対角線の傾き絶対値最小 (deg)] デフォルト=25

マーカーの対角線が検出できないとき、**-sout** オプションで検出直線の傾きを確認し、その結果により調整します。傾きは左上原点の画素座標で計算したものです。

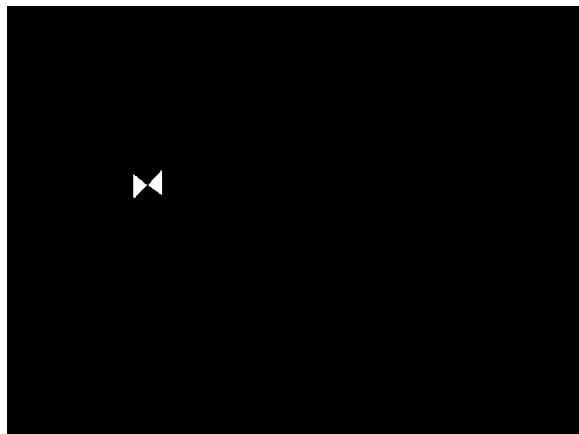
[-smax 検出対角線の傾き絶対値最大 (deg)] デフォルト=65

マーカーの対角線が検出できないとき、**-sout** オプションで検出直線の傾きを確認し、その結果により調整します。傾きは左上原点の画素座標で計算したものです。

検出パラメータ調整を助けるために以下のオプションがあります。

[-xout]

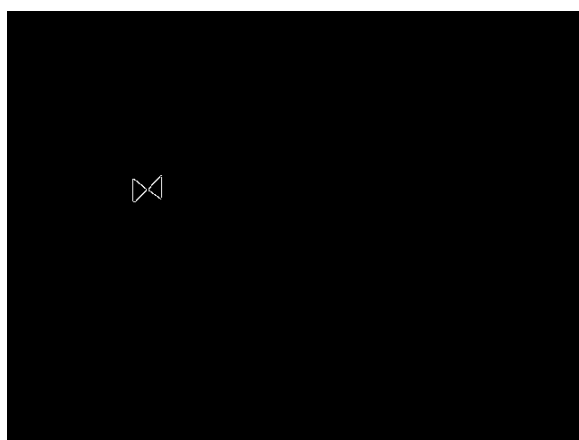
テンプレートが適合した場所の2値化画像を **bmp** ファイルで出力します。このファイル名は **-o** オプションで指定した出力ファイルのベース名に **-extr.bmp** が追加されたものです。出力ファイル名を指定していない場合は **-i** オプションで指定した入力ファイルのベース名を使用します。



-xout オプションの出力画像例

[-eout]

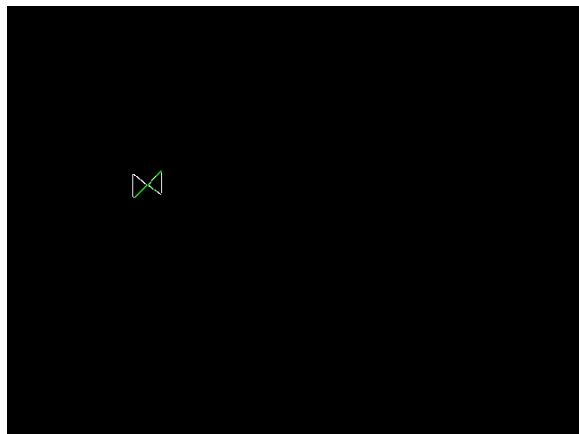
テンプレートが適合した場所のエッジ画像を **bmp** ファイルで出力します。このファイル名は **-o** オプションで指定した出力ファイルのベース名に **-edge.bmp** が追加されたものです。出力ファイル名を指定していない場合は **-i** オプションで指定した入力ファイルのベース名を使用します。



-eout オプションの出力画像例

`[-lout]`

テンプレートが適合した場所の対角線検出結果画像を **bmp** ファイルで出力します。検出した線分はエッジ画像上に緑色で示されます。出力 **bmp** ファイル名は `-o` オプションで指定した出力ファイルのベース名に `-line.bmp` が追加されたものです。出力ファイル名を指定していない場合は `-i` オプションで指定した入力ファイルのベース名を使用します。



`-lout` オプションの出力画像例

`[-sout]`

テンプレートが適合した場所で検出した直線（線分）すべての傾きを標準エラー出力に表示します(単位 **deg**)。傾きは左上原点の画素座標で計算したものです。

`[-xelo]`

`-xout,-eout,-lout` を同時に指定した場合と同じです。

`[-xels]`

`-xout,-eout,-lout,-sout` を同時に指定した場合と同じです。