画像処理ライブラリ

WIL-Builder チュートリアル
(Ver3.0.0 以降対応)

☆第 3 版☆

(株)ファースト

(株)ファーストは、同社が提供していない装置における同社製ソフトウェア・ハードウェアの使用または信頼性についてはいかなる責任も負いません。(株)ファーストは本書で記載されているソフトウェア・ハードウェアの内容、商品価値、又は特定の使用目的に対する責任に対して明示又は黙示に関わらずいかなる保証も行いません。

本書の内容は、予告なしに変更することがあります。内容の変更について、(株)ファーストはいかなる責任も負いません。本書あるいは関連ソフトウェアにおける誤りから生じる損害について、(株)ファーストはいかなる責任も負いません。

御注意

目　次

目　次

1. はじめにお読みください

この度は、FAST Vision Library シリーズ WIL をご購入頂きまして誠にありがとうございます。

1.1 本ドキュメントについて

本書では「WIL-Builder 操作説明書」により、基本操作を理解している方を前提に、WIL-Builder で画像処理を構築する方法を解説しています。

1.2 関連ドキュメントについて

WIL-Builder で画像処理を構築していく場合、Function の機能説明は、以下の関連ドキュメントをご参照ください。

① FVCL Reference(ヘルプ)
WILのライブラリリファレンス(FVCL版)とプログラマーズガイドを統合したものです。WILセットアップ時にインストールされるヘルプファイルにてご提供いたします。

② FVIL Reference(ヘルプ)
WILのライブラリリファレンス(FVIL版)とプログラマーズガイドを統合したものです。
WILセットアップ時にインストールされるヘルプファイルにてご提供いたします。

③ FIE説明書(ヘルプ)
WILのFIEライブラリのリファレンスです。
WILセットアップ時にインストールされるヘルプファイルにてご提供いたします。

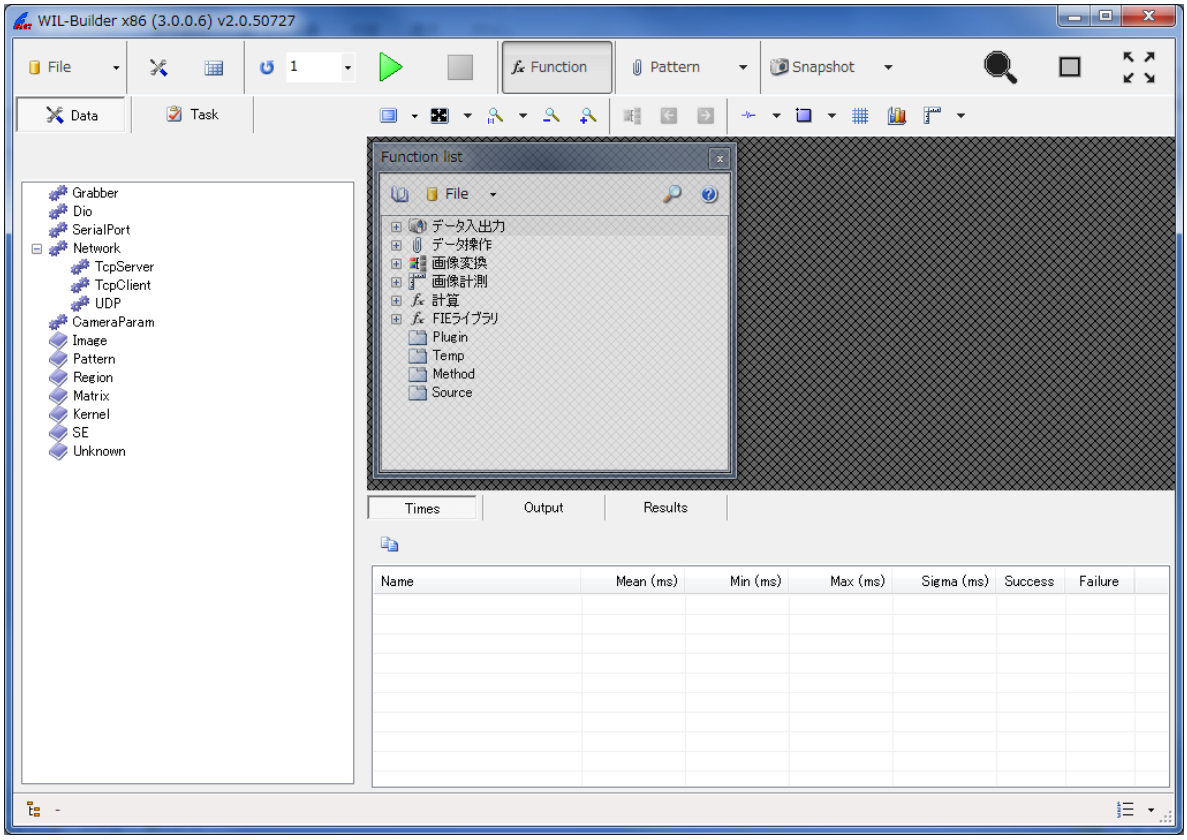
④ 各種ボードの取扱説明書
ファースト製各種ボードのハードウェアに関する情報が記載されています。ファースト製各種ボードをご購入いただきましたお客様を対象とした取扱説明書です。

⑤ 画像処理解説書
WIL に搭載されている各画像処理手法の解説書です。アルゴリズムやパラメータについて解説しています。

2. 基本操作編

WIL-Builder の基本的な操作の流れを説明します。

起動時の画面は以下のようになります。



2.1 画像を入力してみよう

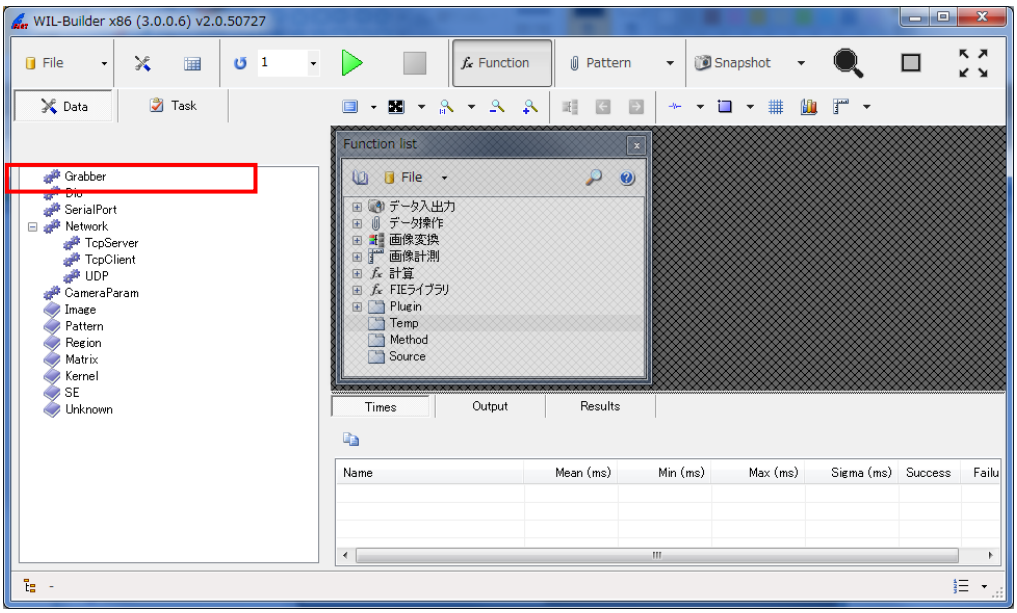
カメラからの画像入力と、画像ファイルを読み込んでの画像入力の手順を説明します。

2.1.1 カメラからの画像入力

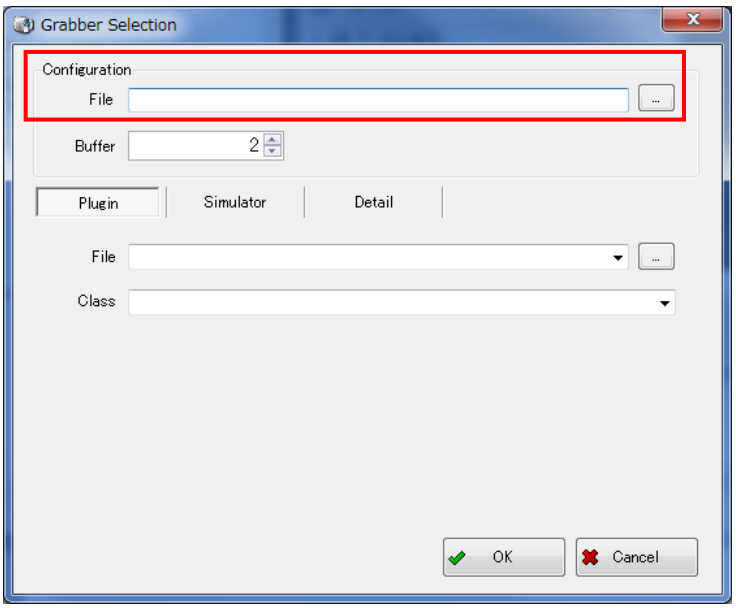
WIL では、Grabber Board 毎に対応しているカメラ用の「カメラ設定ファイル(ini ファイル)」が用意されています。

カメラから画像入力を行う場合、この ini ファイルを選択する必要があります。

① Data タブの「Grabber」を右クリックし、Open を選択します。

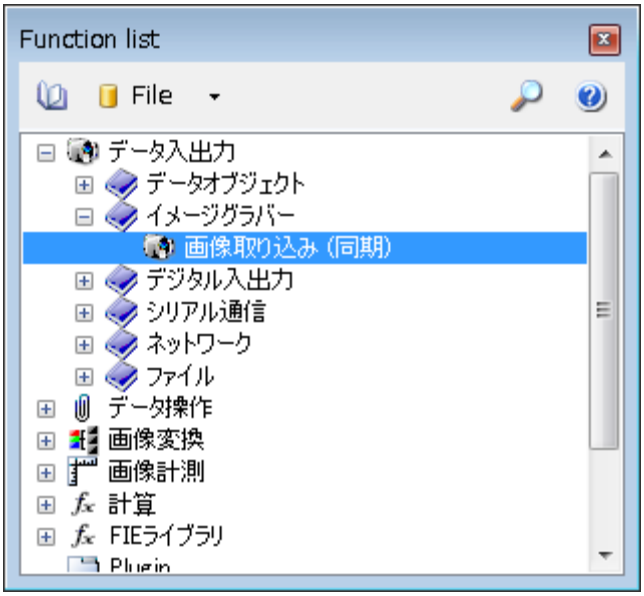


② Grabber Selection ダイアログが表示されますので、Configuration の File から使用するカメラ用の ini ファイルを選択します。選択後「OK」をクリックして、ダイアログを閉じます。

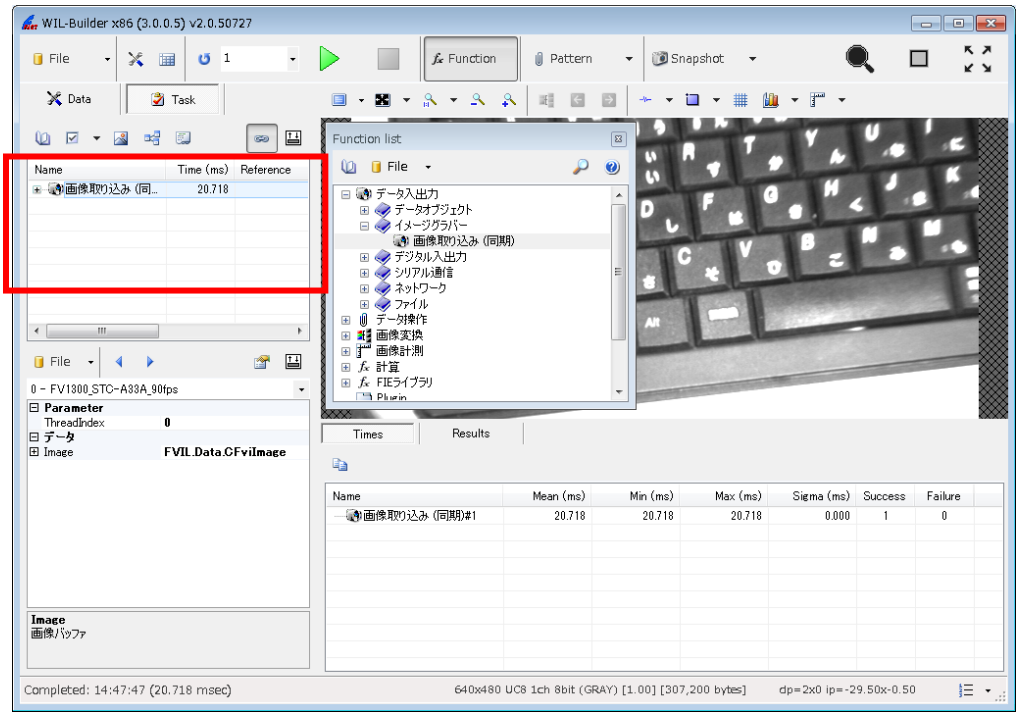


③ ツールバーの「Function」をクリックし、Function list をアクティブにします。

④ Function list の「データ入出力」、「イメージグラバー」の順にリストを展開し「画像取り込み」をダブルクリックします。



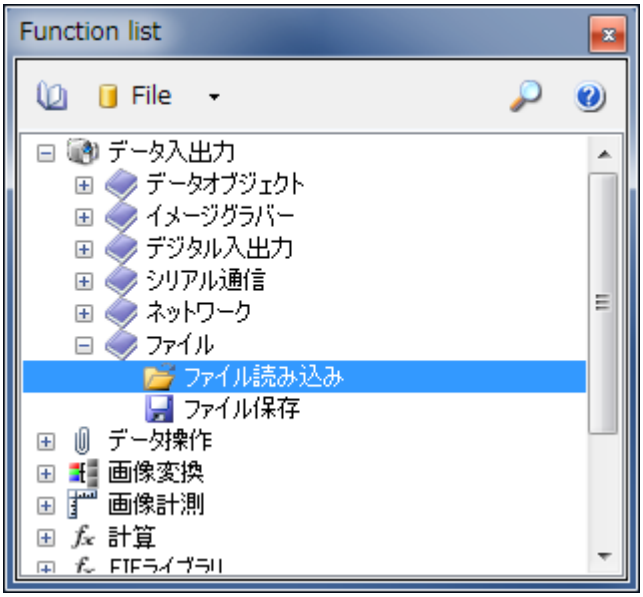
⑤ TASK タブのワークフローリストに「画像取り込み」が追加され、画像ビューにカメラから取り込んだ画像が表示されます。



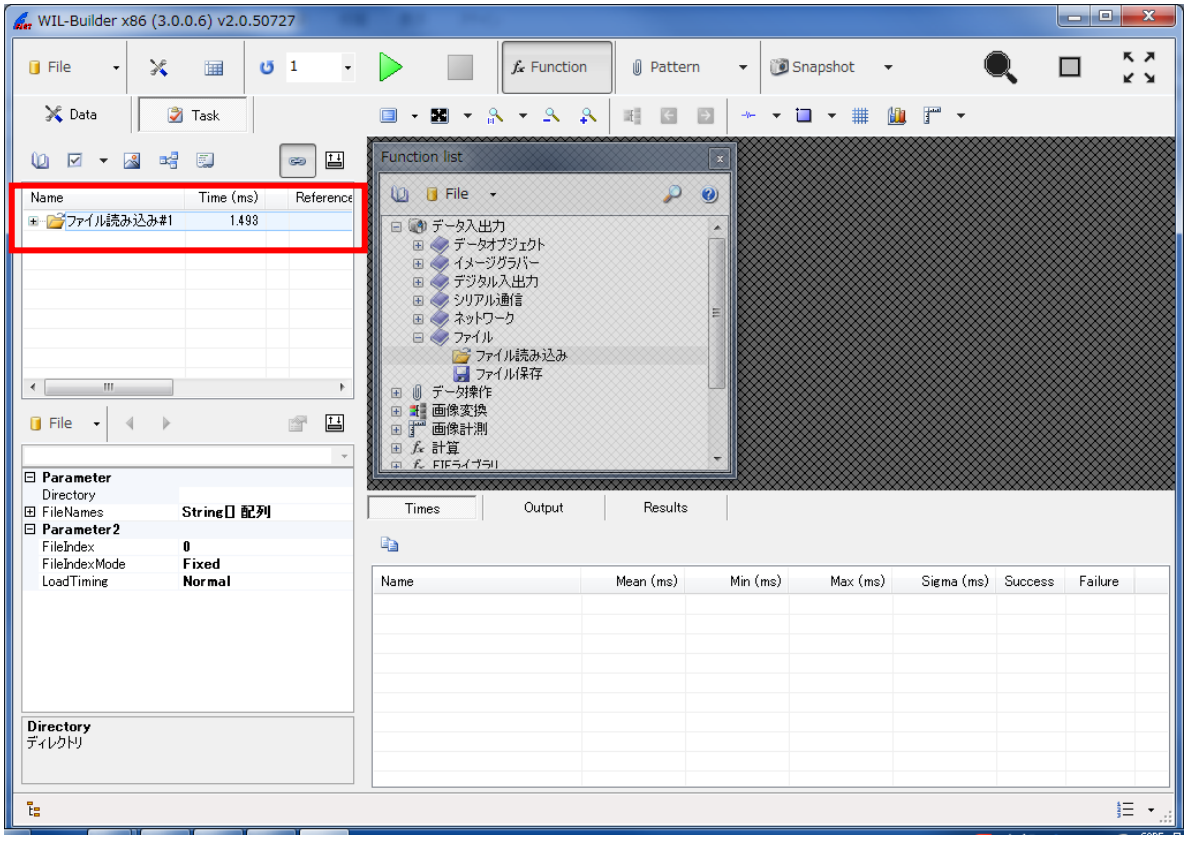
2.1.2 ファイルからの画像入力

カメラからの画像入力ではなく、画像ファイルを読み込んで、画像処理を行う事も可能です。
手順を説明します。

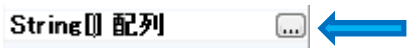
① Function list の「データ入出力」の＋マークをクリックします。
リストが展開されますので、同様に「ファイル」の＋マークをクリックします。
さらにリストが展開されますので「ファイル読み込み」をダブルクリックします。



② TASK タブのワークフローリストに「ファイル読み込み」が追加されます。

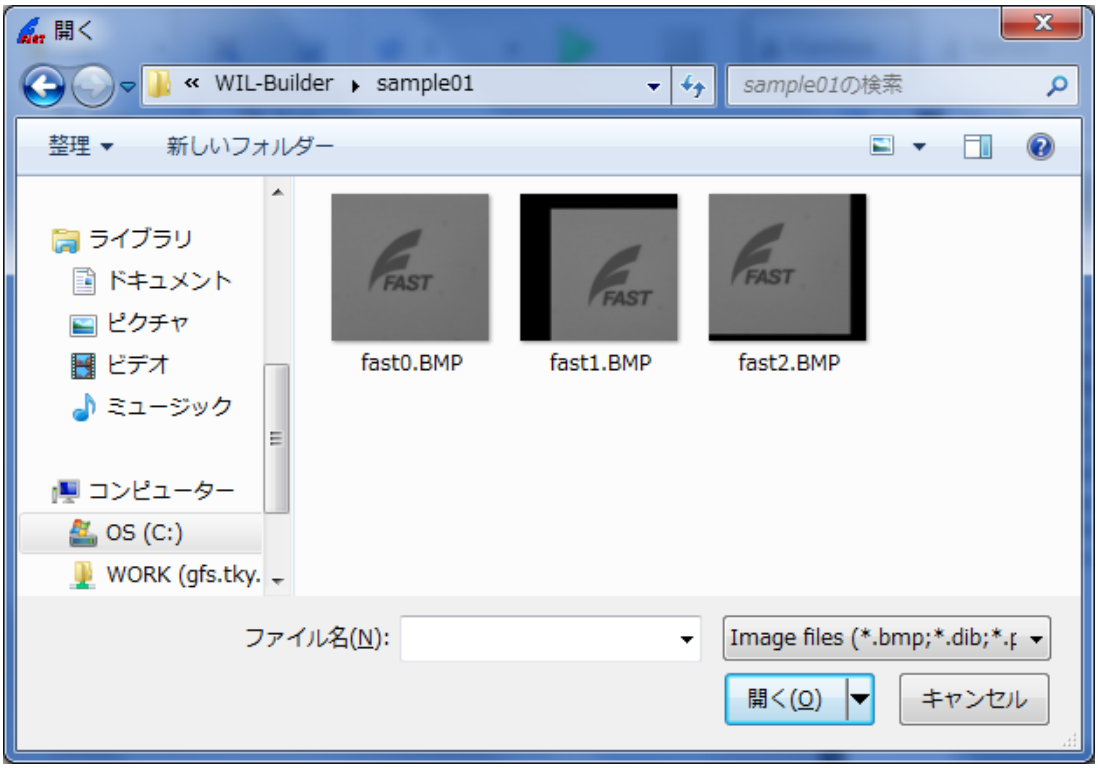


③ TASK タブの下側にプロパティグリッドが表示されていますので、「FileNames」をクリックし、画像ファイルを指定します。

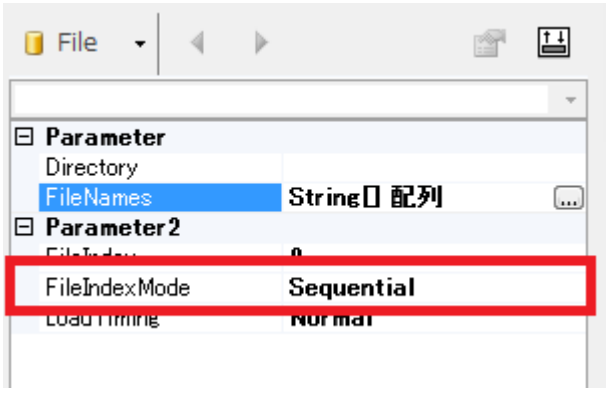


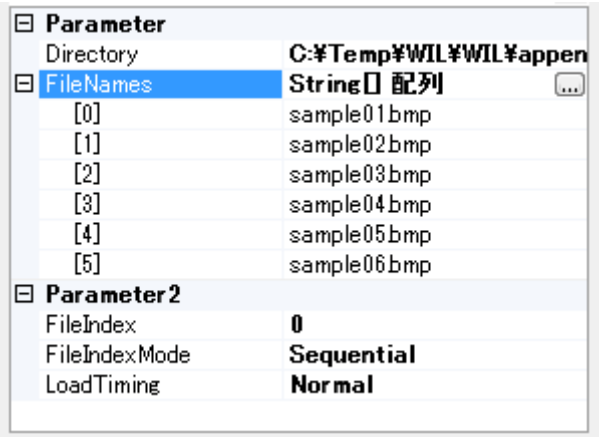
ここをクリックすると画像ファイル名を選択できます。

インストール時に指定したサンプルのフォルダを指定します。デフォルトのフォルダは「デスクトップ¥WIL¥appendix¥Samples¥WIL-Builder」となっています。対象ファイルフォーマットは、BMP、JPEG、PNG、Raw、TIFF となっています。



④ ③で指定した画像ファイルが、画像ビューに表示されます。



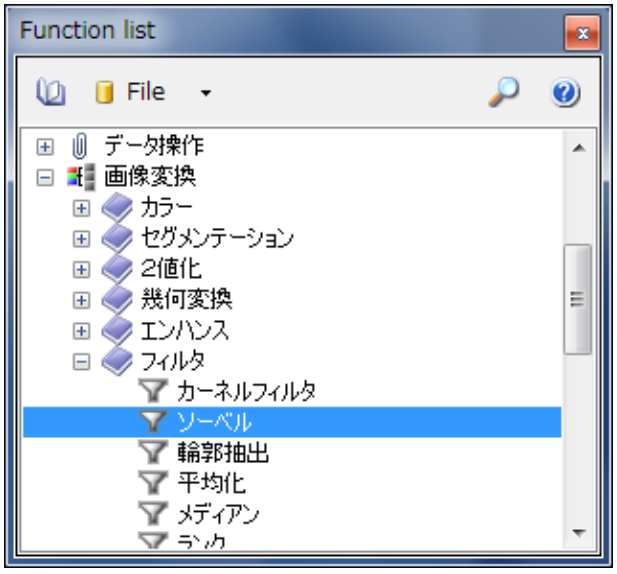


複数の画像ファイルを設定することも可能です。「ファイル読み込み」をクリックし、TASK タブの下側にプロパティグリッドが表示されていますので、Parameter2 の FileIndexMode を Fixed から Sequential に変更し、上記のファイル選択時に Ctrl キーを押しながら、ファイル名を複数選択することができます。

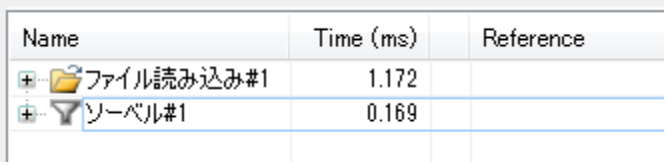
2.2 簡単な画像処理をしてみよう

画像が入力できるようになったので、簡単な画像処理をしてみましょう。

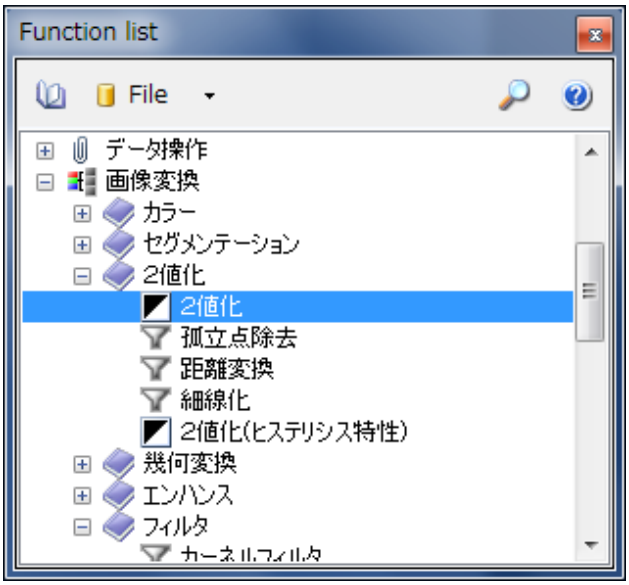
① 画像が入力できるようになった TASK に、画像処理を追加します。
② Function list の「画像変換」、「フィルタ」の順にリストを展開し、「ソーベル」をダブルクリックします。



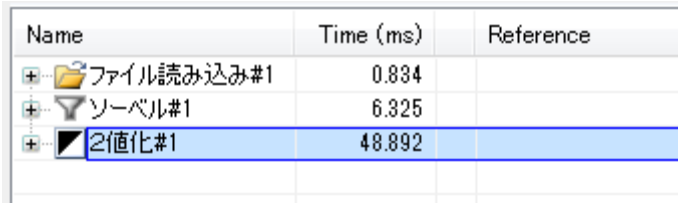
③ TASK タブのワークフローリストに「ソーベル」が追加され、画像ビューに変換された画像が表示されます。



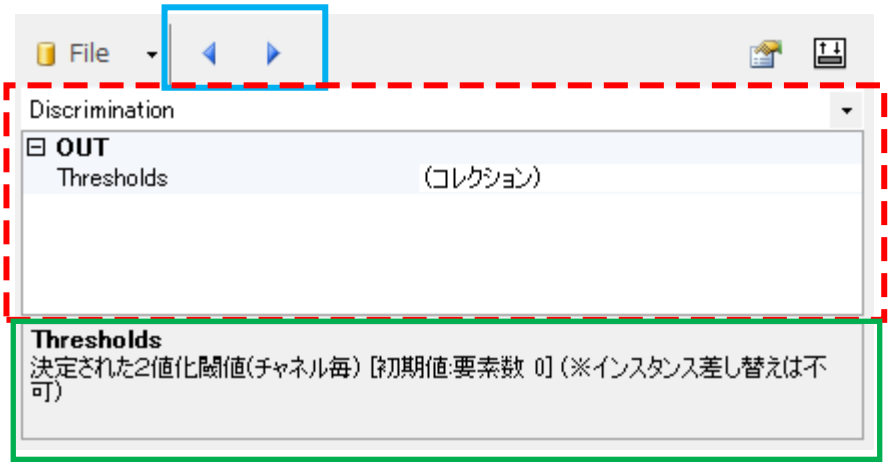
④ 変換された画像を 2 値化します。Function list の「画像変換」、「2 値化」の順にリストを展開し、「2 値化」をダブルクリックします。



⑤ TASK タブのワークフローリストに「2 値化」が追加され、画像ビューに、2 値化された画像が表示されます。

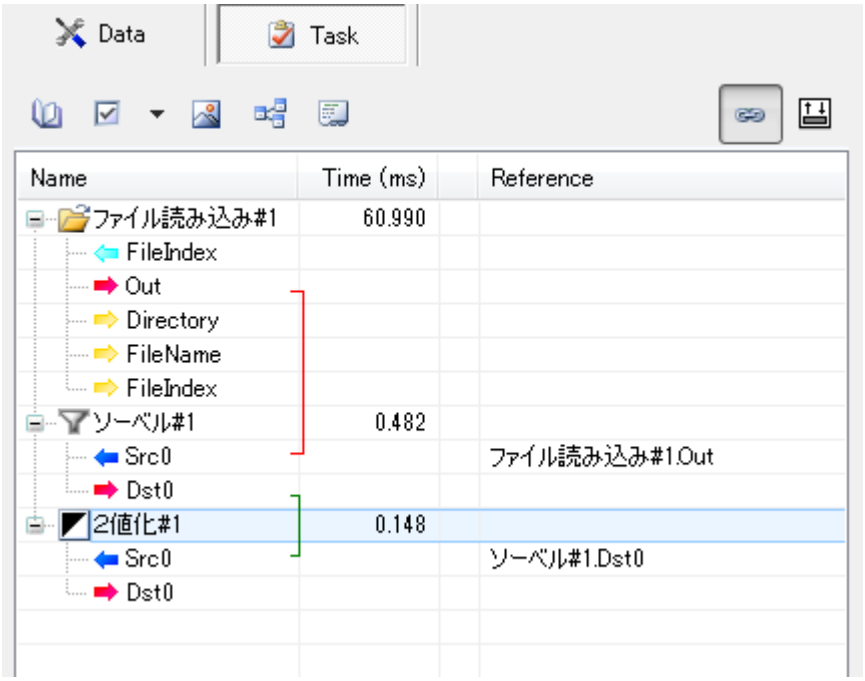


複数の機能を持つノードを選択すると青いカーソル(下図青枠)とリストボックス(下図橙枠)が有効になります。 これらで機能の指標を切り替えることができます。 青いカーソルは機能の指標をシーケンスに増減します。(左 : 1 つ減少、右 : 1 つ増加) リストボックスは機能の指標と識別名の一覧を表示しています。



※対象ブローブが 2 値化できる最適なノードを選択してください。
ノードの切り替えは、上図緑枠を確認してください。現在設定されているノードが表示されます。

※画像データの受け渡しは、自動的に行っていますが、任意に設定することも可能です。
出力ファイル(Out or Dst)を入力ファイル(Src)へドラッグアンドドロップすることにより、受け渡しを行う事ができます。
受け渡しの修正方法は、ドラッグをやり直すか、右クリックし、「Reset」を選択してください。

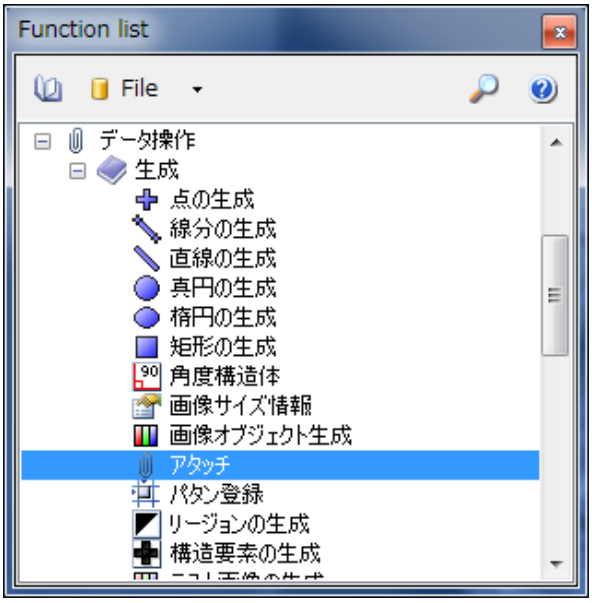


2.3 処理範囲を設定してみよう

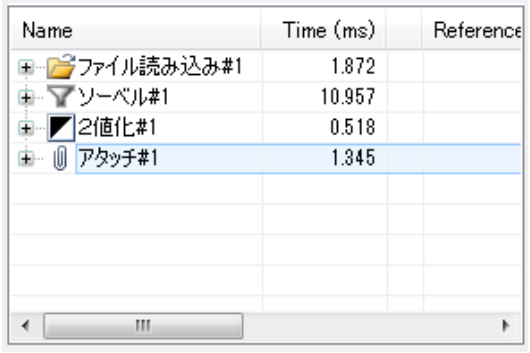
処理範囲を設定してみましょう。処理範囲の設定は、画像メモリを管理するクラス（FVIL.Data.CFviImage）で行います。

① 画像処理が行えるようになった TASK に、処理範囲を追加します。

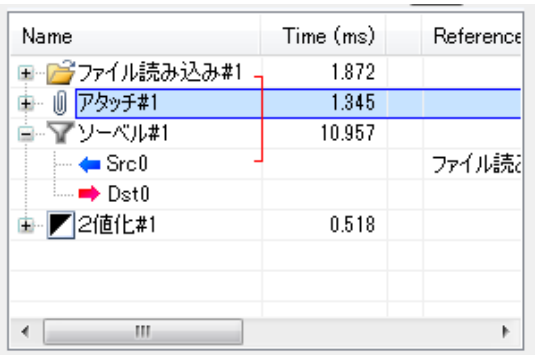
② Function list の「データ操作」、「生成」の順にリストを展開し、「アタッチ」をダブルクリックします。



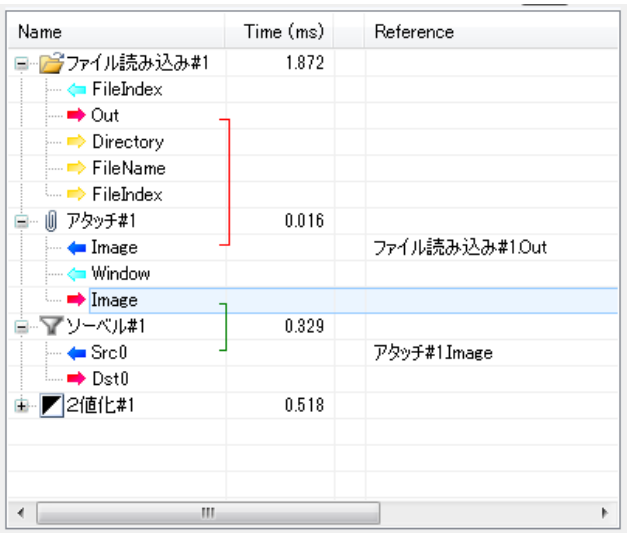
③ TASK タブのワークフローリストの最後の行に「アタッチ」が追加されます。



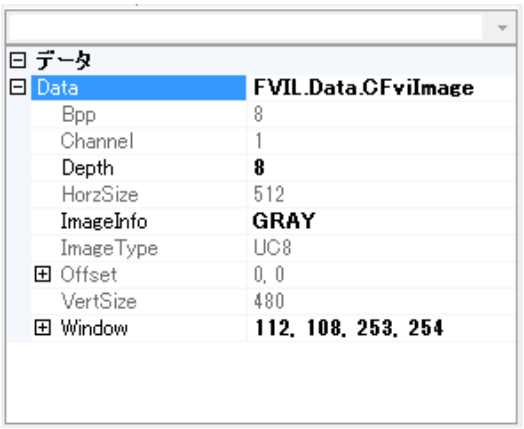
④ 「アタッチ」を、ドラッグして「ファイル読み込み」の後ろに移動します。



⑤ アタッチを移動した為、画像データの受け渡しが、崩れたので、ドラッグアンドドロップで修正します。



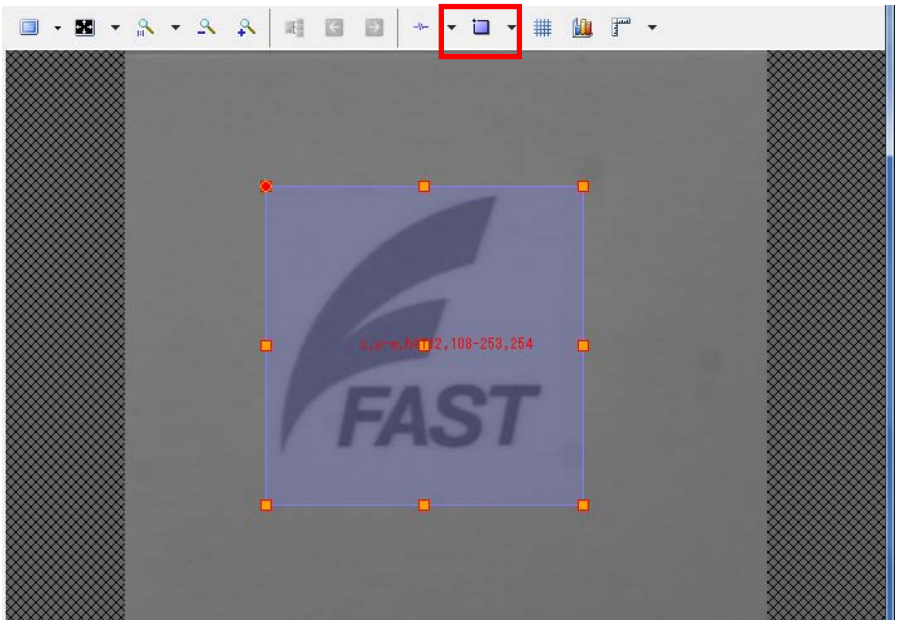
⑥ 「アタッチ」をクリックすると、TASK タブの下側にプロパティグリッドが表示されていますので、「Data」をクリックすると Parameter が展開されます。一番下の「Window」が処理範囲となっています。



X 始点、Y 始点、X サイズ、Y サイズの順です。

⑦ 「Window」の値を変更すると、画像処理を行う範囲が変更されます。

※TASK タブのワークフローリスト内の「アタッチ」をクリックした状態で、ツールバーの WOI メニューをクリックすると処理範囲をマウスで設定することもできます。設定終了後、再度 WOI メニューをクリックするとマウス設定モードが解除されます。



3. 実践編

基本的な操作が理解できたので、実践的な操作をしてみましょう。

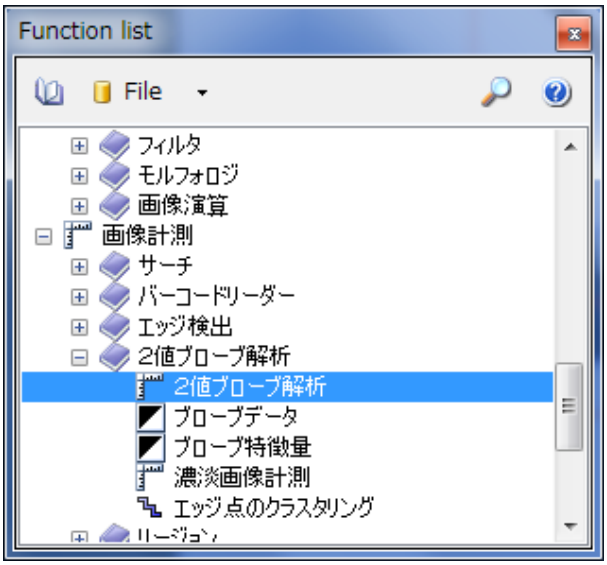
3.1 2値ブローブ解析を設定してみよう

2値化した画像に対しブローブ解析を設定します。
なお、画像入力や2値化、処理範囲の設定は基本操作編で説明したので省略します。

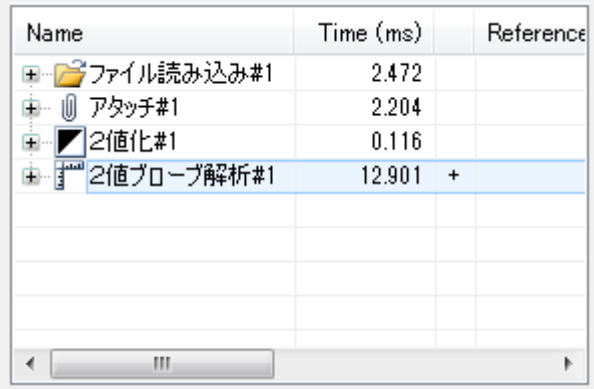
3.1.1 2値ブローブ解析

2値化した画像から、選択した色のブローブを解析する機能(FVIL.Blob)を設定します。

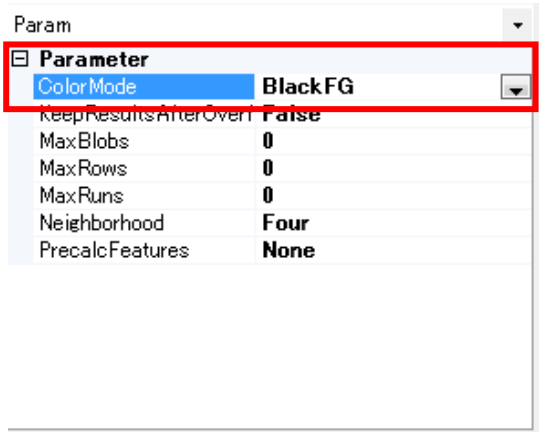
① Function list の「画像計測」、「2値ブローブ解析」の順にリストを展開し、「2値ブローブ解析」をダブルクリックします。



② TASK タブのワークフローリストの最後の行に「2値ブローブ解析」が追加されます。



③ 「2 値ブローブ解析」をクリックすると、TASK タブの下側にプロパティグリッドが表示されていますので、「ColorMode」をクリックし、解析したい色を選択できます。



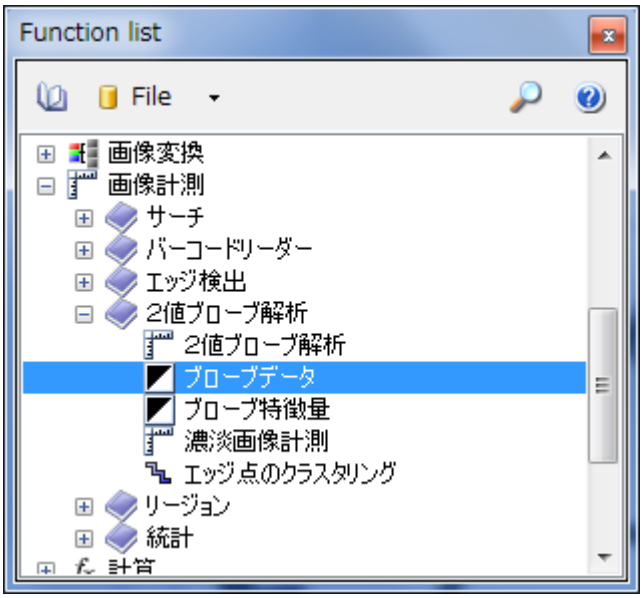
④ 画像ビューに選択された色のブローブが解析され、表示されます。

※この画像は SAMPLE01 フォルダの FAST0.BMP の画像に対し、Threshold を 80 に設定し、Black を対象に解析したものです。

3.1.2 結果の表示等

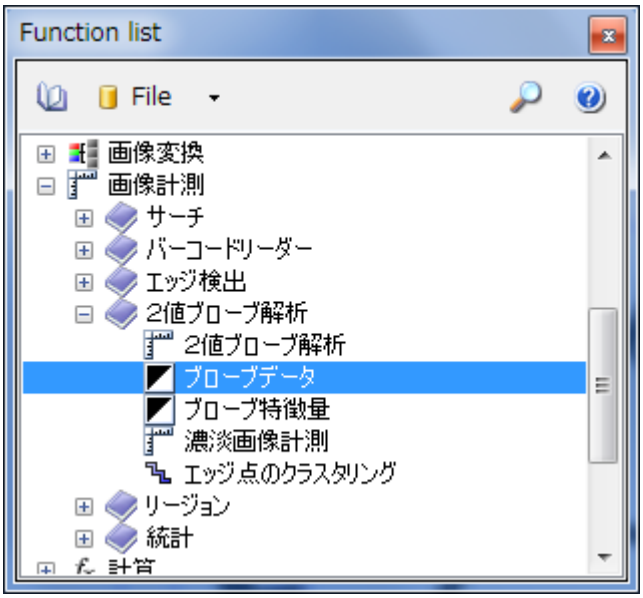
2 値ブローブ解析で得られた結果を使用して、2 点間の距離を計測します。

① Function list の「画像計測」、「2 値ブローブ解析」の順にリストを展開し、「ブローブデータ」をダブルクリックします。

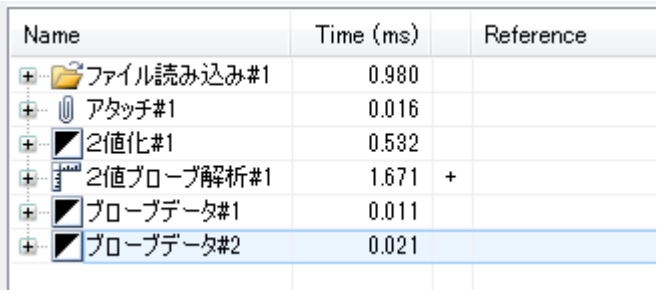


② TASK タブのワークフローリストの最後の行に「ブローブデータ#1」が追加されます。

③ さらにもう一つ「ブローブデータ」を追加します。



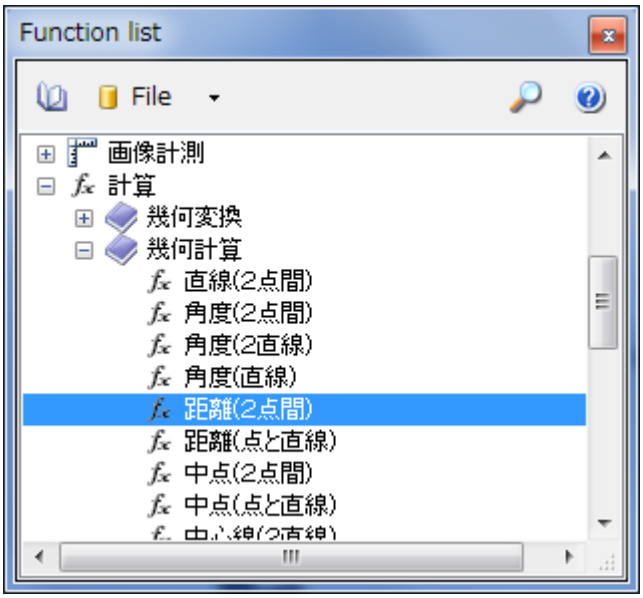
④ TASK タブのワークフローリストの最後の行に「ブローブデータ#2」が追加されます。



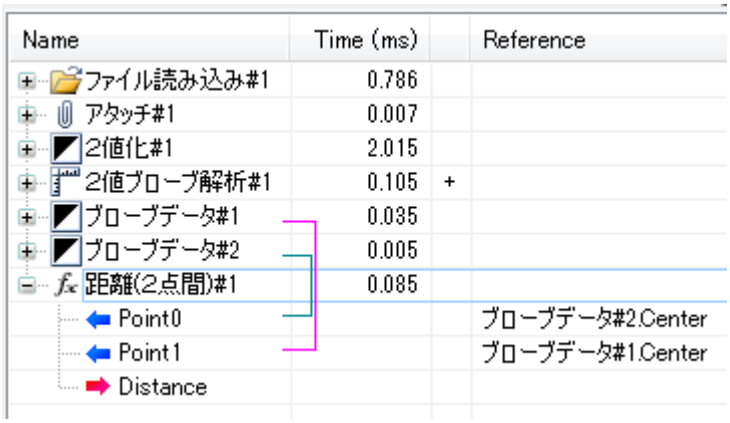
⑤ 「ブローブデータ#2」をクリックし、TASK タブの下側にプロパティグリッドが表示されていますので、Parameter の「Index」を「1」に変更します。

※画面内に複数個のブローブがあることを前提に、設定しています。

⑥ Function list の「計算」、「幾何計算」の順にリストを展開し、「距離(2 点間)」をダブルクリックします。



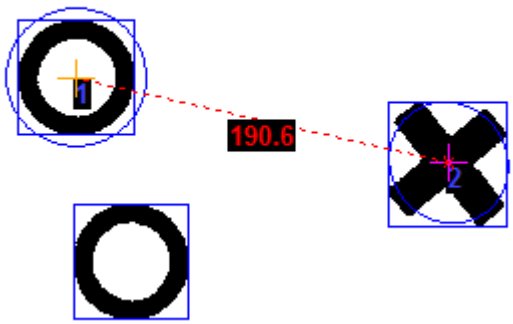
⑦ TASK タブのワークフローリストの最後の行に「距離(2 点間)」が追加されます。



(FVIL.Caliper.Function.Distance)

⑧ 実行アイコンをクリックしてみましょう

画像ビューに 2 点間を計測した値が、画素で表示されます。

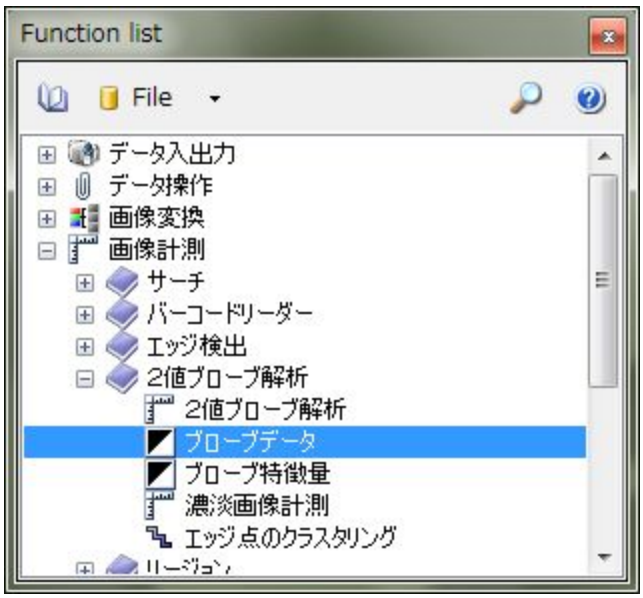


3.1.3 2 値ブローブ解析結果を利用してみよう

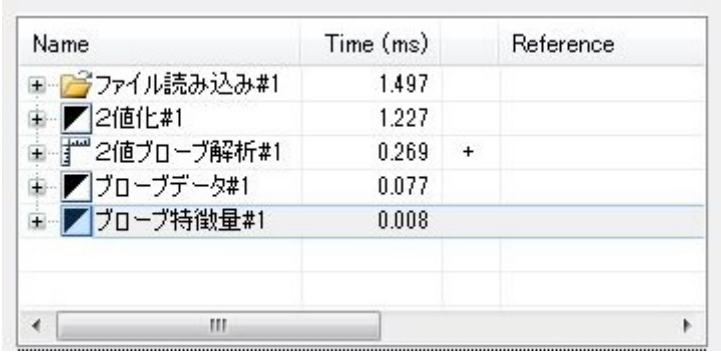
2 値ブローブ解析で得られたブローブの特徴量を取り出し、画像処理に利用できます。今回は周囲座標(Boundary)を利用して、ブローブの頂点の抽出を行います。
なお、ファイル読み込みや 2 値化、2 値ブローブ解析は前項で説明していますので省略します。

3.1.4 ブローブ特徴量の取得

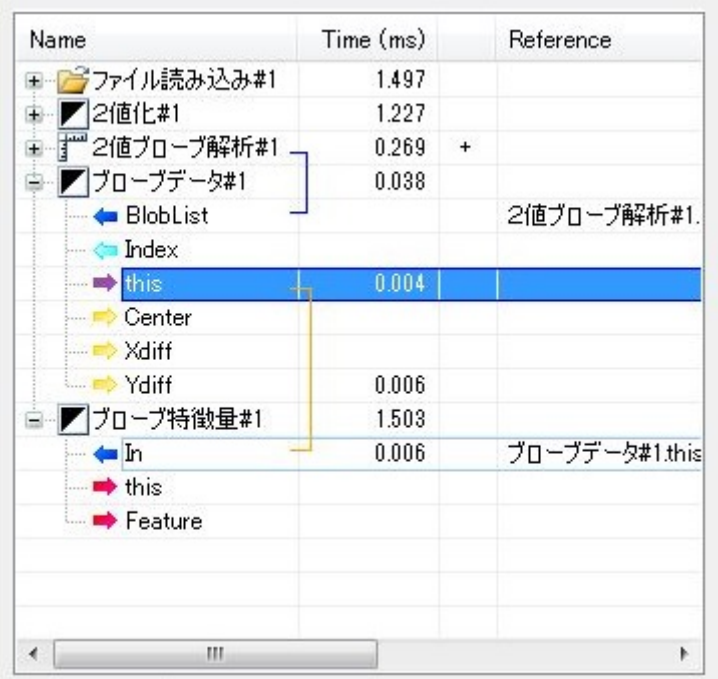
① 2 値ブローブ解析の結果から、ブローブ特徴量を取得します。
Function list の「画像計測」、「2 値ブローブ解析」の順にリストを展開し、「ブローブデータ」と「ブローブ特徴量」をダブルクリックします。



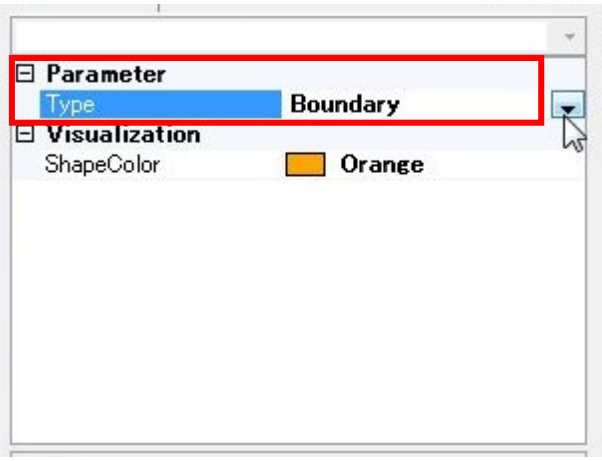
② TASK タブのワークフローリストの最後の行に「ブローブデータ#1」および「ブローブ特徴量#1」が追加されます。



③ 「ブローブデータ#1」の this を「ブローブ特徴量#1」の In とリンクします。

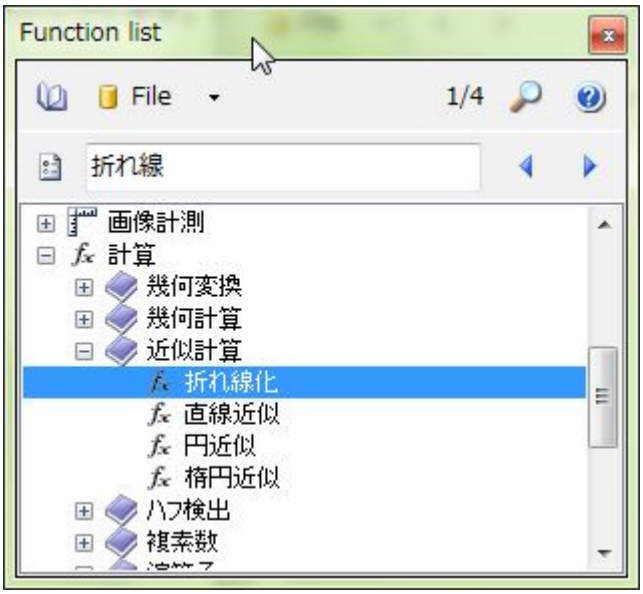


④ 「ブローブ特徴量#1」をクリックすると、TASK タブの下側にプロパティグリッドが表示されていますので、「Type」をクリックし、Boundary を選択します。

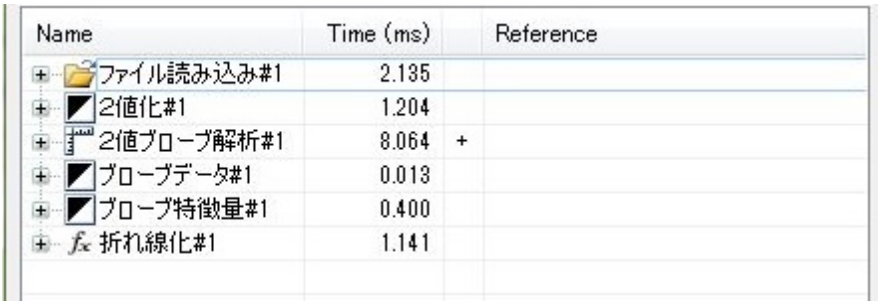


3.1.5折れ線化

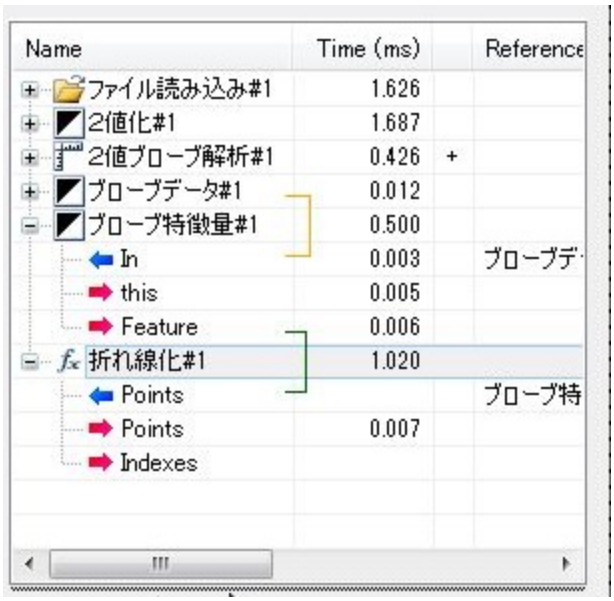
① Function list に戻り「計算」、「近似計算」の順にリストを展開し、「折れ線化」をダブルクリックします。



② TASK タブのワークフローリストの最後の行に「折れ線化#1」が追加されます。



③「ブローブ特徴量#1」の Feature を「折れ線化#1」の Points とリンクします。



④ 「折れ線化#1」をクリックすると、折れ線化の実行結果が表示されます。

3.2 グレイサーチを設定してみよう

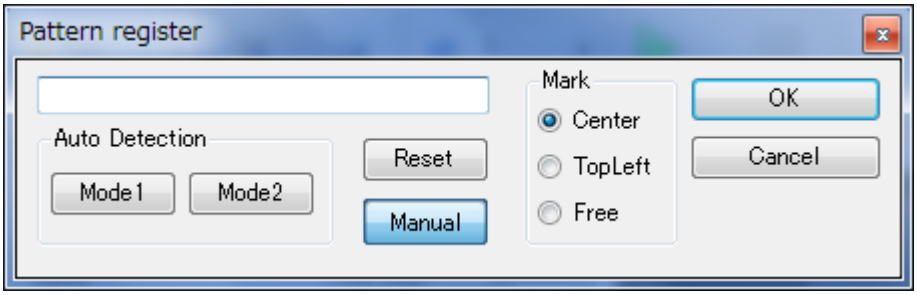
濃淡画像でパタンを登録し、そのパタンを処理範囲の中からサーチする機能を設定します。
なお、画像入力や処理範囲の設定は基本操作編で説明しましたので省略します。

3.2.1 パタン登録

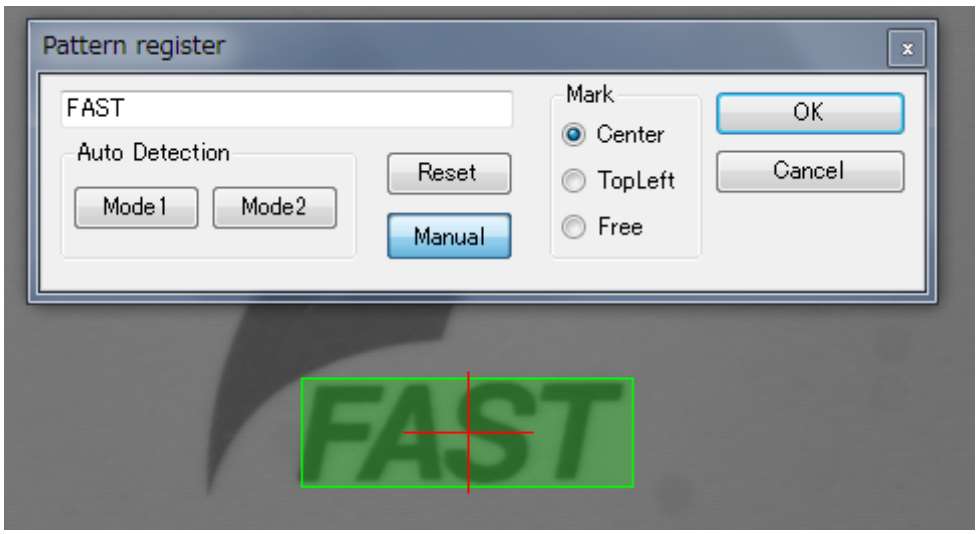
パタンの登録は、ワークフロー内ではなく、メイン画面のツールバーにある「Pattern」をクリックすることで登録することができます。
※登録方法の詳細「WIL-Builder 説明書」参照してください。

① メイン画面のツールバーにある「Pattern」をクリックします。

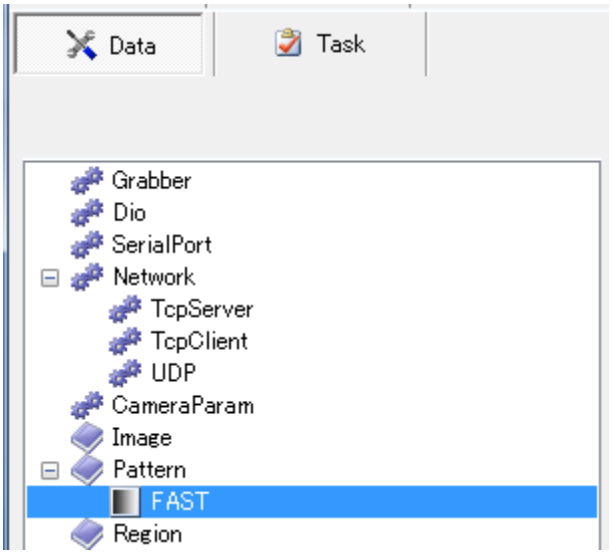
② ツールボックスが表示されます



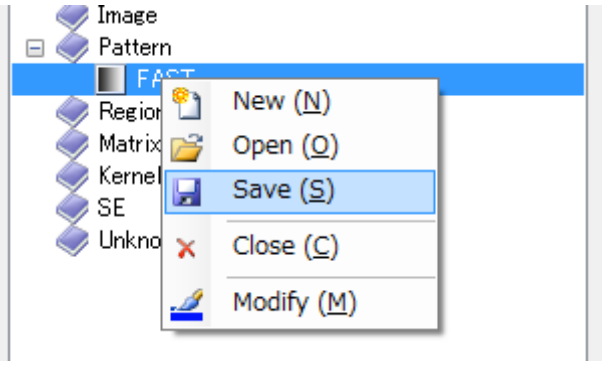
③ 登録したい領域をマウスで設定し、パタン名称を記入し登録します。



④ 登録されたパタンは、DATA タブの「Pattern」に表示されます。



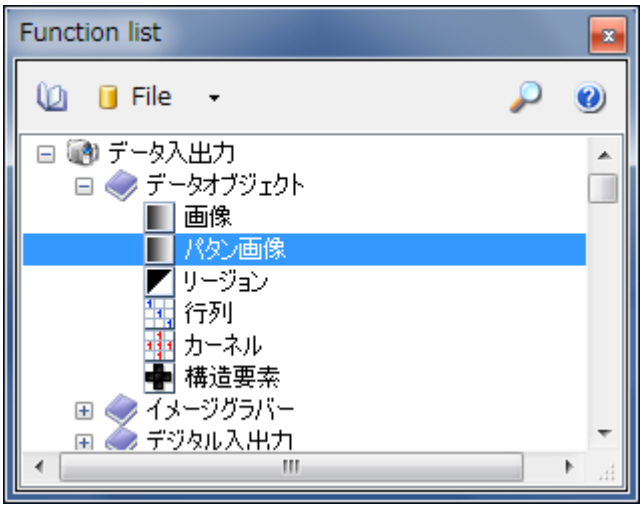
⑤ 登録したパタンを保存する場合は、パタン名称を右クリックし、Save を選択します。



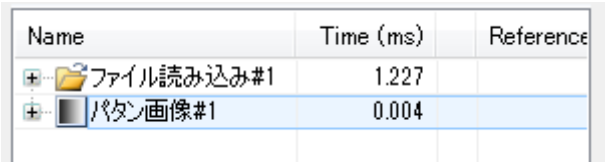
3.2.2 パタンの設定

サーチを行う為のワークフローを設定します。
メイン画面のツールバーにある「Pattern」をクリックして、パタンは登録したので、それを使用するための手続きを設定します。
なお、画像入力や処理範囲の設定は基本操作編で説明しましたので省略します。

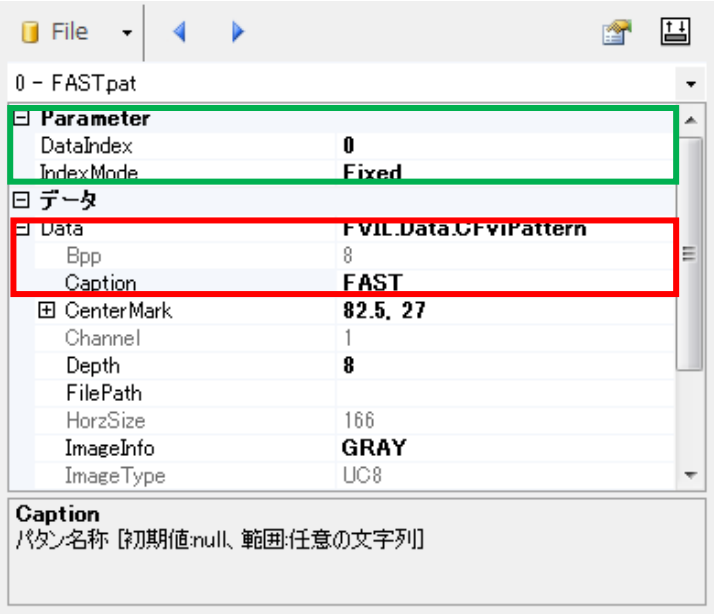
① Function list の「データ入出力」、「データオブジェクト」の順にリストを展開し、「パタン画像」をダブルクリックします。



② TASK タブのワークフローリストの最後の行に「パタン画像#1」が追加されます。



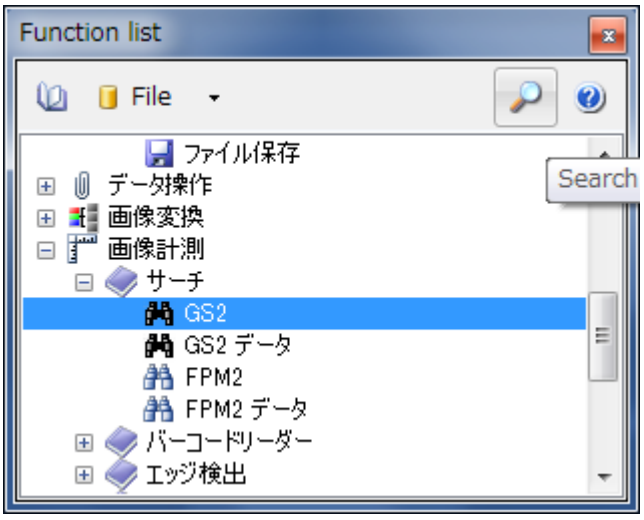
Auto Link によりパタンが関連付けされます。



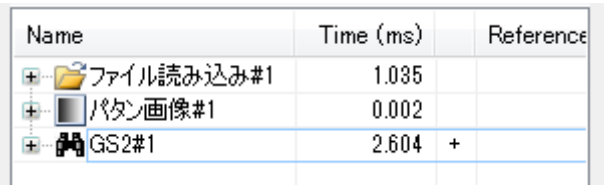
Data の Caption に先ほど登録したパタン名称がない場合、またパタンの名称を変更したい場合は、Caption へパタンの名称を入力してください。(左図赤枠)
複数パタンを登録した場合は、登録順に Index が割り振られているので Parameter の DataIndex を変更することでパタンが変更されます。(左図緑枠)

3.2.3 サーチ実行

① Function list の「画像計測」、「サーチ」の順にリストを展開し、「GS2」をダブルクリックします。

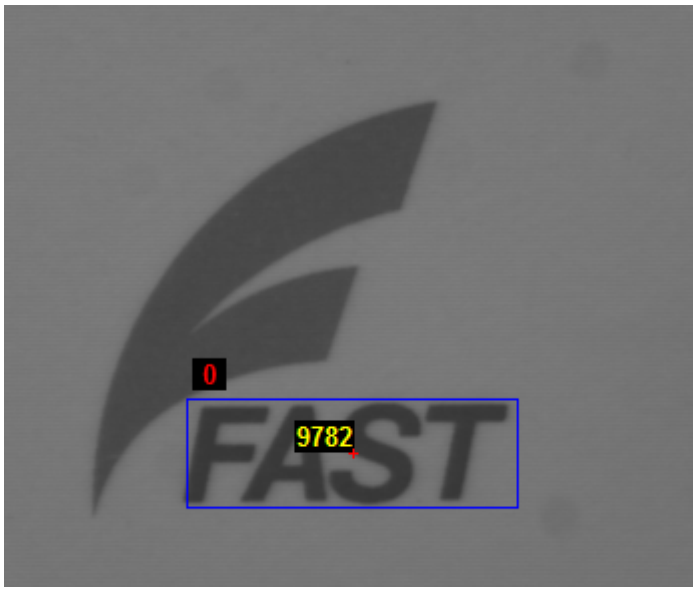


② TASK タブのワークフローリストの最後の行に「GS2#1」が追加されます。

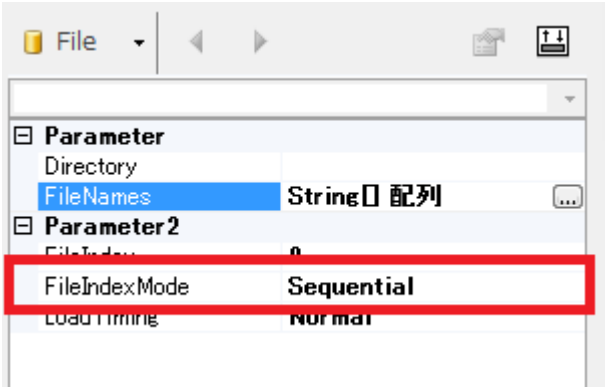


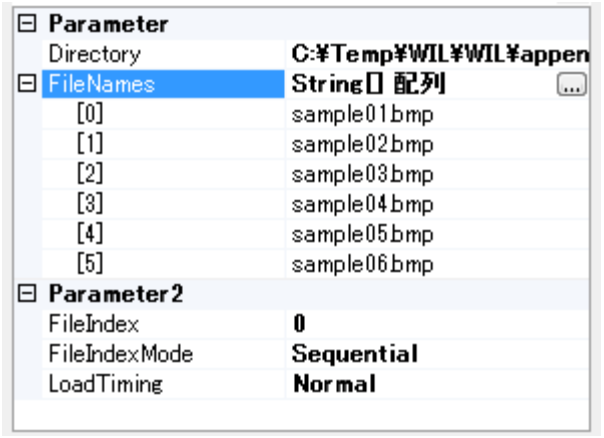
③ 実行アイコンをクリックしてみましょう。

画像ビューに登録したパタンをサーチした画像が表示されます。



複数の画像ファイルを設定することも可能です。「ファイル読み込み」をクリックし、TASK タブの下側にプロパティグリッドが表示されていますので、Parameter2 の FileIndexMode を Fixed から Sequential に変更し、上記のファイル選択時に Ctrl キーを押しながら、ファイル名を複数選択することができます。

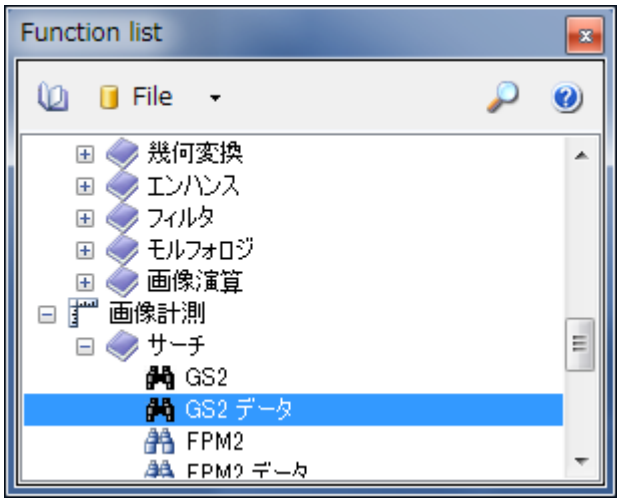




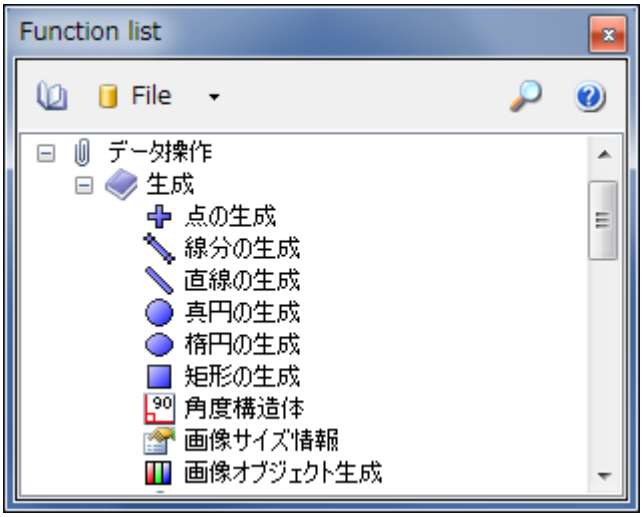
3.2.4 結果の表示等

グレイサーチで得られた結果を使用して、任意の点との 2 点間の距離を計測します。
「GS2」から単一のデータを取得するため、「GS2 データ」を設定します。

① Function list の「画像計測」、「サーチ」の順にリストを展開し、「GS2 データ」をダブルクリックします。

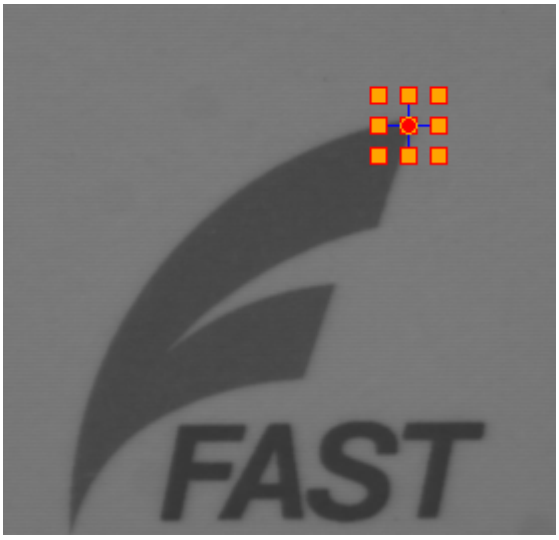


② 続けて、Function list の「データ操作」、「生成」の順にリストを展開し、「点の生成」をダブルクリックします。

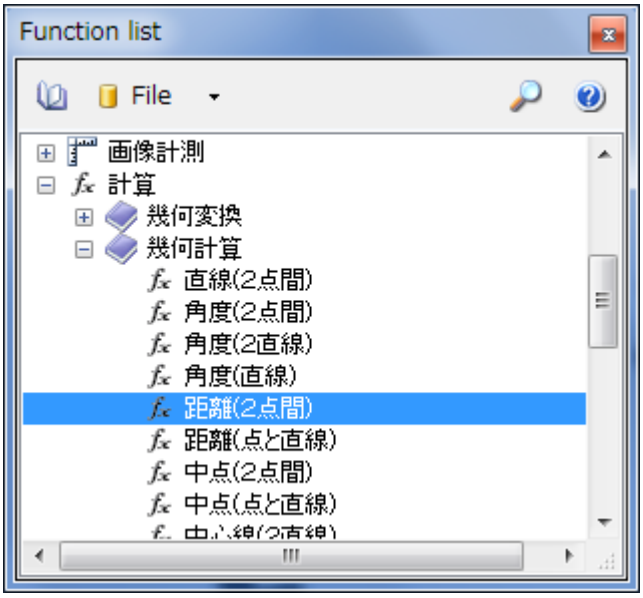


③ TASK タブのワークフローリストの最後の行に「点の生成#1」が追加されます。

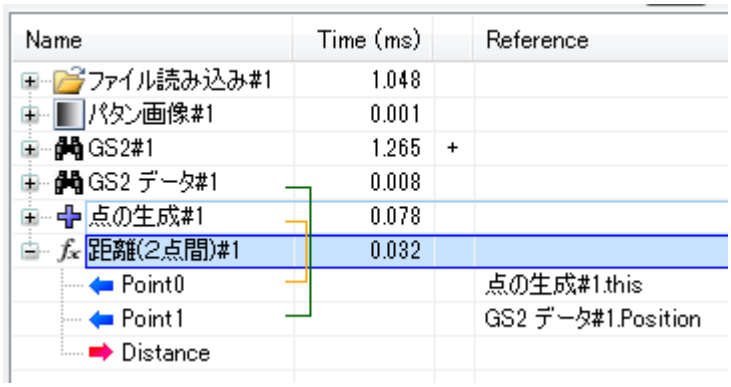
④ ワークフローリストの「点の生成#1」をクリックすると、点の位置をマウスで設定することができますので、任意の位置に設定します。



⑤ Function list の「計算」、「幾何計算」の順にリストを展開し、「距離(2 点間)」をダブルクリックします。

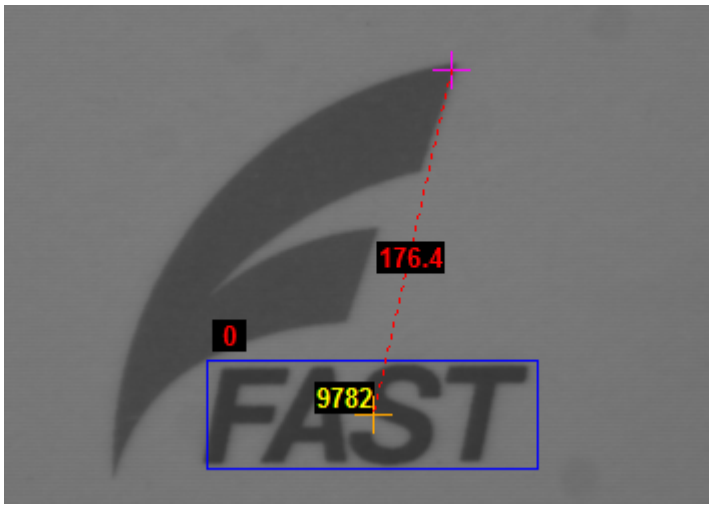


⑥ TASK タブのワークフローリストの最後の行に「距離(2 点間) #1」が追加されます。



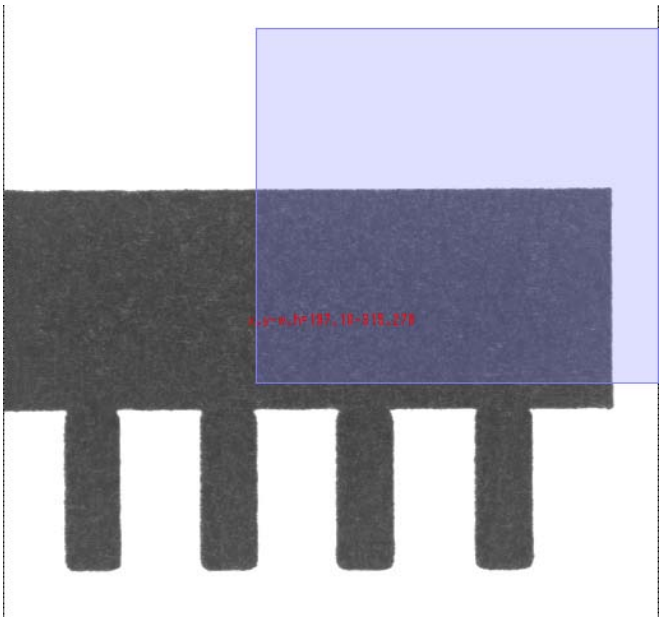
⑦ 実行アイコンをクリックしてみましょう

⑧ 画像ビューに 2 点間を計測した値が、画素で表示されます。



3.3 ハフ検出を設定してみよう

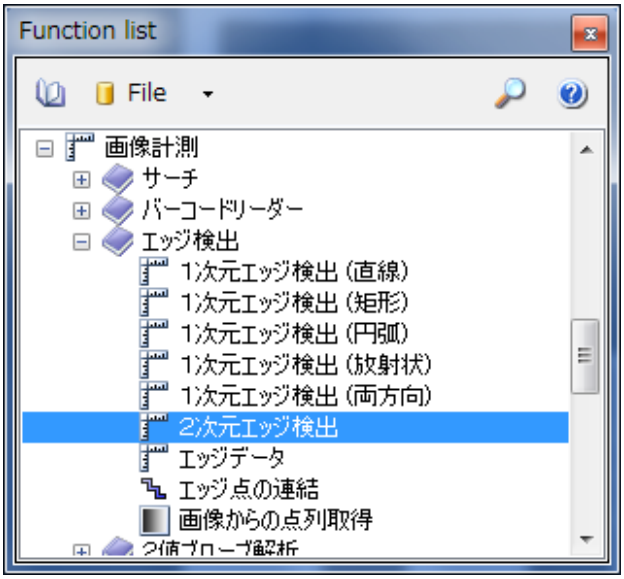
濃淡画像でエッジを抽出し、ハフ変換で直線を検出する機能を設定します。
なお、画像入力や処理範囲の設定は基本操作編で説明しましたので省略します。
使用する画像は、「sample05¥007.bmp」を使用し、処理範囲も設定します。



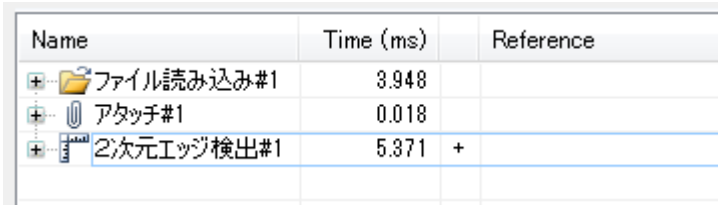
3.3.1 エッジを抽出しよう。

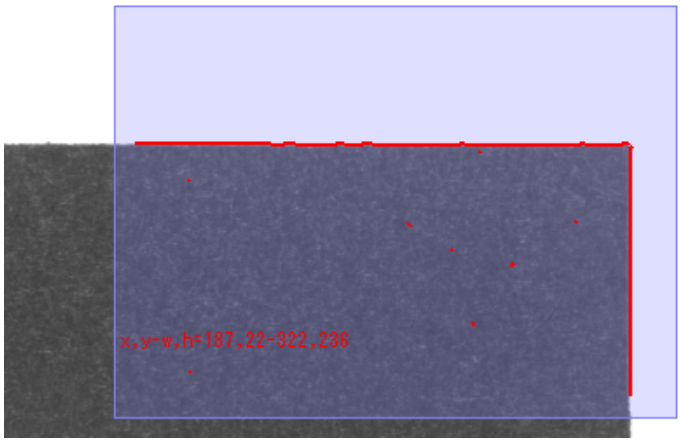
濃淡画像から、エッジを抽出します。

① Function list の「画像計測」、「エッジ検出」の順にリストを展開し、「2 次元エッジ検出」をダブルクリックします。



② TASK タブのワークフローリストの最後の行に「2 次元エッジ抽出#1」が追加されます。

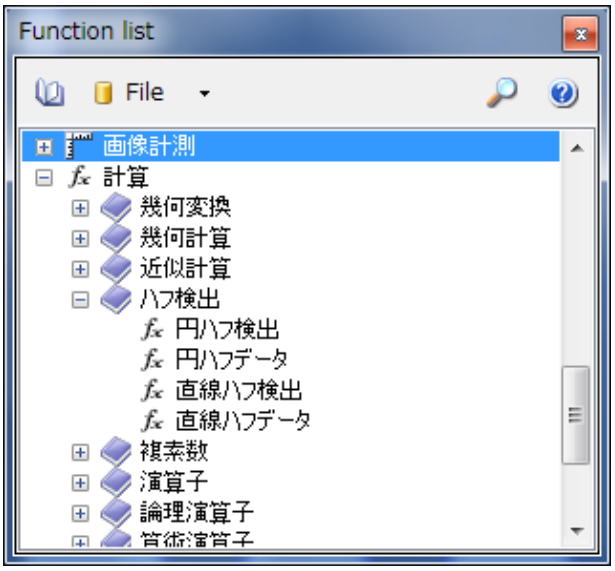




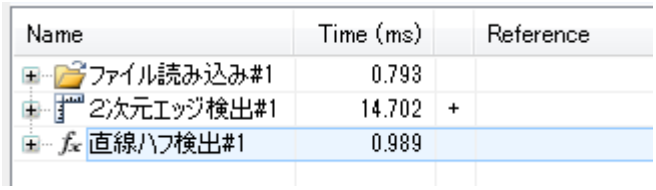
3.3.2 抽出したエッジデータでハフ検出

抽出されたエッジデータでハフ検出を設定し、実行します。

① Function list の「計算」、「ハフ検出」の順にリストを展開し、「直線ハフ検出」をダブルクリックします。

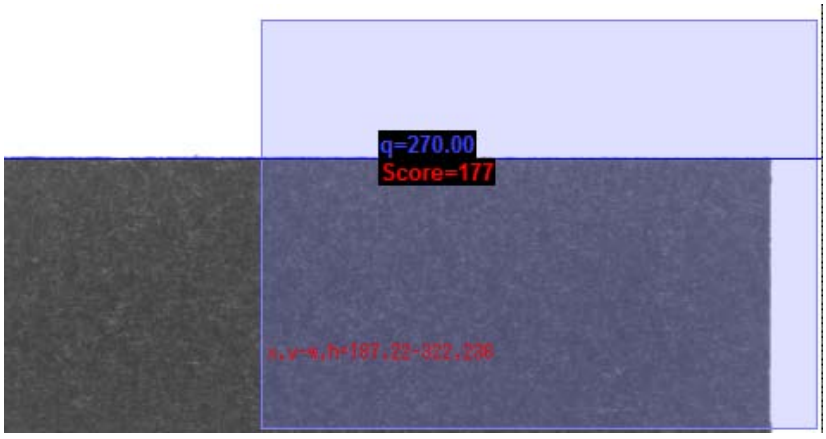


② TASK タブのワークフローリストの最後の行に「直線ハフ検出#1」が追加されます。



③ 実行アイコンをクリックしてみましょう

④ 画像ビューに検出された直線が表示されます。

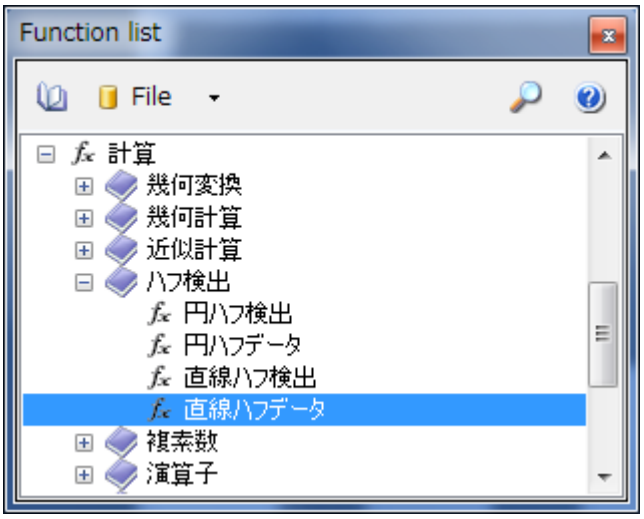


3.3.3 2 直線の交点

ハフ検出で直線を 2 本検出し、その交点を算出する設定をします。

① 「直線ハフ検出」をクリックし、TASK タブの下側にプロパティグリッドが表示されていますので、Parameter の「RequestNum」を「2」に変更します。

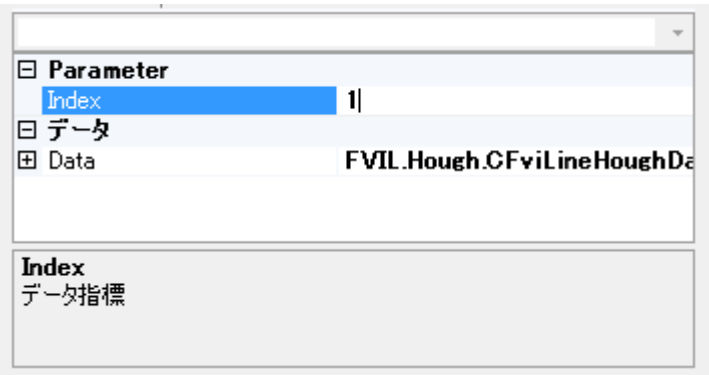
② Function list の「計算」、「ハフ検出」の順にリストを展開し、「直線ハフデータ」をダブルクリックします。



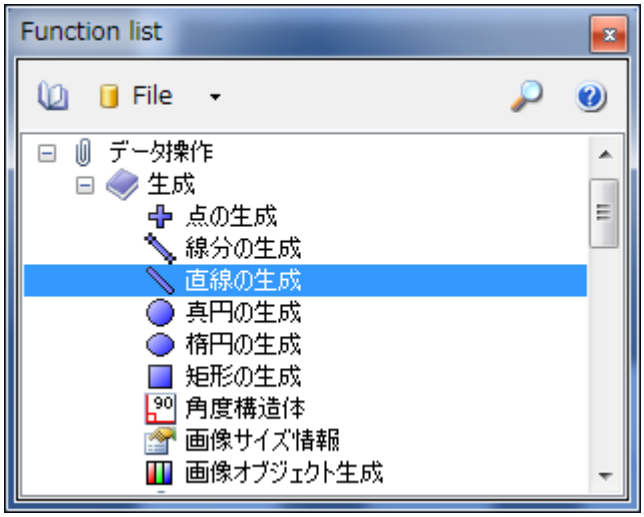
③ さらにもう一つ「直線ハフデータ」を追加します。

④ TASK タブのワークフローリストの最後の行に「直線ハフデータ#2」が追加されます。

⑤ 「直線ハフデータ#2」をクリックし、TASK タブの下側にプロパティグリッドが表示されていますので、Parameter の「Index」を「1」に変更します。

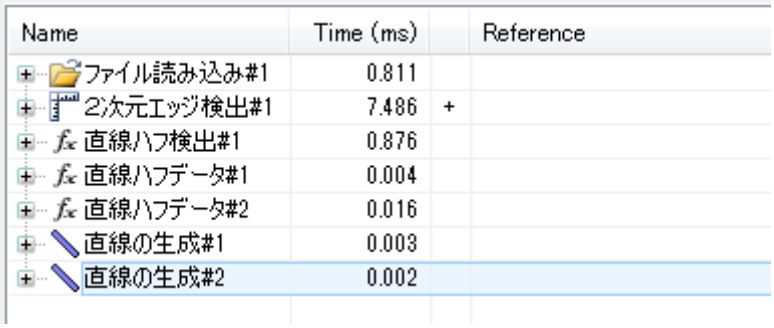


⑥ Function list の「データ操作」、「生成」の順にリストを展開し、「直線の生成」をダブルクリックします。

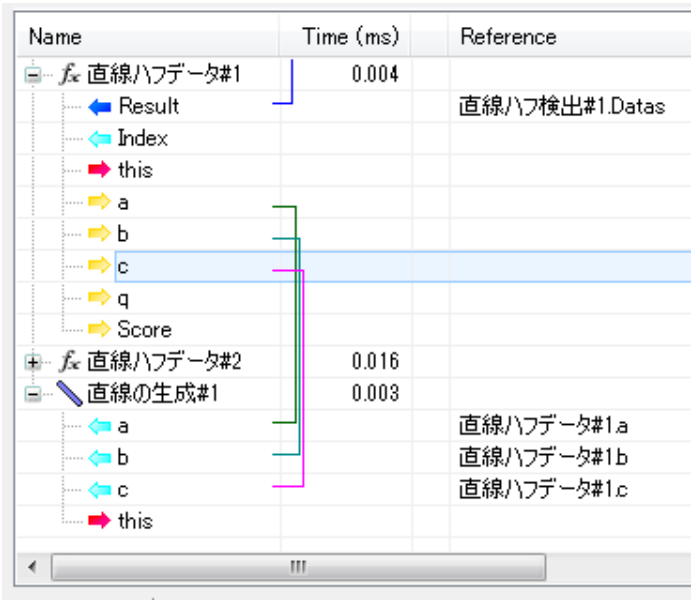


⑦ さらにもう一つ「直線の生成」を追加します。

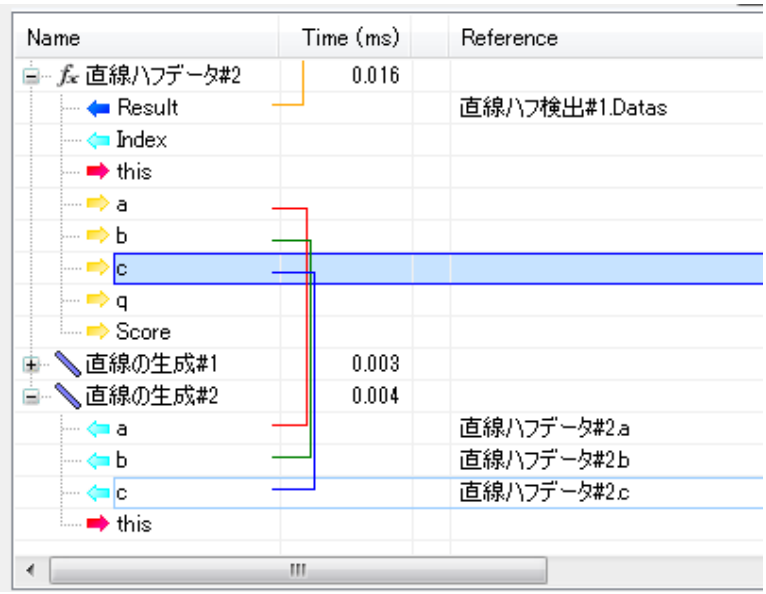
⑧ TASK タブのワークフローリストの最後の行に「直線の生成#2」が追加されます。



⑨ 「直線ハフデータ#1」の値を「直線の生成#1」の値とリンクします。

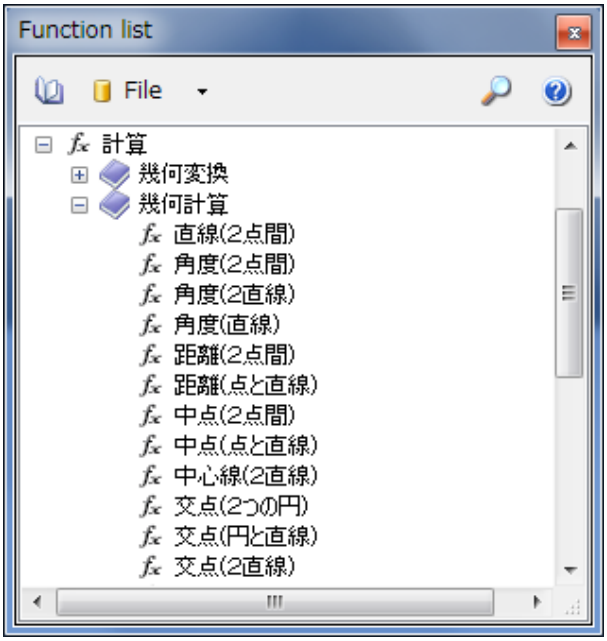


⑩ 「直線ハフデータ#2」の値を「直線の生成#2」の値とリンクします。

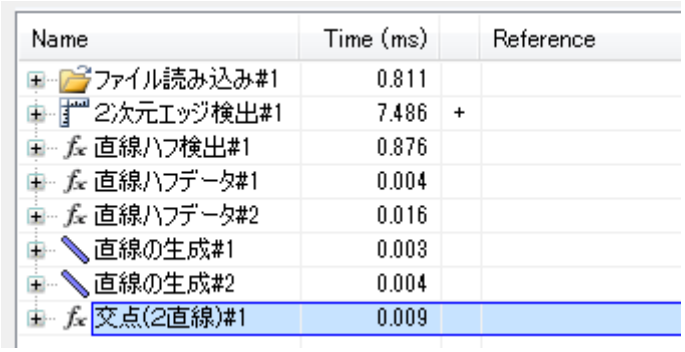


直線の生成を組み入れるのは、「直線ハフデータ」の出力結果を直接、「交点(2 直線)」へ渡せない為です。直線ハフ検出の実行結果のデータ構造は、 ax+by+c=0 の直線式と q(角度)、score で構成されますが、交点(2 直線)に渡すデータ構造、は ax+by+c=0 の直線式だけで構成されています。

⑪ Function list の「計算」、「幾何計算」の順にリストを展開し、「交点(2 直線)」をダブルクリックします。



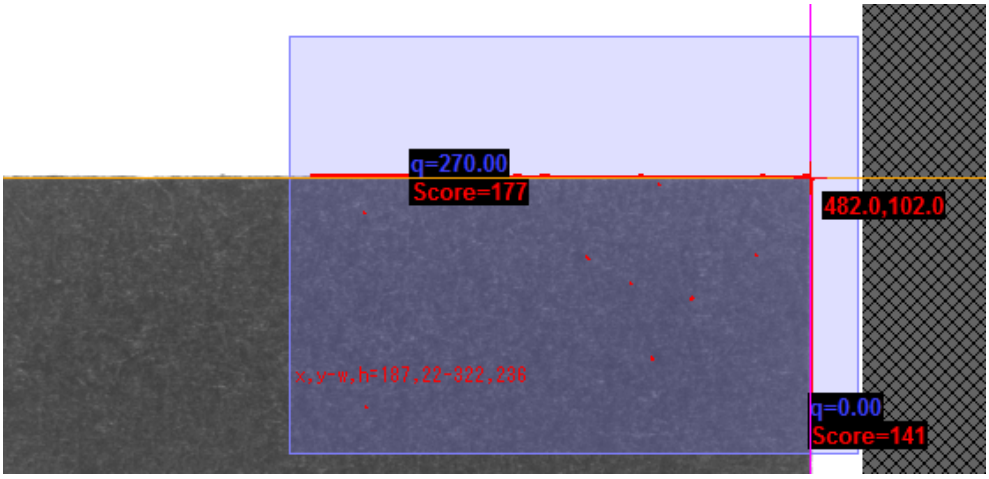
⑫ TASK タブのワークフローリストの最後の行に「交点(2 直線) #1」が追加されます。



(FVIL.Caliper.Function.CrossPoint)

⑬ 実行アイコンをクリックしてみましょう。

⑭ 画像ビューに検出された交点が表示されます。



3.4 FPM2 特徴点応用マッチングを設定してみよう

濃淡画像でパタンを登録し、そのパタンを処理範囲の中から特徴点応用マッチングする機能を設定します。
「3.2 グレイサーチを設定してみよう」の 正規化相関サーチ と比較し、FPM2 特徴点応用マッチング は次の点で優れています。

・パタンの回転、スケール変化に対応できる 。
・シェーディング、背景の変化の影響を受けない 。
・汚れの付着による欠損があっても影響を受けない 。
・対象物の一部が視野外にあっても探索が可能 。

逆に次の点では正規化相関サーチの方が優れています。

・処理時間が非常に高速である

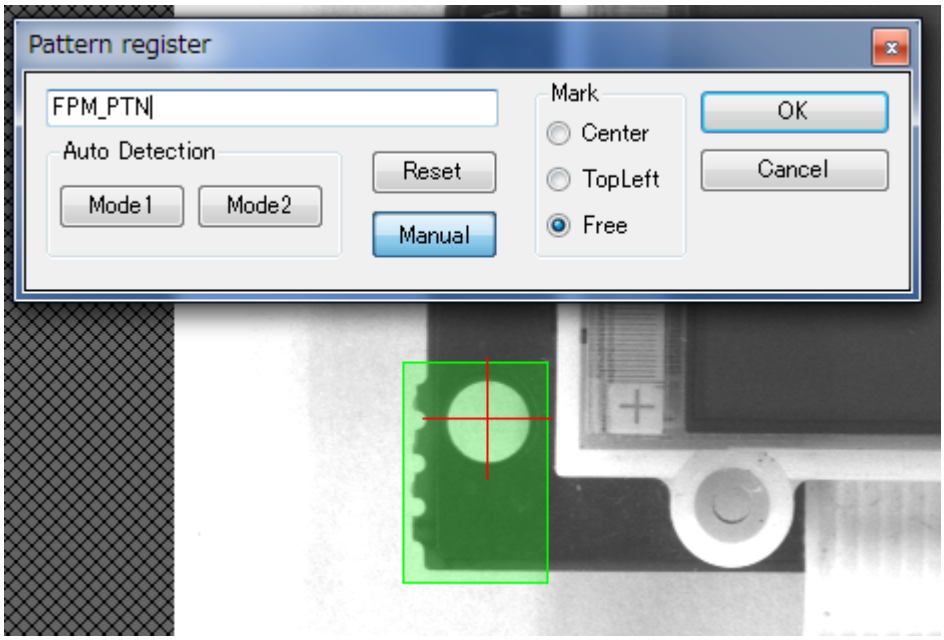
なお、画像入力や処理範囲の設定は基本操作編で説明しましたので省略します。
使用する画像は、「sample00¥sample01.bmp」を使用します。

3.4.1 パタン登録

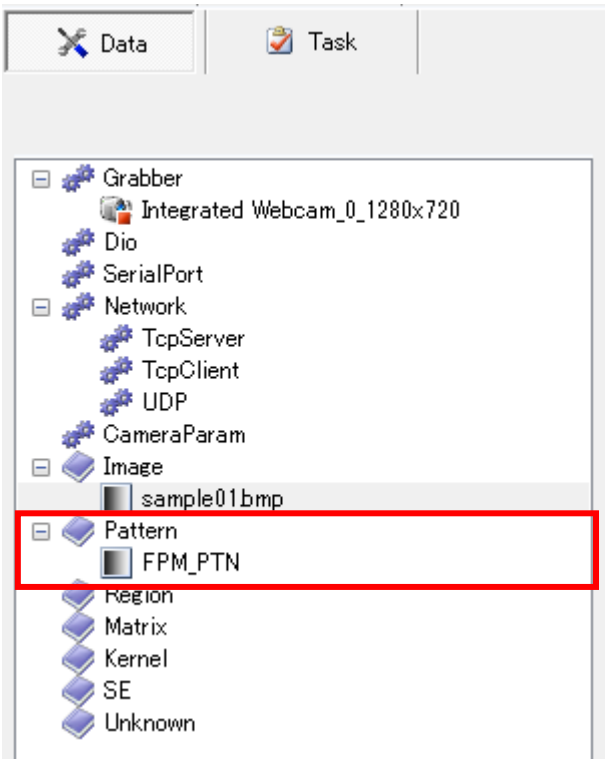
パタンの登録は、ワークフロー内ではなく、メイン画面のツールバーにある「Pattern」をクリックすることで登録することができます。
※登録方法の詳細「WIL-Builder 説明書」参照してください。

① メイン画面のツールバーにある「Pattern」をクリックします。

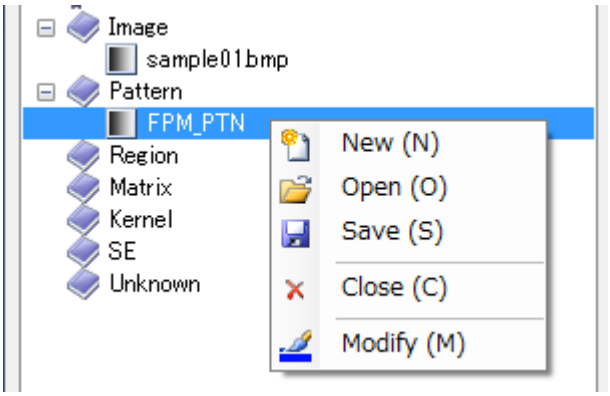
② ツールボックスが表示されます。
③ 登録したい領域をマウスで設定し、パタン名称を記入し登録します。



④ 登録されたパタンは、DATA タブの「Pattern」に表示されます。



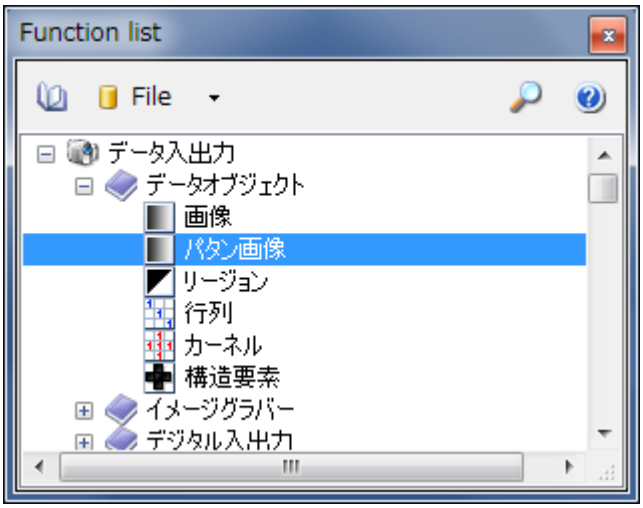
⑤ 登録したパタンを保存する場合は、パタン名称を右クリックし、Save を選択します。



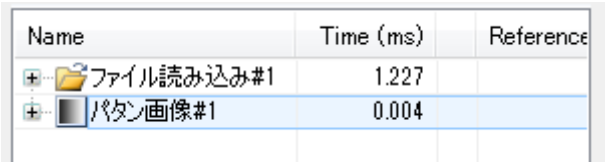
3.4.2 パタンの設定

特徴点応用マッチングを行う為のワークフローを設定します。
メイン画面のツールバーにある「Pattern」をクリックして、パタンは登録したので、それを使用するための手続きを設定します。
なお、画像入力や処理範囲の設定は基本操作編で説明しましたので省略します

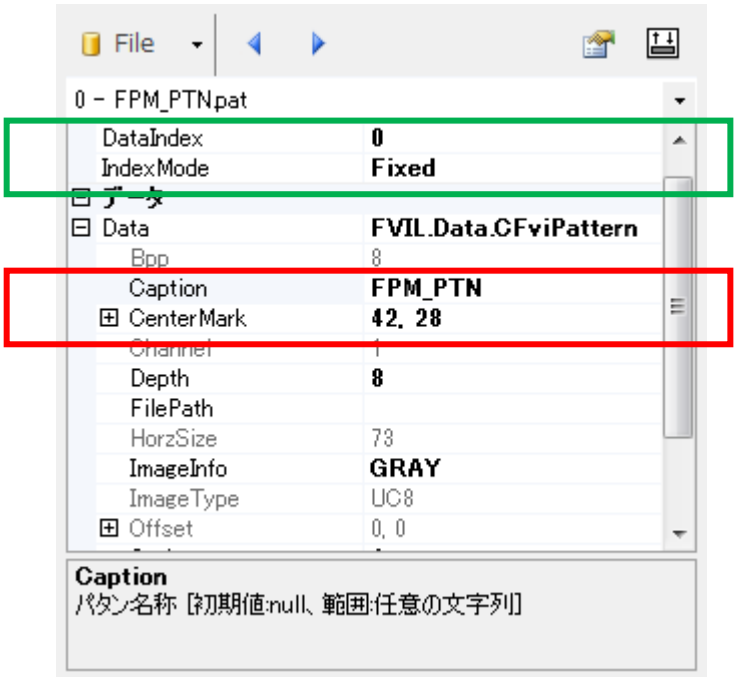
① Function list の「データ入出力」、「データオブジェクト」の順にリストを展開し、「パタン画像」をダブルクリックします。



② TASK タブのワークフローリストの最後の行に「パタン画像」が追加されます。



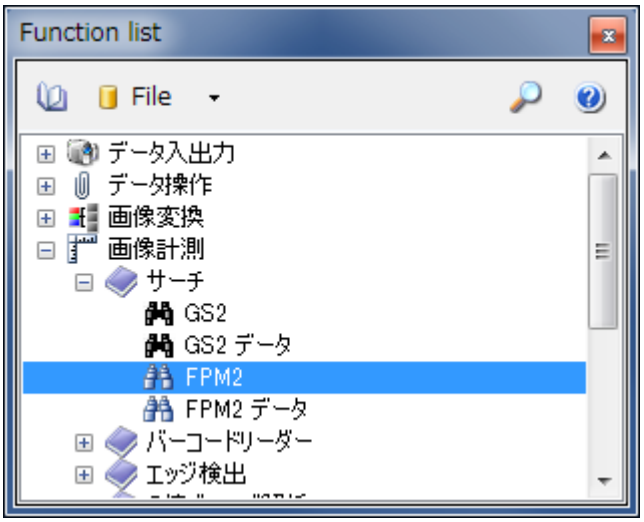
Auto Link によりパタンが関連付けされます。(FVIL.Data.CFviPattern)



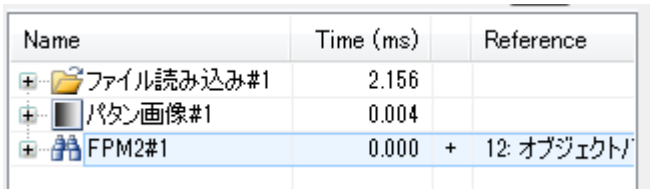
Data の Caption に先ほど登録したパタン名称がない場合、またパタンの名称を変更したい場合は、Caption へパタンの名称を入力してください。(左図赤枠)
複数パタンを登録した場合は、登録順に Index が割り振られているので Parameter の DataIndex を変更することでパタンが変更されます。(左図緑枠)

3.4.3 特徴点応用マッチング実行

① Function list の「画像計測」、「サーチ」の順にリストを展開し、「FPM2」をダブルクリックします。

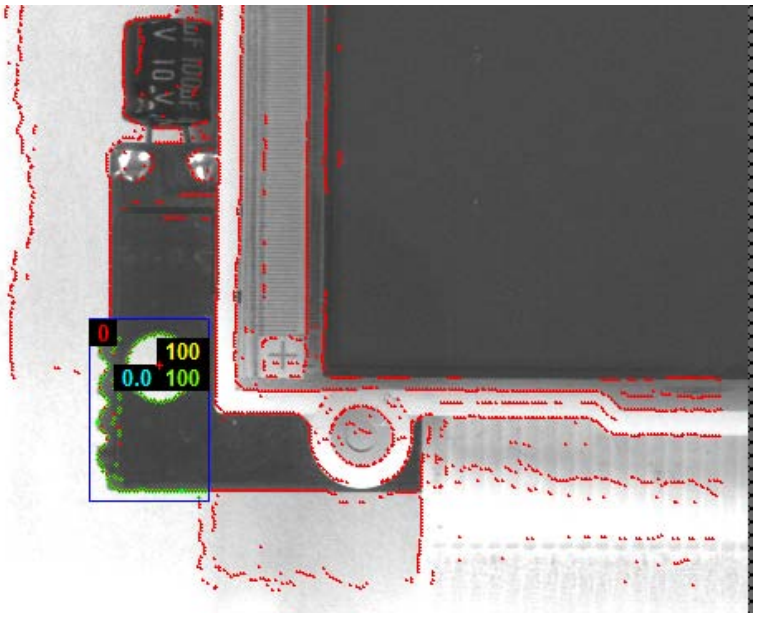


② TASK タブのワークフローリストの最後の行に「FPM2#1」が追加されます。



③ 実行アイコンをクリックしてみましょう。

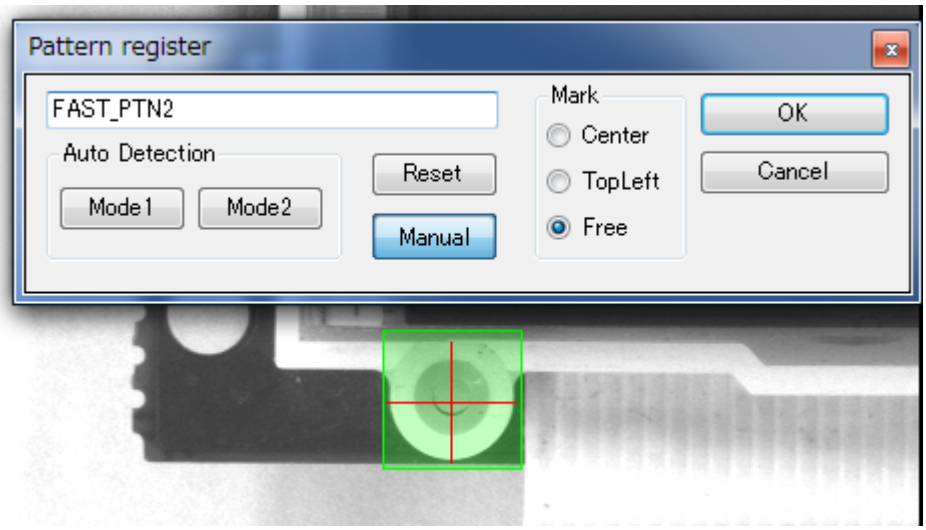
画像ビューに登録したパタンを特徴点応用マッチングした画像が表示されます。



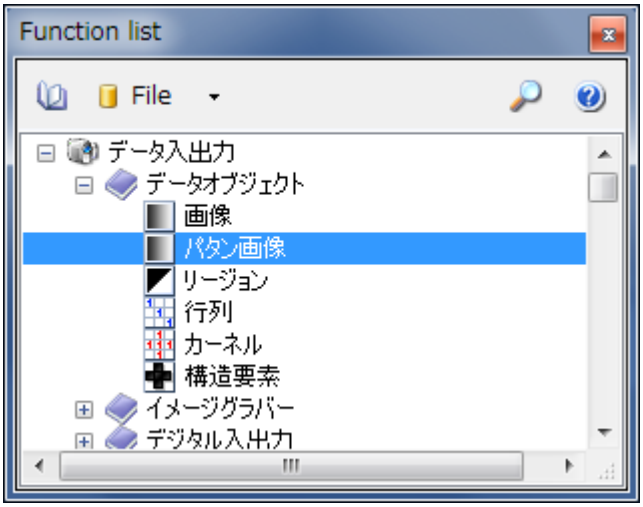
3.4.4 結果の表示等

新たにもう一つパタンを登録し、パタン特徴点応用マッチングで得られた 2 つの結果を使用して、2 点間の距離を計測します。
「FPM2」から単一のデータを取得するため、「FPM2 データ」を設定します。

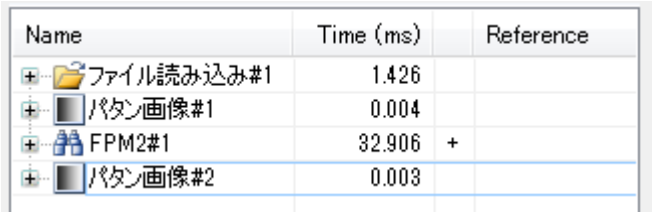
① 新たにパタンを登録します。
登録したい領域をマウスで設定し、パタン名称を記入し登録します。



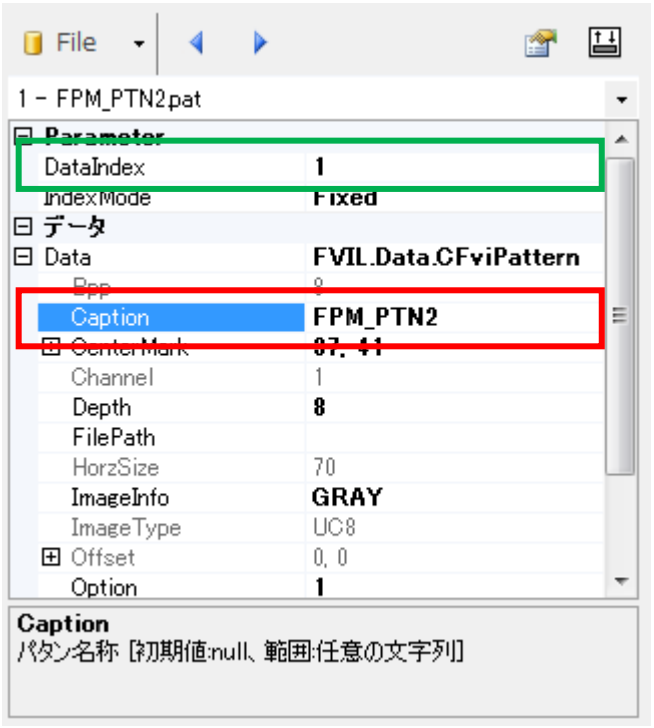
② Function list の「データ入出力」、「データオブジェクト」の順にリストを展開し、「パタン画像」をダブルクリックします。



③ TASK タブのワークフローリストの最後の行に「パタン画像#2」が追加されます。

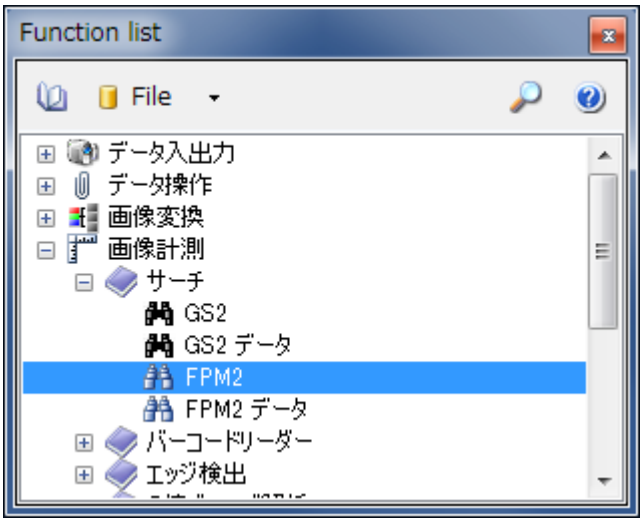


Auto Link によりパタンが関連付けされます。



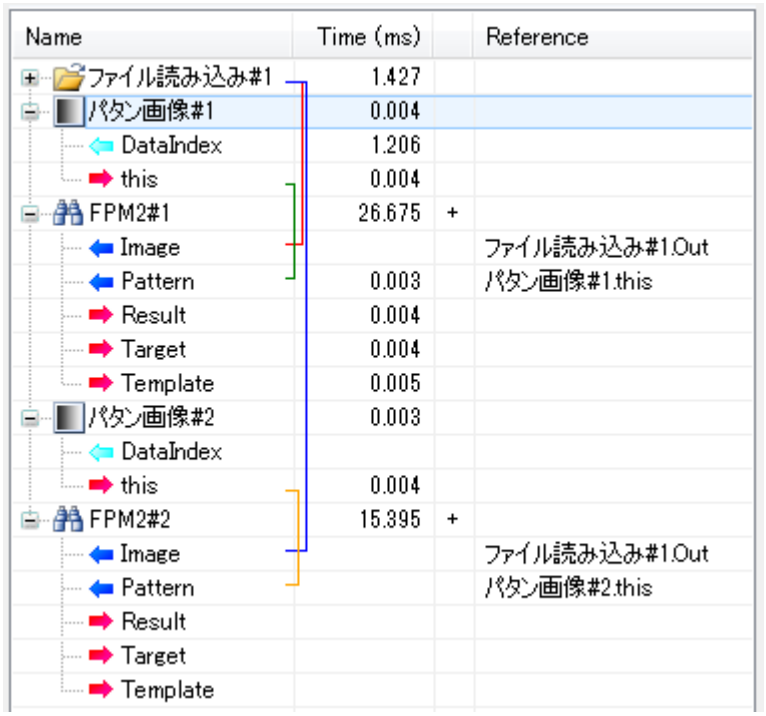
Data の Caption に先ほど登録したパタン名称がない場合、またパタンの名称を変更したい場合は、Caption へパタンの名称を入力してください。(左図赤枠)
複数パタンを登録した場合は、登録順に Index が割り振られているので Parameter の DataIndex を変更することでパタンが変更されます。(左図緑枠)

④ Function list の「画像計測」、「サーチ」の順にリストを展開し、「FPM2」をダブルクリックします。

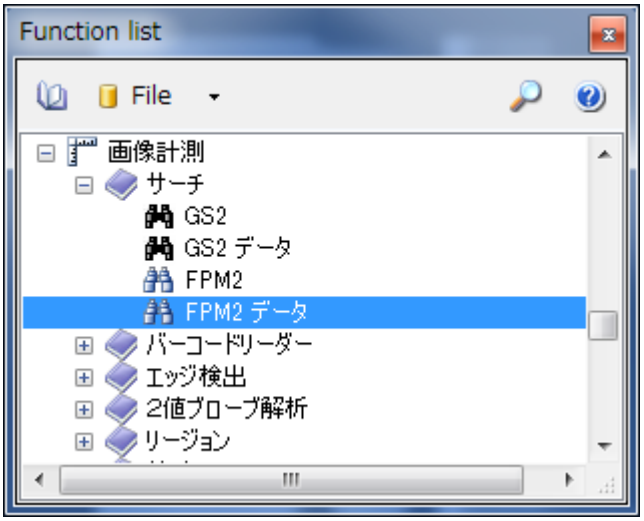


⑤ TASK タブのワークフローリストの最後の行に「FPM2#2」が追加されます。

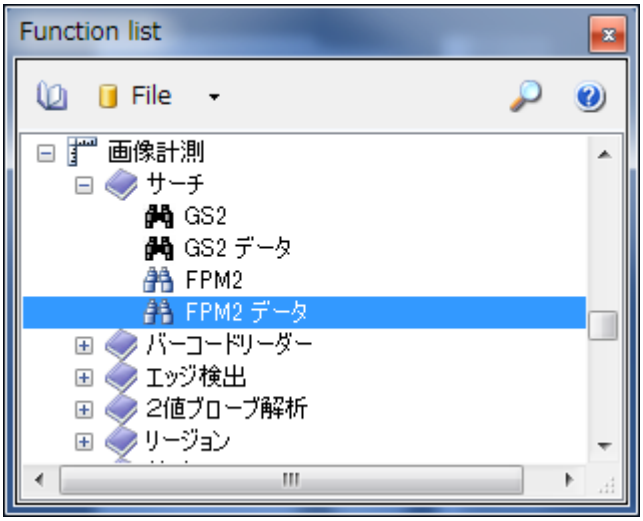
この時、Auto Link でパタンとの関連付けが行われていますが、下の図のような Link になっているかを確認してください。



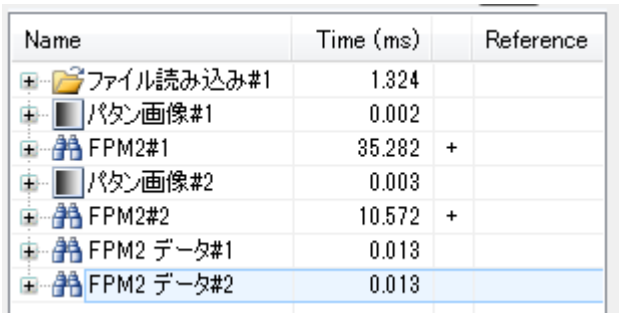
⑥ Function list の「画像計測」、「サーチ」の順にリストを展開し、「FPM2 データ」をダブルクリックします。



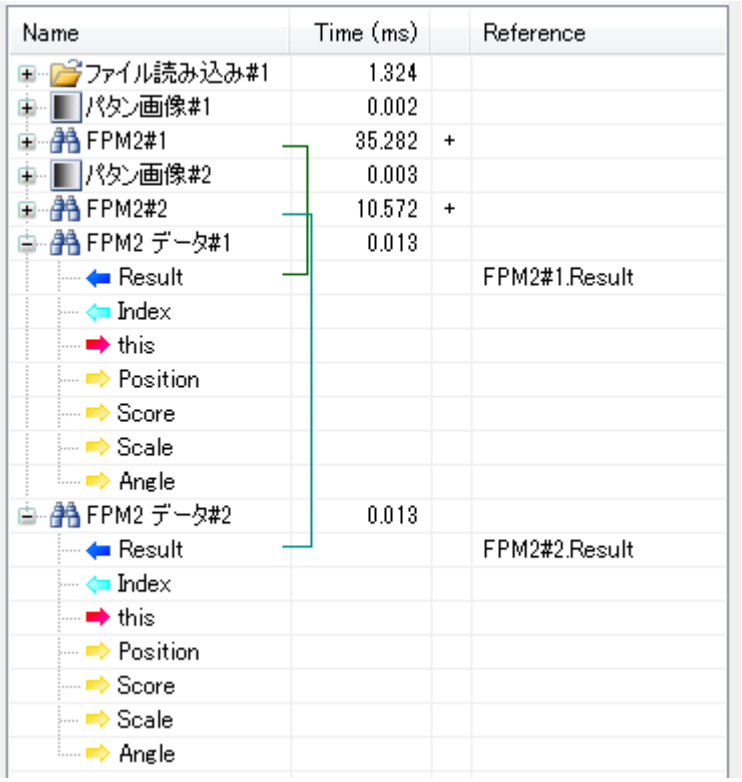
⑦ 更に Function list の「画像計測」、「サーチ」の順にリストを展開し、「FPM2 データ」をダブルクリックします。



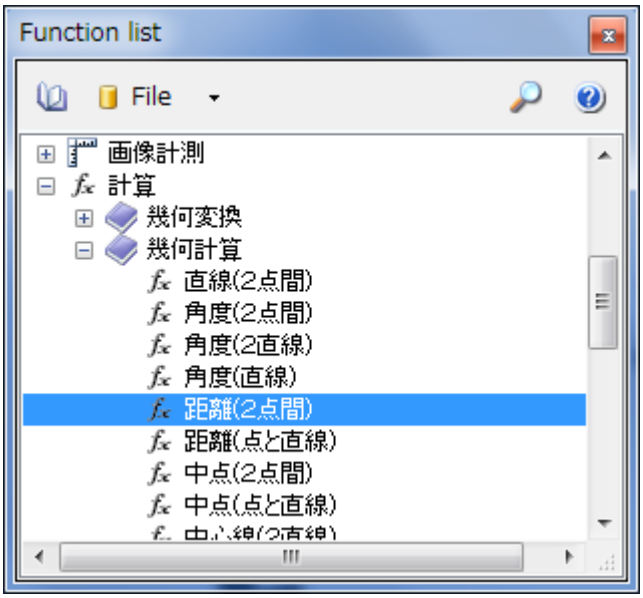
⑧ TASK タブのワークフローリストの最後の行に「FPM2 データ#2」が 2 行追加されます。



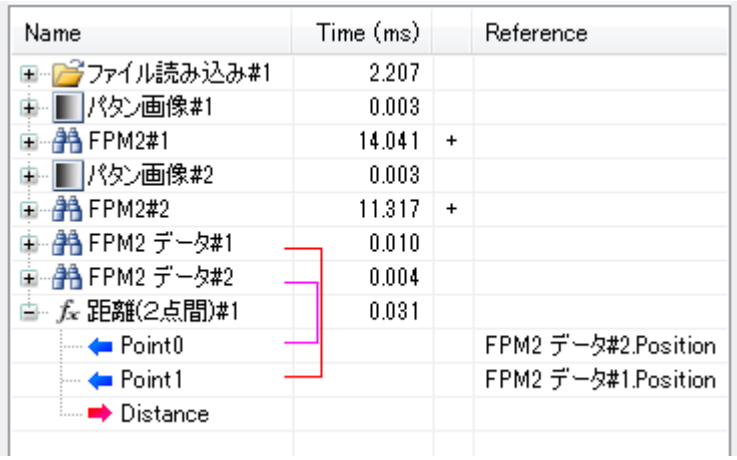
この時、Auto Link でパタンとの関連付けが行われていますが、下の図のような Link になっているかを確認してください。



⑨ Function list の「計算」、「幾何計算」の順にリストを展開し、「距離(2 点間)」をダブルクリックします。

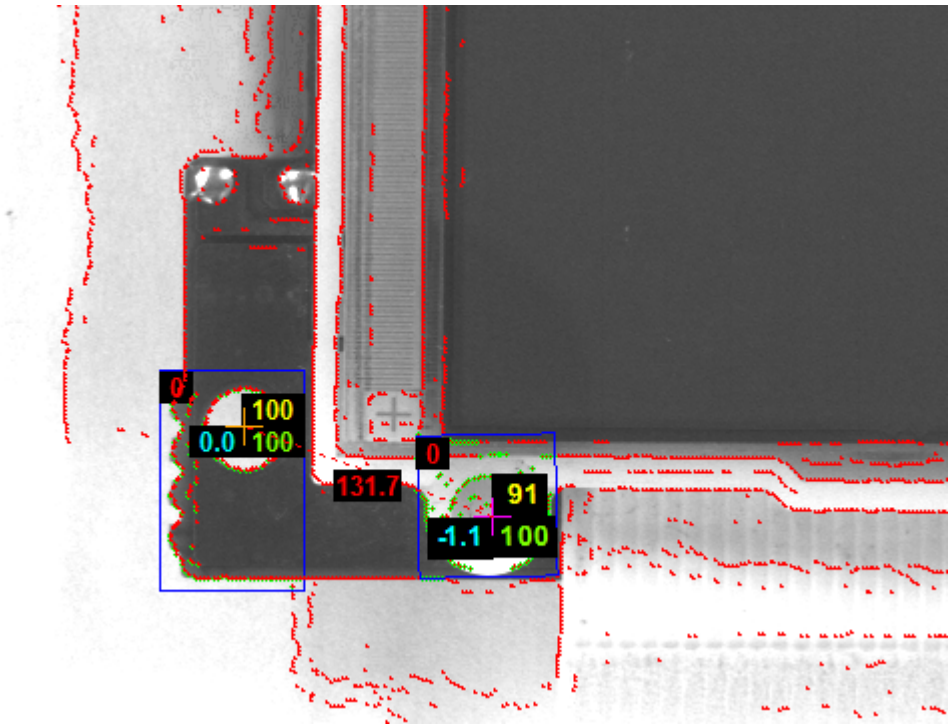


⑩ TASK タブのワークフローリストの最後の行に「距離(2 点間)」が追加されます。



⑪ 実行アイコンをクリックしてみましょう

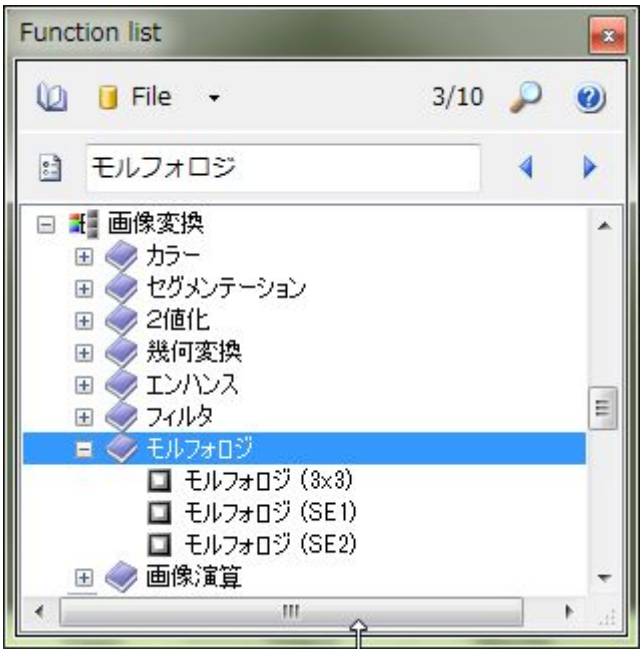
⑫ 画像ビューに 2 点間を計測した値が、画素で表示されます。



3.5 モルフォロジ演算を実行してみよう

3.5.1 モルフォロジの種類

モルフォロジ演算を実行するFunctionは3種類用意しています。

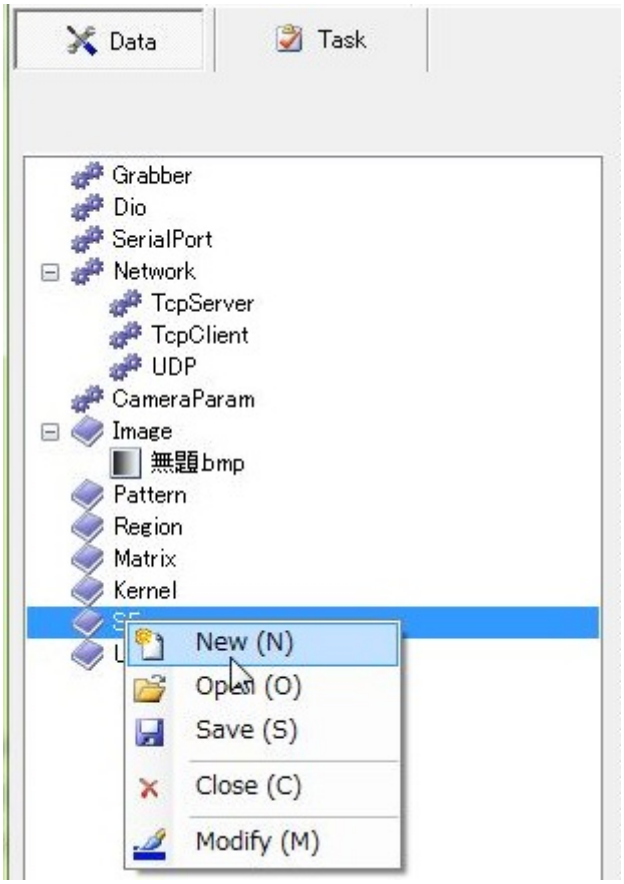


モルフォロジ(3x3)は、3x3サイズの構造要素を使用するフィルタです。対象画像を入力にリンクするだけでモルフォロジ処理が行われます。
モルフォロジ(SE1)およびモルフォロジ(SE2)は、構造要素(SE)を設定します。SE1は、構造要素を1つ、SE2は構造要素を2つ指定します。今回は、モルフォロジ(SE1)を実行します。

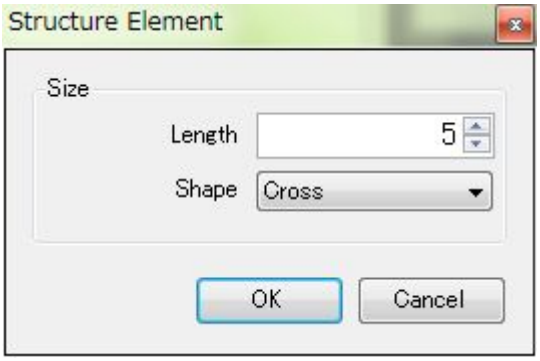
3.5.2 任意の構造要素の生成

モルフォロジ(SE1)に設定する任意の構造要素を生成します。

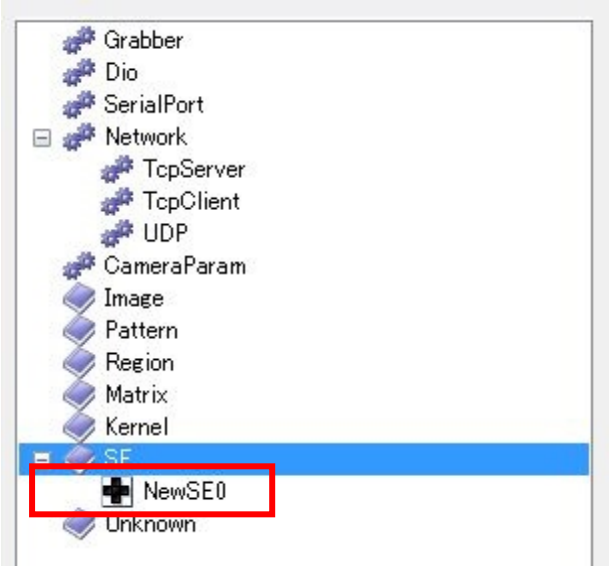
① Data タブの「SE」を右クリックし、New を選択します。



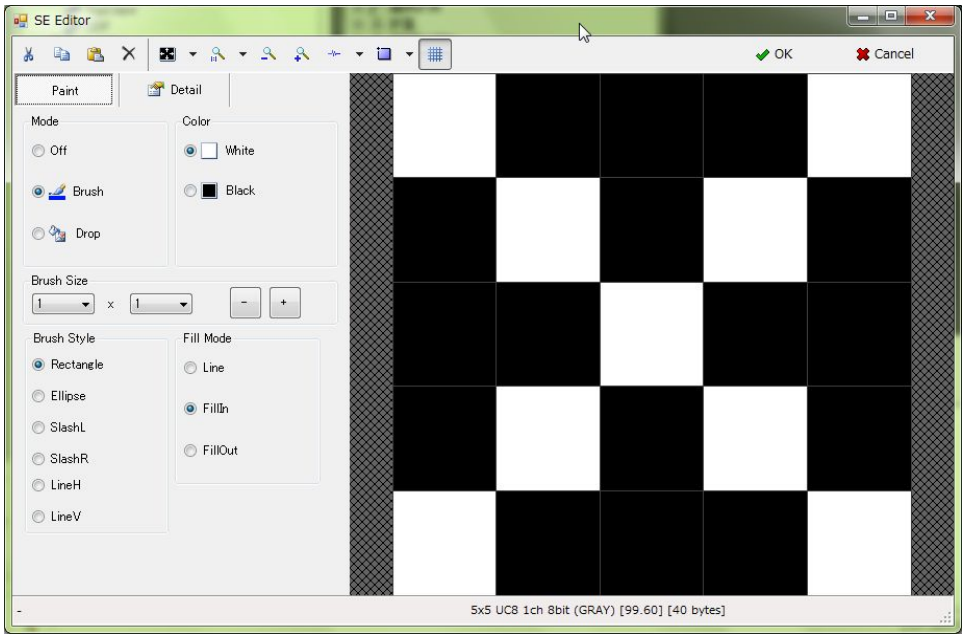
② 構造要素の Length と Shape を指定して OK をクリックします。今回は、**「Length:5」** 、**「Shape：Cross」** の構造要素を作成します。



③ 構造要素「NewSE0」が生成されます。



④ 生成された構造要素を参照・編集する場合は、「NewSE0」をダブルクリックして SE Editor を表示します。

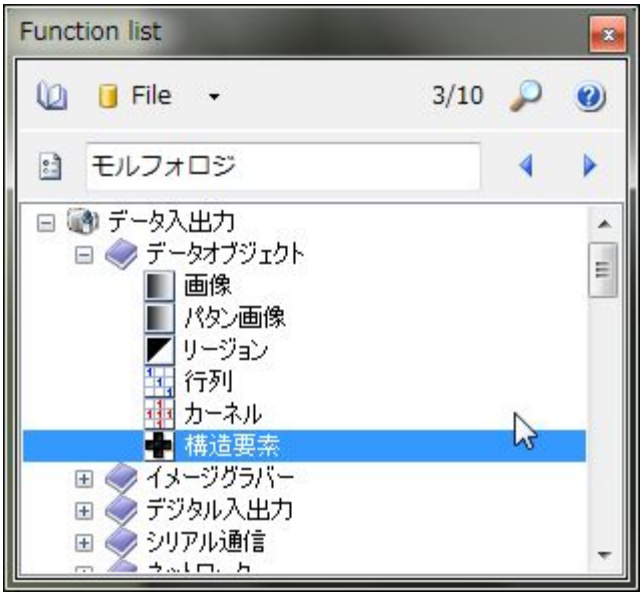


（「Length:5」「Shape：Cross」）

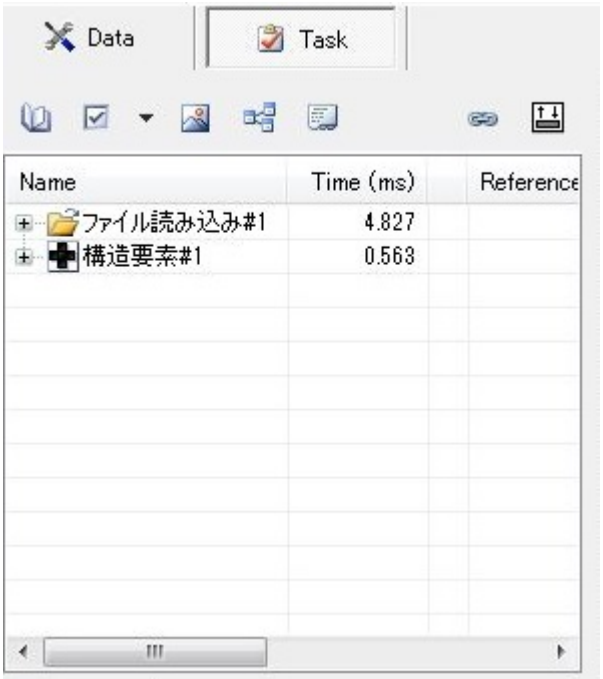
3.5.3 モルフォロジの実行

モルフォロジを行うためのワークフローを設定します。Data タブの SE で作成した構造要素を使用するための設定を行います。なお、画像入力については省略します。

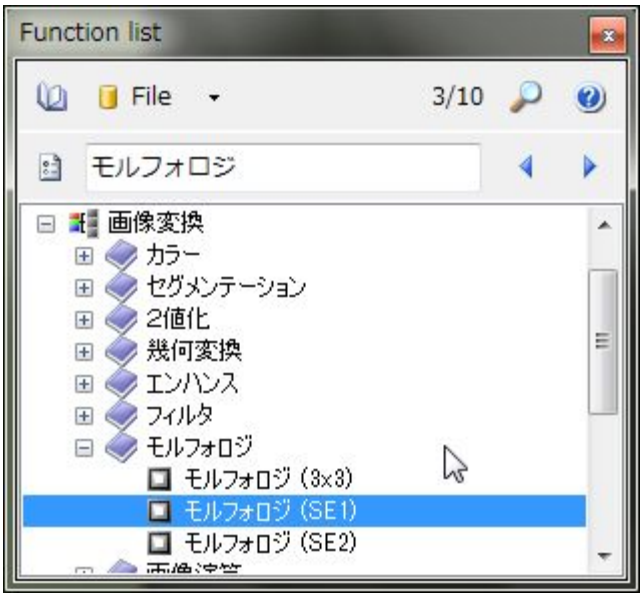
Function list に戻り「データ入出力」、「データオブジェクト」の順にリストを展開し、「構造要素」をダブルクリックします。



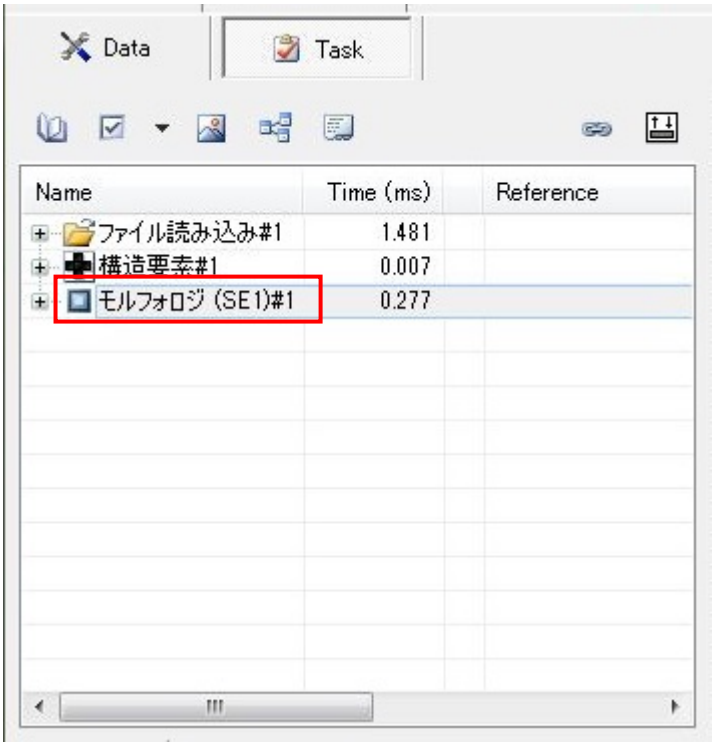
① TASK タブのワークフローリストの最後の行に「構造要素#1」が追加されます。



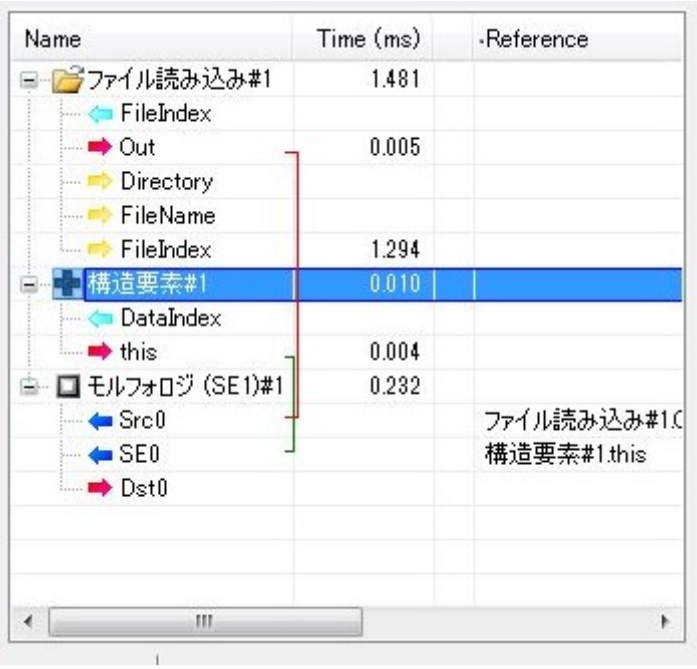
② Function list に戻り「画像変換」、「モルフォロジ」の順にリストを展開し、「モルフォロジ(SE1)」をダブルクリックします。



③ TASK タブのワークフローリストの最後の行に「モルフォロジ(SE1)＃1)」が追加されます。

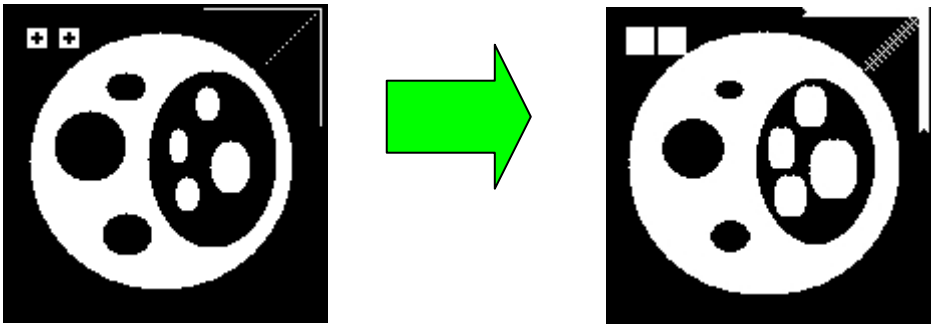


④ 「ファイル読み込み#1」の「Out」を「モルフォロジ(SE1)#1」の「Src0」に、「構造要素」の「this」を「モルフォロジ(SE1)#1」の「SE0」にリンクします。



⑤ 実行アイコンをクリックしてみましょう。

⑤ モルフォロジ(SE1)の実行結果が表示されます。



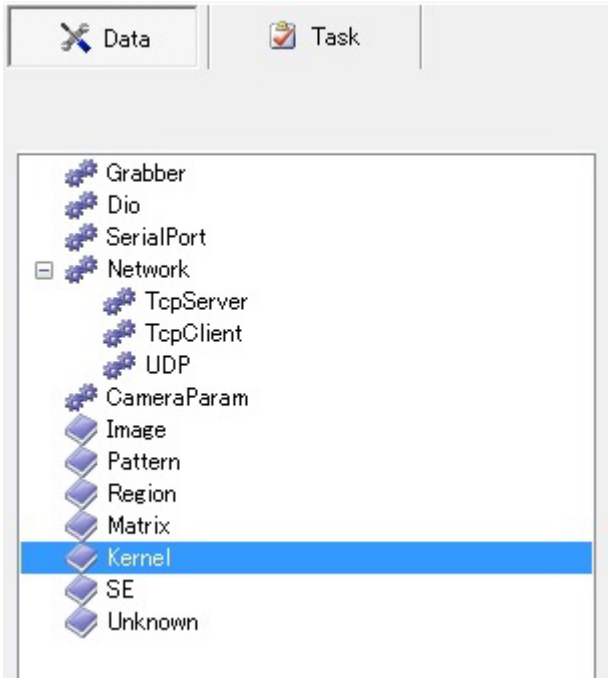
3.6 任意カーネルフィルタを設定してみよう

Function list の「画像変換」、「フィルタ」の順に展開したリストのノードにある「カーネルフィルタ」以外に関しては、選択したフィルタ固有の画素領域と値にて画像処理を行います。
「2.2 簡単な画像処理をしてみよう」の「ソーベル」では、用意されている 3×3 画素領域のソーベル微分フィルタを使用しています。
一方「カーネルフィルタ」は、任意のサイズと任意の値の画素領域を作成し、それを使用して画像処理を行うことができます。
ここでは、入力した画像に対して、任意カーネルフィルタを使用する方法での画像処理を説明します。
なお、画像入力を行う「ファイル読み込み」に関しては基本操作編を参照して下さい。

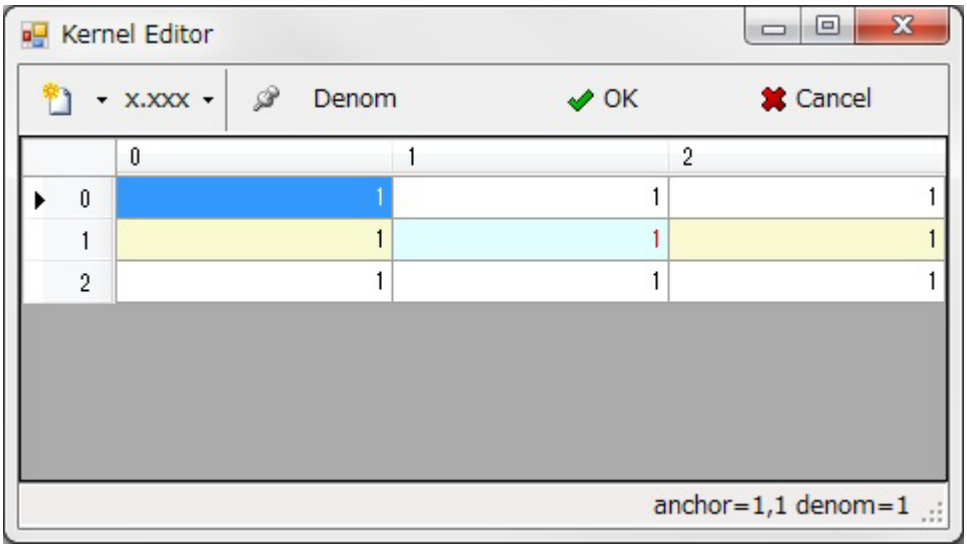
3.6.1 Data タブのカーネルオブジェクトの作成

カーネルフィルタに設定するカーネルオブジェクトを作成します。

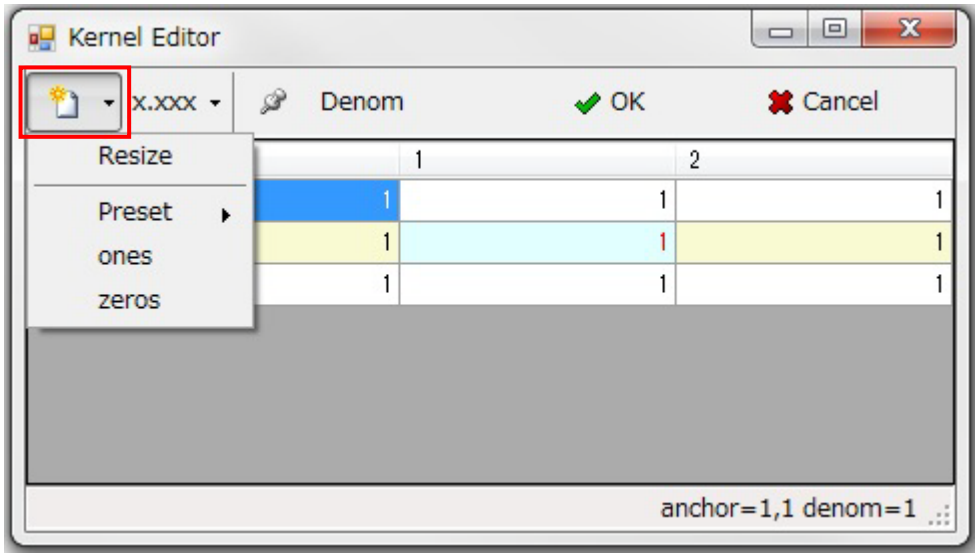
① Data タブの「Kernel」を右クリックし、New を選択します。



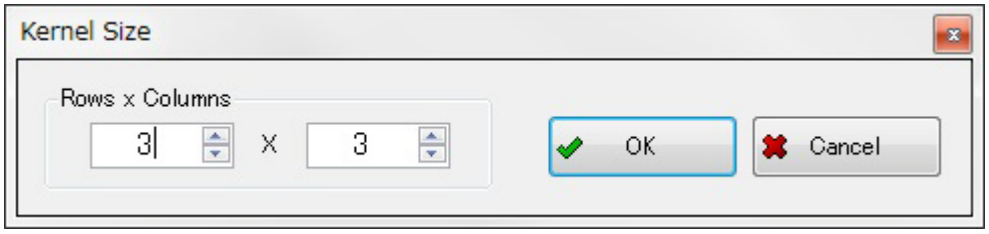
② カーネル編集フォームが表示されます。



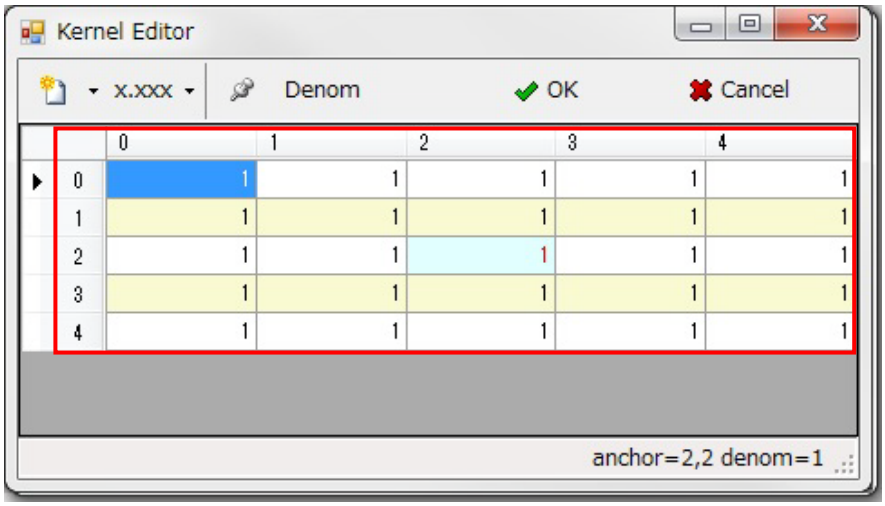
③ カーネル編集フォームにて編集を行い、カーネルオブジェクトを作成します。
New ボタン(赤矩形)をクリックし、Resize を選択します。



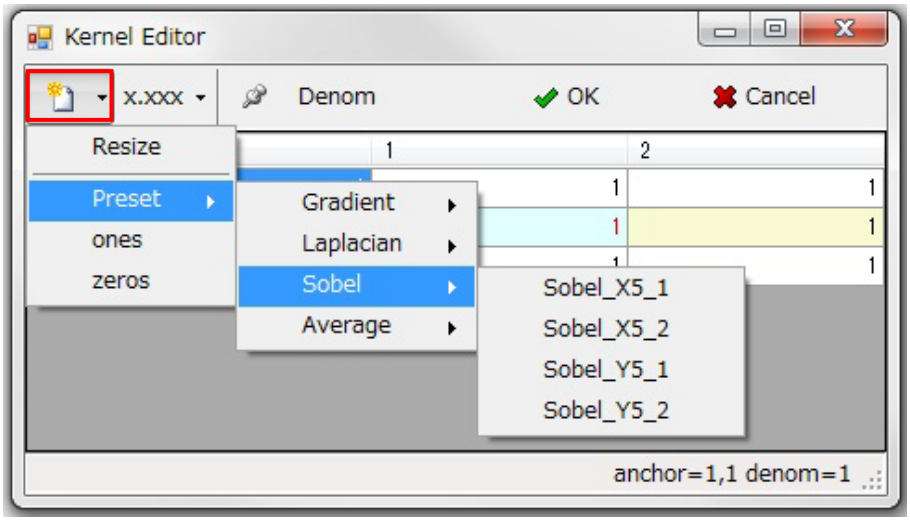
④ カーネルサイズ編集フォームが表示されます。
「Rows」と「Columns」に任意の値を設定し、OK ボタンを押してサイズを決定します。



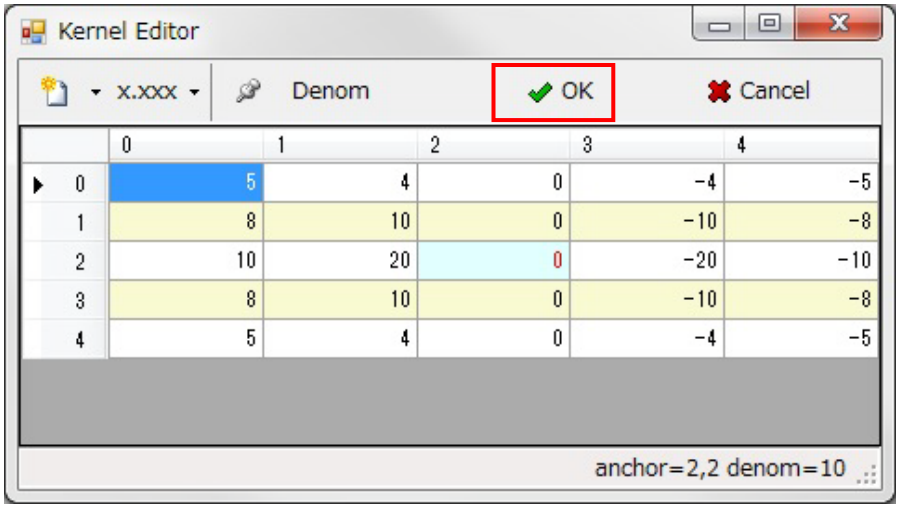
⑤ たとえば、5×5 に設定し作成すると下記の様になりますので、赤矩形部を編集し、OK ボタンを押すとカーネルオブジェクトが作成されます。



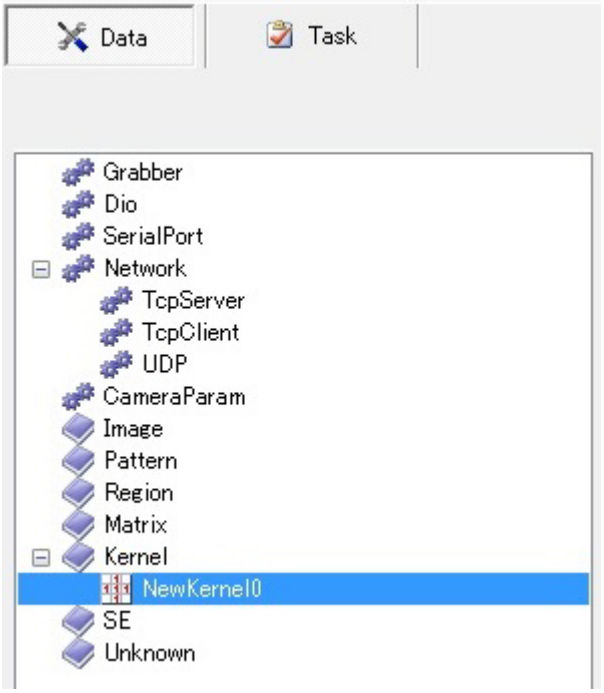
⑥ また、カーネル編集フォームには、プリセットデータが用意されていますので、New ボタン(赤矩形)をクリックし、「Preset」からプリセットデータを選択する事も可能です。
以下の説明では「Sobel」の「Sobel_X5_1」を使用して、画像処理を行います。



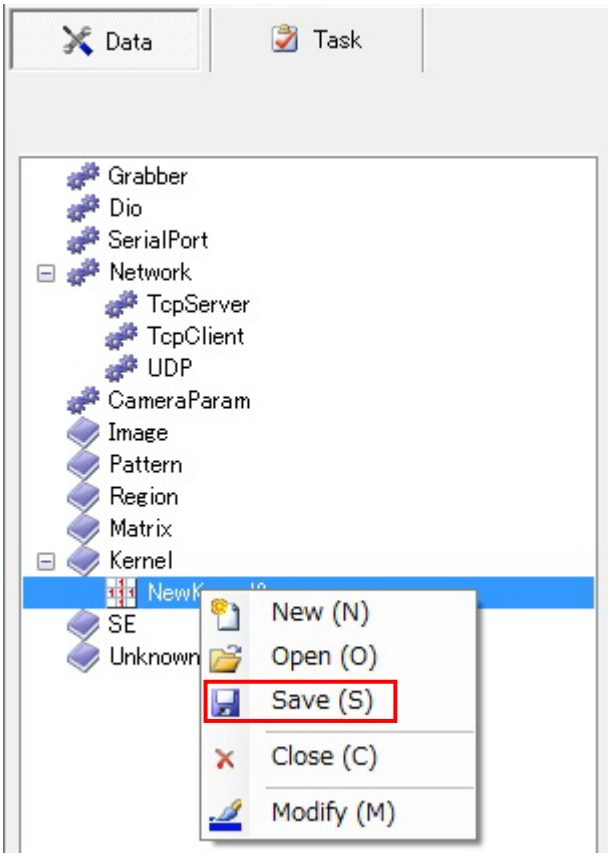
⑦ 「Sobel_X5_1」を選択すると、下記の様にプリセットデータが設定されます。
OK(赤矩形)ボタンを押してカーネルオブジェクトを作成します。



⑧ 作成したカーネルオブジェクトは、DATA タブの「Kernel」に追加されます。



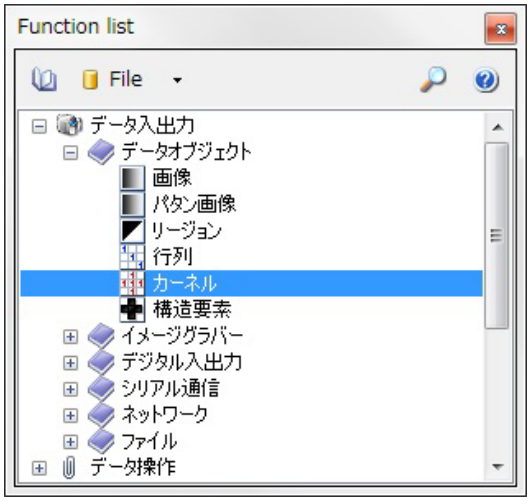
⑨ 作成したカーネルオブジェクトを保存する場合は、カーネル名称を右クリックし、Save を選択します。



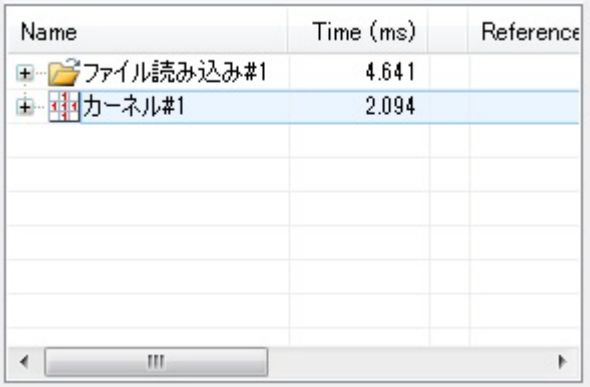
3.6.2 Data タブのカーネルオブジェクトの参照

作成したカーネルオブジェクトを使用する為、ワークフローを設定します。

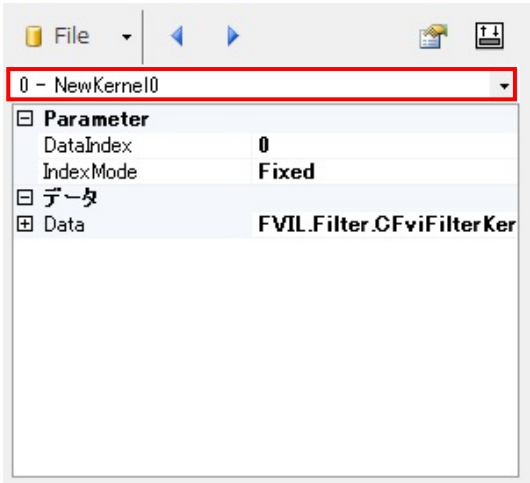
① Functionlist の「データ入出力」、「データオブジェクト」の順にリストを展開し、「カーネル」をダブルクリックします。



② TASK タブのワークフローリストの最後の行に「カーネル」が追加されます。



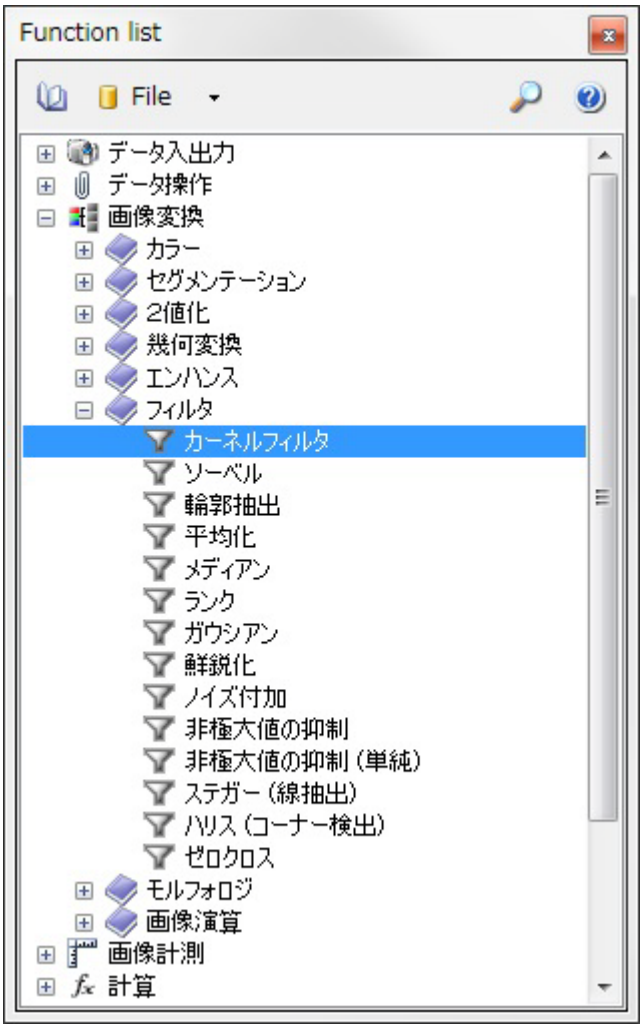
③ 「カーネル」をクリックすると、TASK タブの下側にプロパティグリッドが表示されますので、赤矩形部のリストボックスか青カーソルにて、作成したカーネルオブジェクトを選択します。



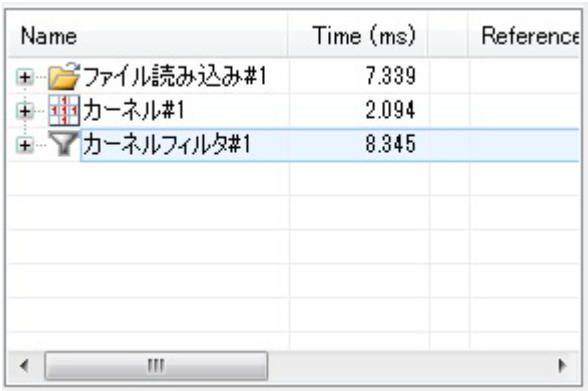
3.6.3 任意カーネルフィルタの実行

「ファイル読み込み」にて入力した画像に対して、作成したカーネルフィルタにて画像処理を実行します。

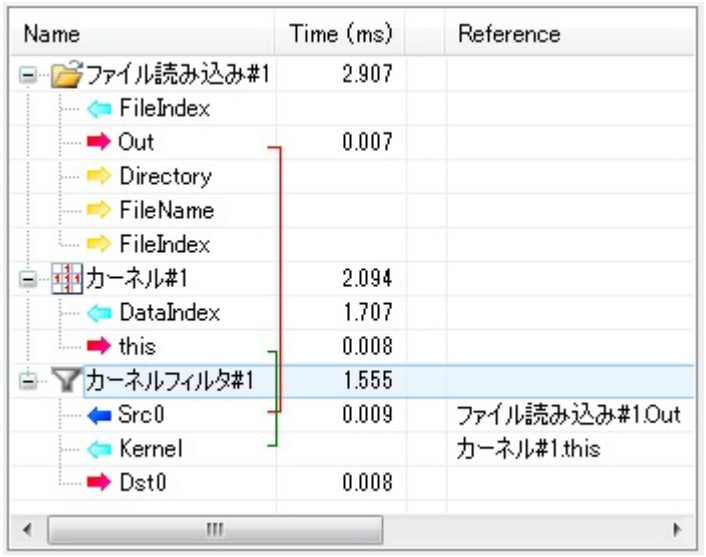
① Function list の「画像変換」、「フィルタ」の順にリストを展開し、「カーネルフィルタ」をダブルクリックします。



② TASK タブのワークフローリストの最後の行に「カーネルフィルタ#1」が追加されます。

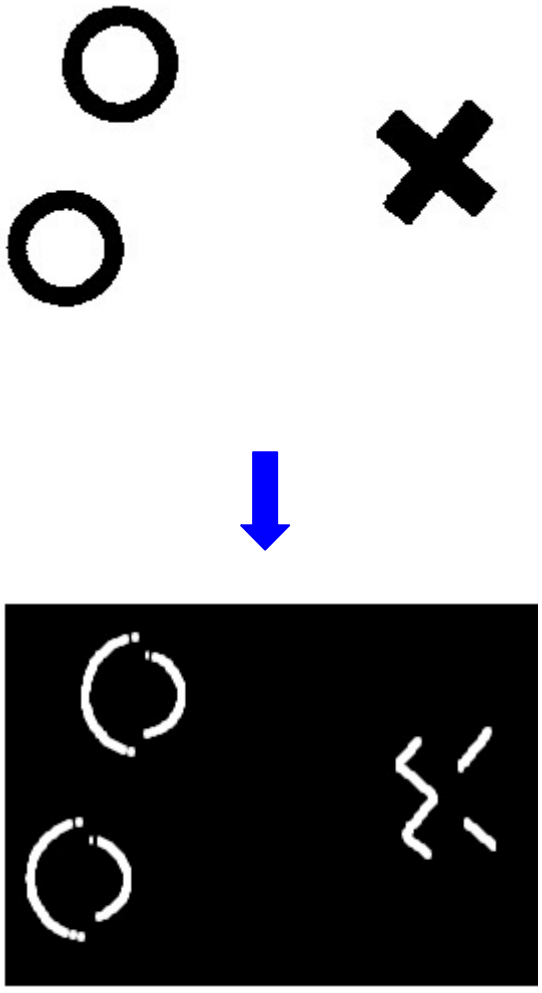


③ 「ファイル読み込み」の「Out」を「カーネルフィルタ」の「Src0」に、「カーネル」の「this」を「カーネルフィルタ」の「Kernel」にリンクします。



④ 実行アイコンをクリックしてみましょう。

フィルタ処理が行われた画像が表示されます。



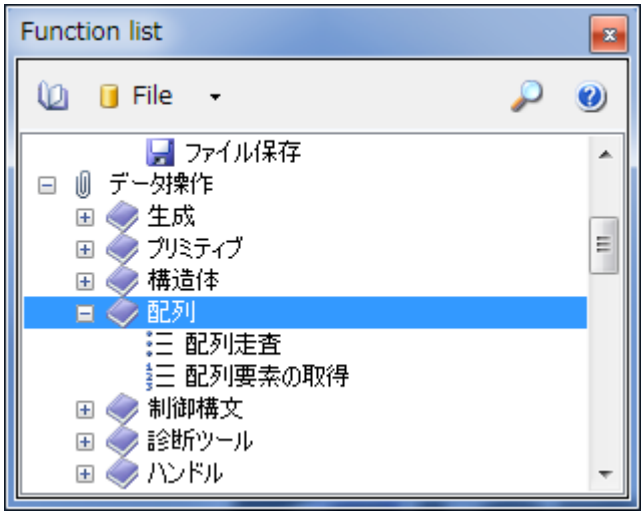
4. 応用編

WIL-Builder は配列走査や制御構文も行う事ができます。

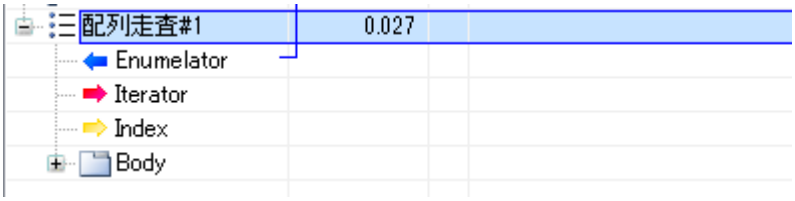
4.1 配列走査について

「配列走査」は、2 値ブローブ解析やグレイサーチなど複数個の結果を得られるデータから、個々のデータを使用するときに用います。個数分のループ処理を行います。

「配列走査」は、Function list では「データ操作」、「配列」の下にあります。

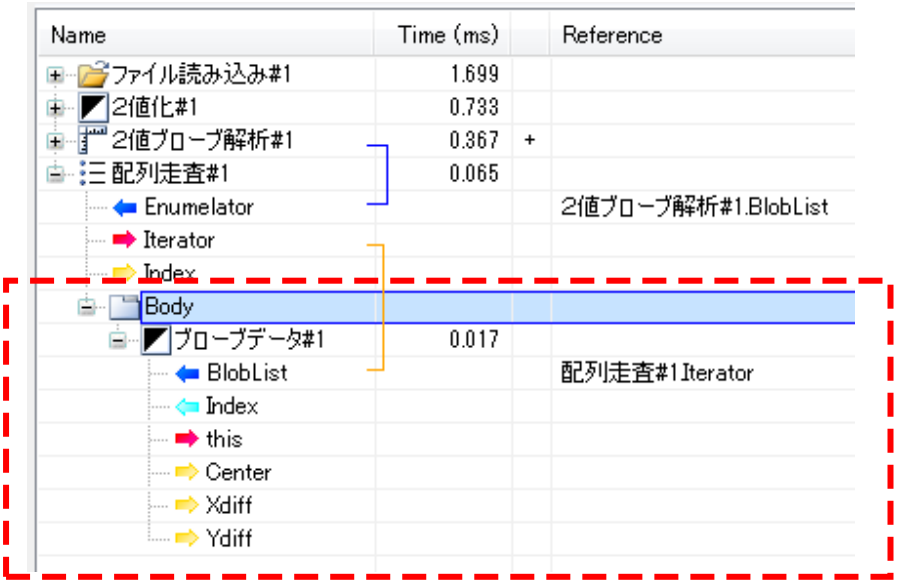


その構造は、以下のようにワークフローに表示されています。



Enumerator に配列走査したいデータをリンクし、Body の中で個々のデータ操作を記入します。

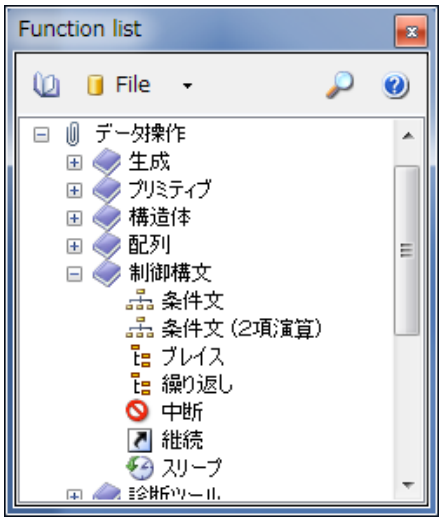
例としては Iterator に配列内のデータが入っており、それを Body の中のブローブデータへ渡して実行しています。



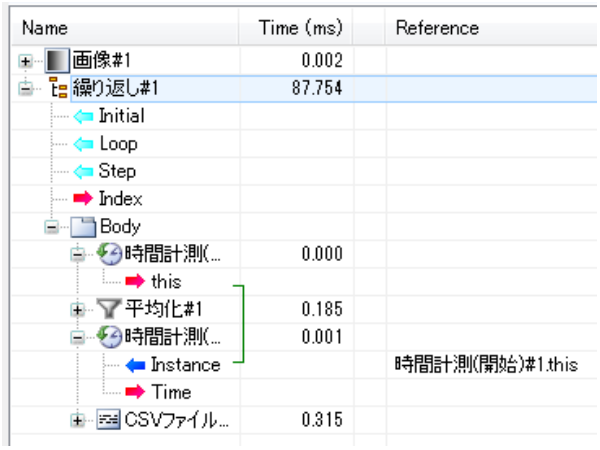
左の点線内の Body 部分をデータ個数分ループ処理します。

4.2 ループ処理について

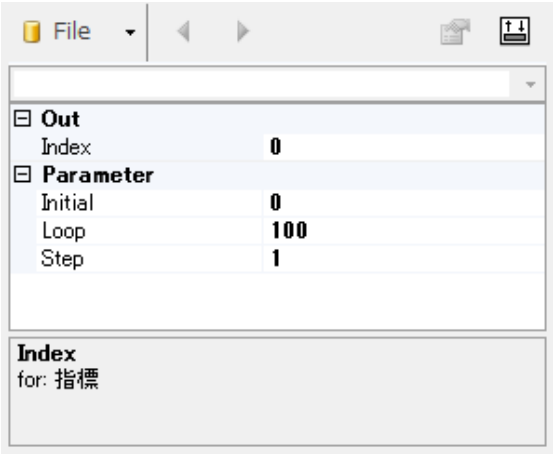
ループ処理を行う場合は、「データ操作」、「制御構文」の下に「繰り返し」を用意しています。
指定した回数分、繰り返し処理を行います。



その構造は、以下のようにワークフローに表示されています。



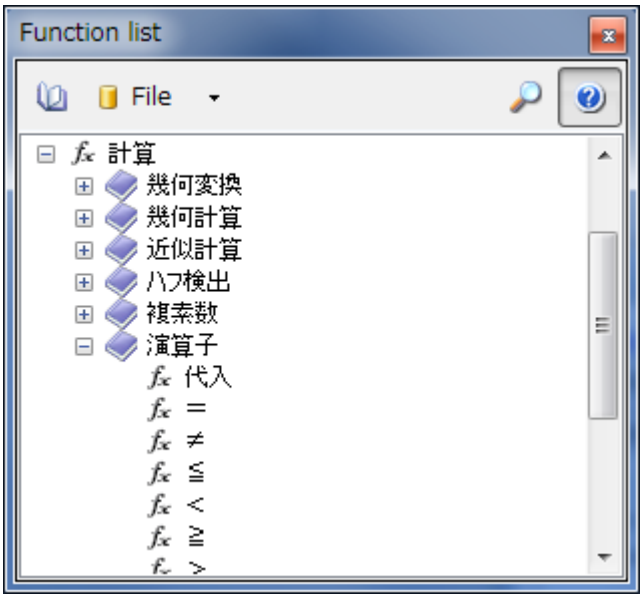
body の部分が繰り返し実行されます。
上記の例では「フィルタ」の「平均化」を実行し、処理時間を CSV ファイルに書き込みます。



「Parameter」の「Index」を「Initial」の値で初期化し、1 回の実行ごとに「Step」の数だけ「Index」が加算されます。「Index」が「Loop」で設定した値に達するまで、ループ処理を行います。

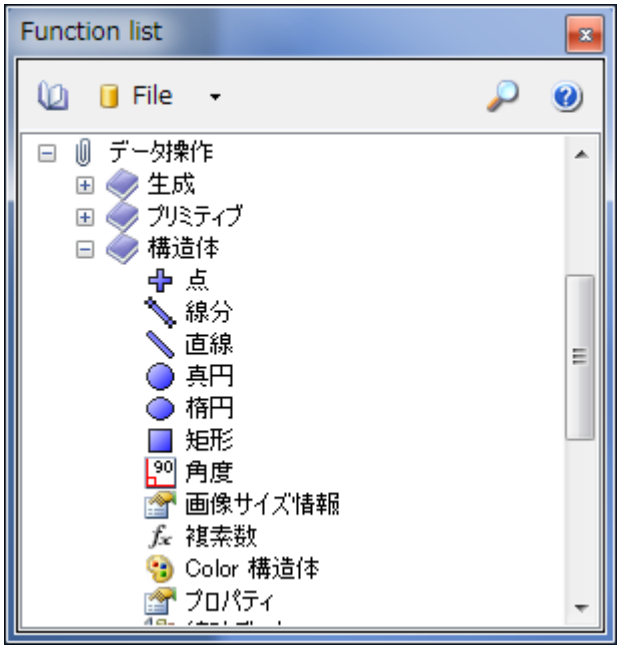
4.3 条件分岐について

条件分岐を行う場合は、「計算」の「演算子」を選択し、その結果から処理を分岐することができます。
「演算子」は、Function list では「計算」、「演算子」の下に各種用意しています。

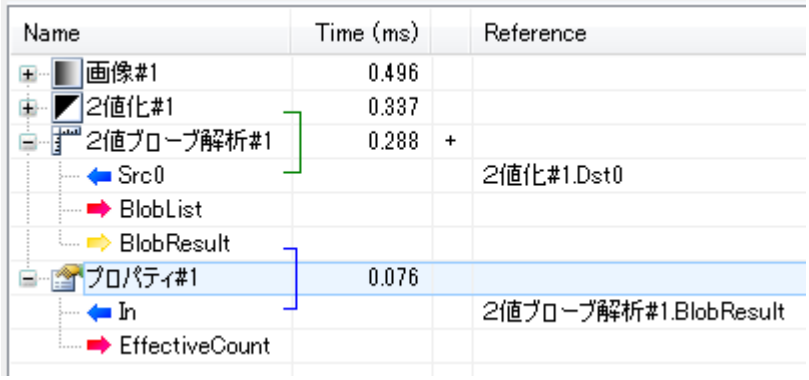


条件分岐は、複雑なので、2 値ブローブ解析の結果による分岐を例にとり説明いたします。
なお、画像 2 値ブローブ解析までの説明は省略します。

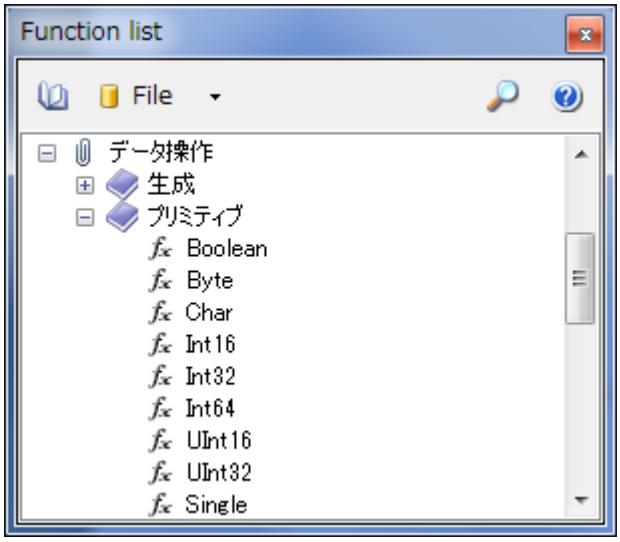
① 「2 値ブローブ解析」の結果から有効ブローブ数(EffectiveCount)を抽出するために「プロパティ」を追加します。
Function list の「データ操作」、「構造体」の順にリストを展開し、「プロパティ」をダブルクリックします。



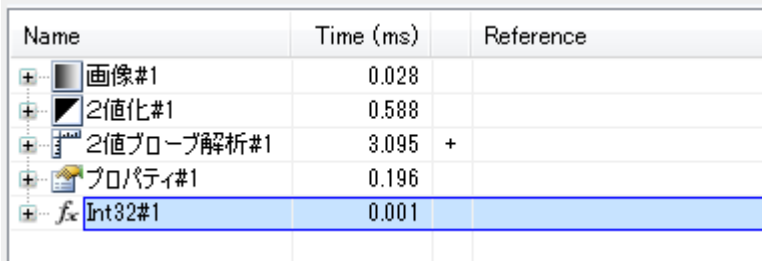
② TASK タブのワークフローリストの最後の行に「プロパティ」が追加されます。



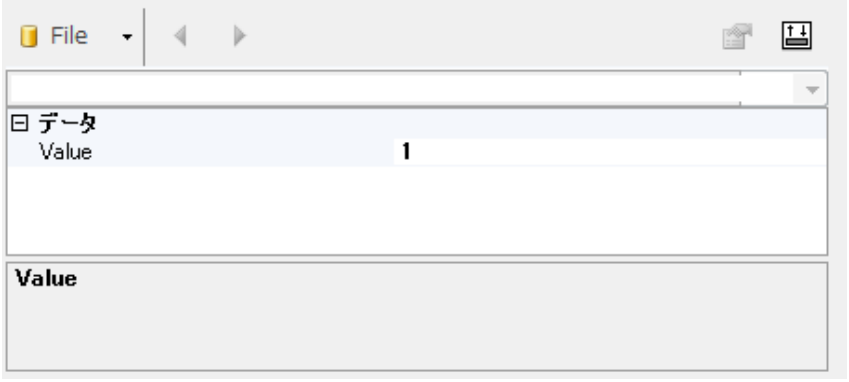
③ 次に条件分岐の対象となる値を設定するために「Int32」を追加します。
Function list の「データ操作」、「プリミティブ」の順にリストを展開し、「Int32」をダブルクリックします。



④ TASK タブのワークフローリストの最後の行に「Int32#1」が追加されます。

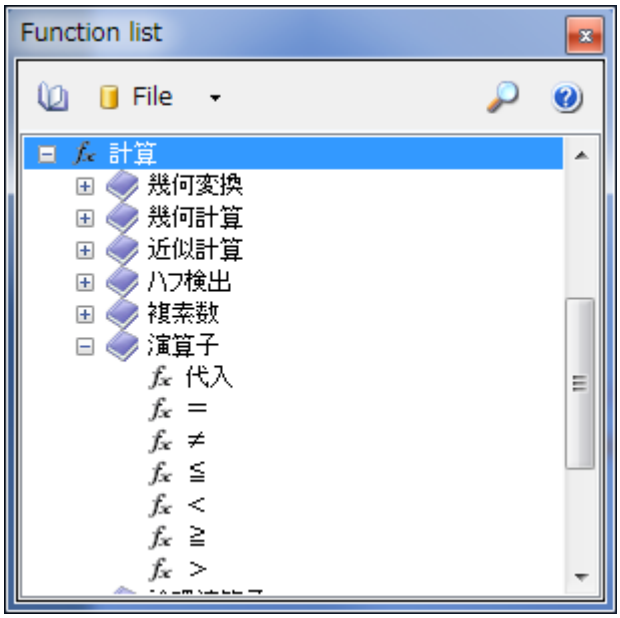


⑤ 「Int32#1」をクリックし、TASK タブの下側にプロパティグリッドが表示されていますので、Parameter の「Value」を「1」に変更します。

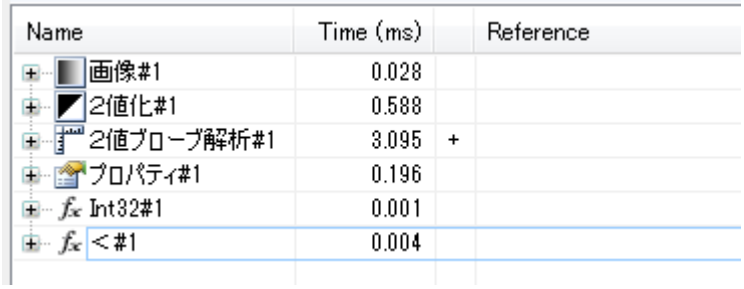


⑥ 「2 値ブローブ解析」の結果から有効ブローブ数(EffectiveCount)と比較する値が設定できたので、条件を追加します。

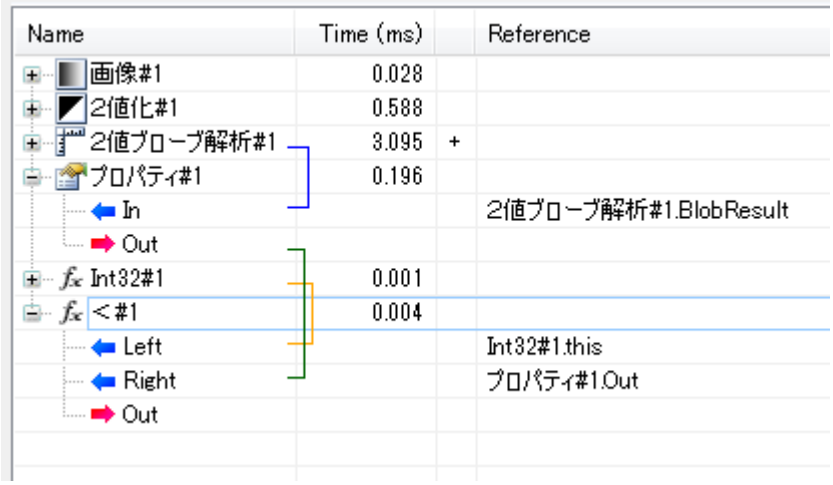
Function list の「計算」、「演算子」の順にリストを展開し、「fx ＜」をダブルクリックします。



⑦ TASK タブのワークフローリストの最後の行に「fx ＜#1」が追加されます。

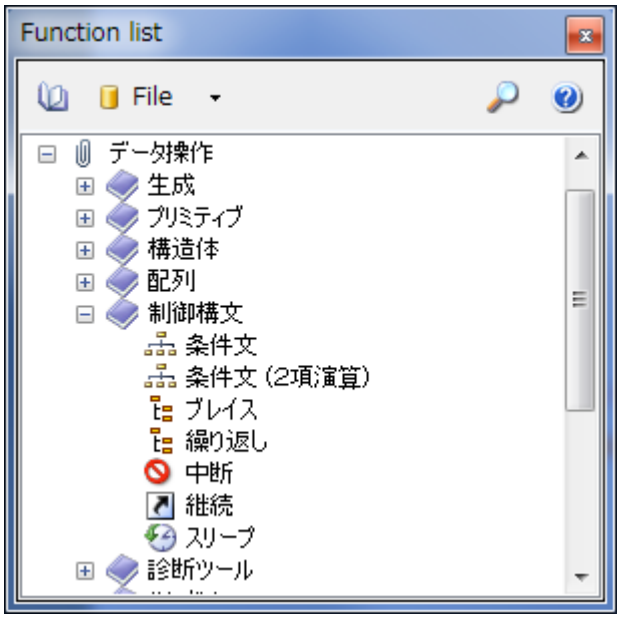


リンクは以下のようになっています。



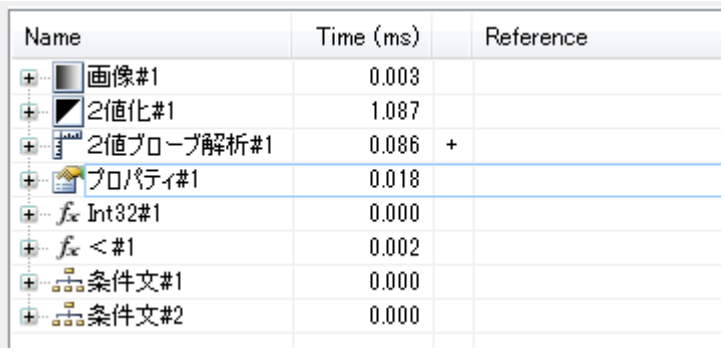
⑧ 演算子の結果から条件文を追加します。

Function list の「データ操作」、「制御構文」の順にリストを展開し、「条件文」をダブルクリックします。



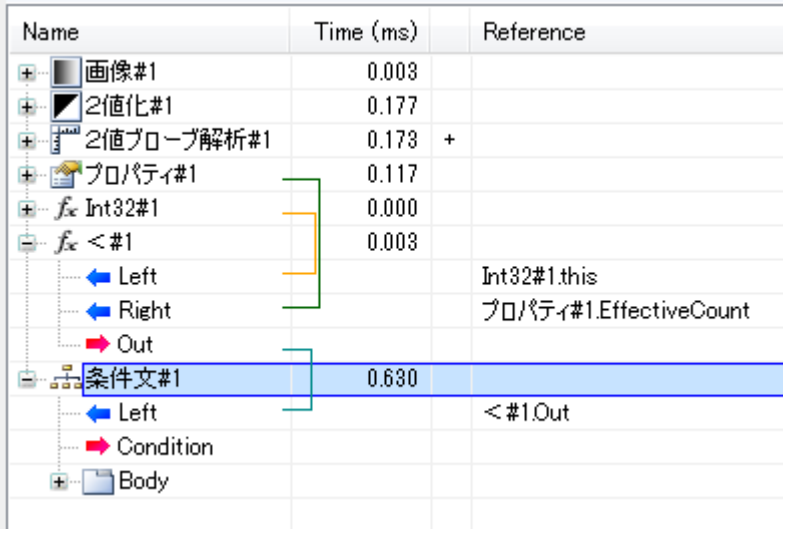
⑨ TASK タブのワークフローリストの最後の行に「条件文#1」が追加されます。

さらにもう一つ「条件文」を追加します。

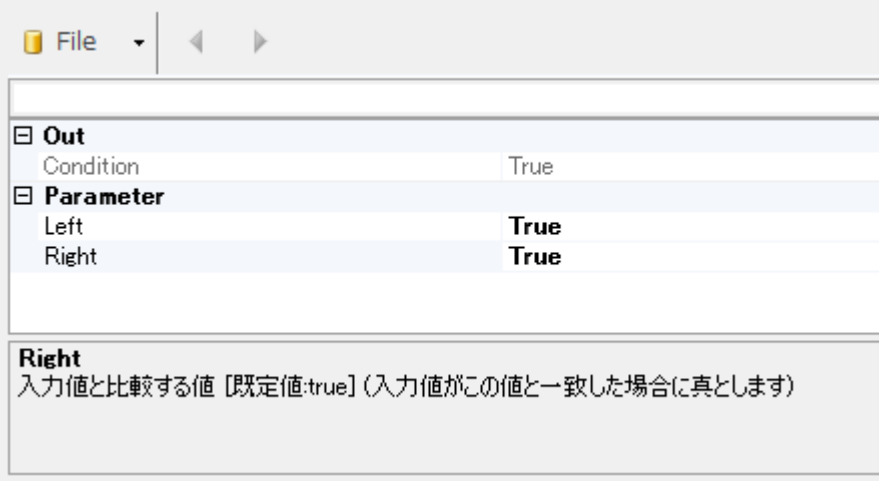


⑩ 条件が真なら、body の部分を実行することが、可能です。

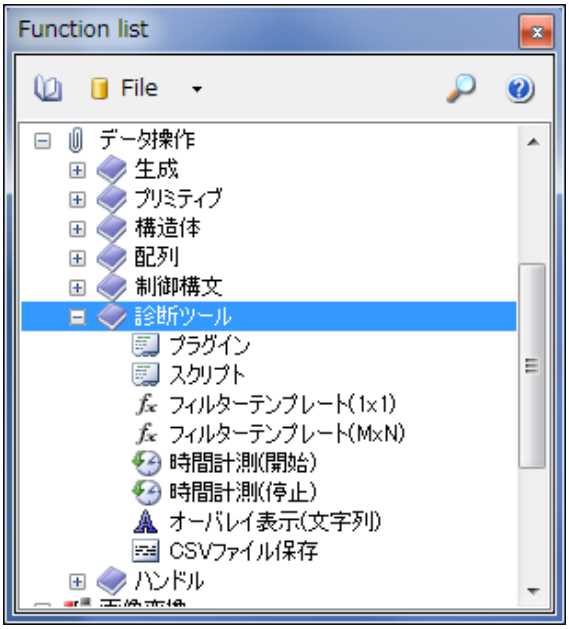
もちろん偽の時に、body の部分を実行することも可能です。



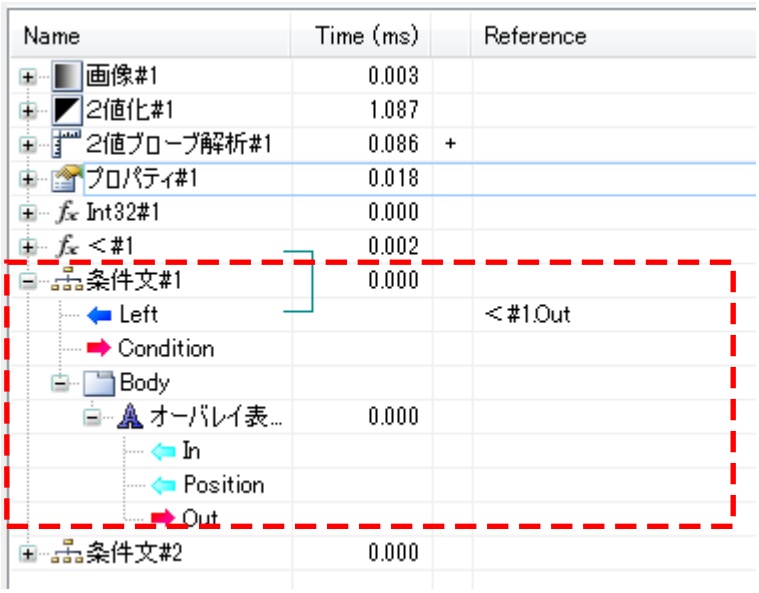
⑪ 「条件文」をクリックし、TASK タブの下側にプロパティグリッドが表示されていますので、Parameter の Left、Right を True から False にすることにより、真から偽へ変更する事ができます。



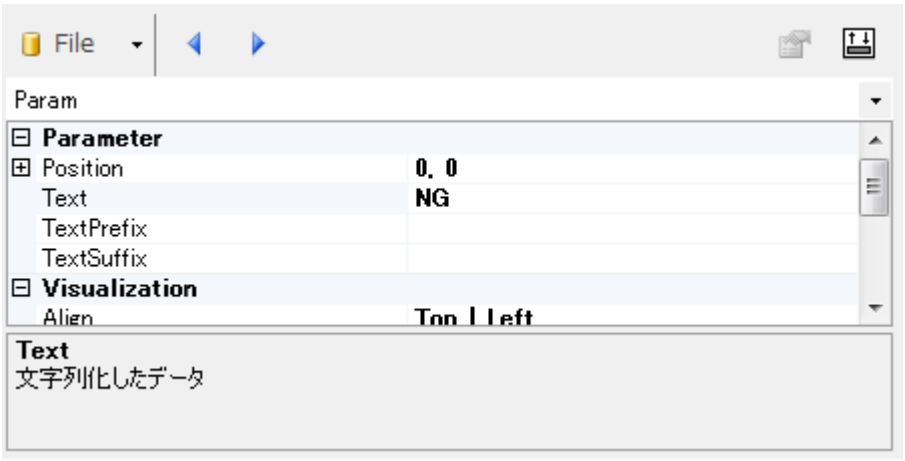
⑫ Body 部分に文字列を表示する機能を追加します。
Function list の「データ操作」、「診断ツール」の順にリストを展開し、「オーバレイ表示(文字列)」をダブルクリックします。
さらにもう一つ「オーバレイ表示(文字列)」を追加します。



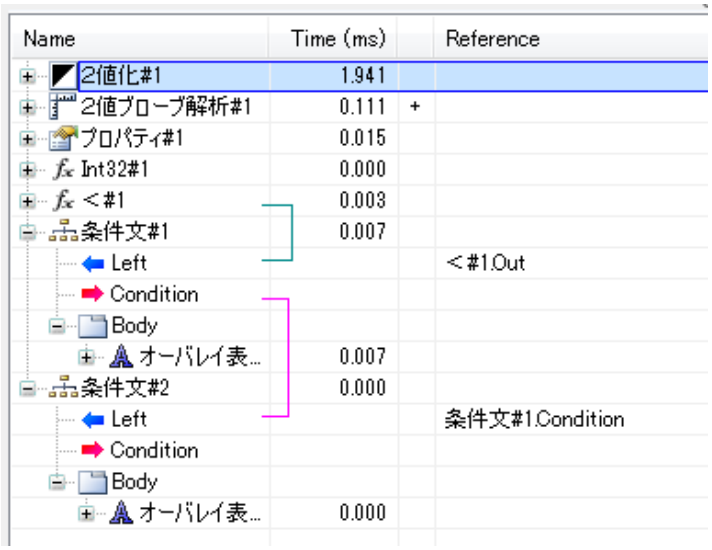
⑬ TASK タブのワークフローリストの最後の行に「オーバレイ表示(文字列)」が追加されます。
最後の行から「条件文#1」の body へドラッグします。
もう一つの「オーバレイ表示(文字列)」を最後の行から「条件文#2」の body へドラッグします。



⑭ 「条件文#1」の「オーバレイ表示(文字列)」をクリックし、TASK タブの下側にプロパティグリッドが表示されていますので、Text に表示したい文字列を入力します。ここでは「NG」を入力します。
同様に「条件文#2」の「オーバレイ表示(文字列)」には「OK」を入力します。

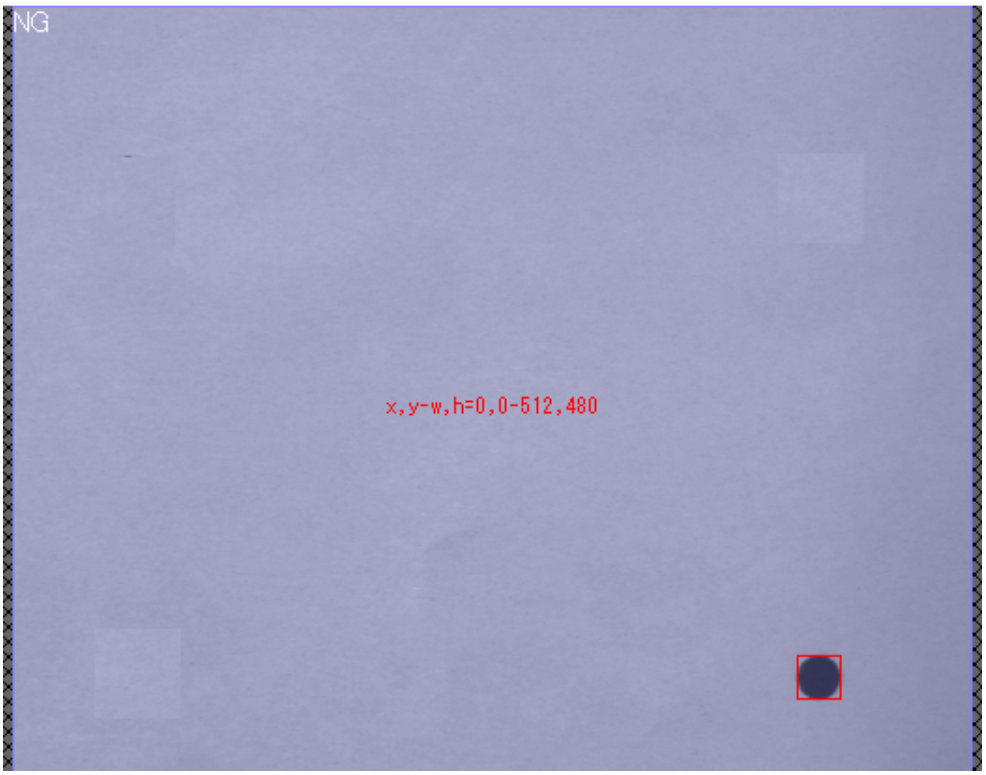


⑮ ワークフローは以下のようになります。



⑯ 実行アイコンをクリックしてみましょう。

⑰ 画像ビューにブローブが検出された場合「NG」がモニタに表示されます。



5. FIE ライブラリ編

WIL-Builder では、FIE ライブラリを使用した処理も実行できます。

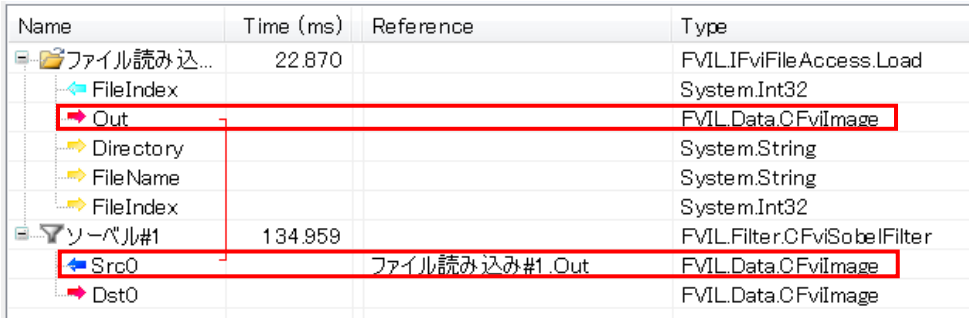
FIE ライブラリにおける画像オブジェクトのデータ入出力の扱いは、WIL ライブラリでの扱いとは異なります。

本章では FIE ライブラリを実行する際の画像オブジェクトの扱いについて説明します。

5.1 FIE ライブラリでの画像オブジェクトの扱い

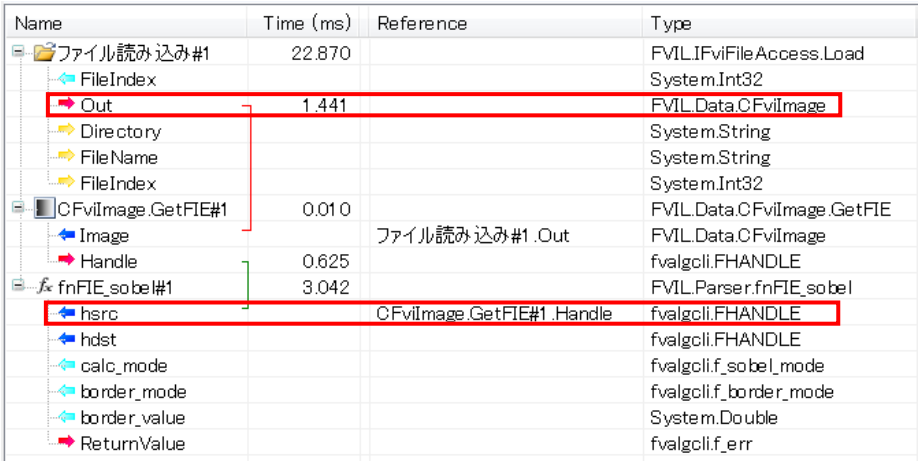
前章までで解説した WIL ライブラリでは、画像データを画像処理ライブラリの「Src」に直接リンクすることで、画像データを入力することができました。

以下の例は「ソーベルフィルタ」を実行する例ですが、「ソーベル#1」のデータ入力の Type は「CfviImage」であり、「ファイル読み込み#1」の「Out」と同じ Type である為、直接リンクする事ができます。



それに対して、FIE ライブラリではデータの入力 Type が「FHANDLE」であるため、画像データ「CfviImage」を直接リンクすることができません。入力する画像データのハンドル「FHANDLE」を取得し「hsrc」や「hdst」にリンクする必要があります。

以下の例は「fnFIE_sobel」を実行する例ですが、画像データ「CfviImage」の「FHANDLE」を取得し、「fnFIE_sobel」の「hsrc」にリンクしています。



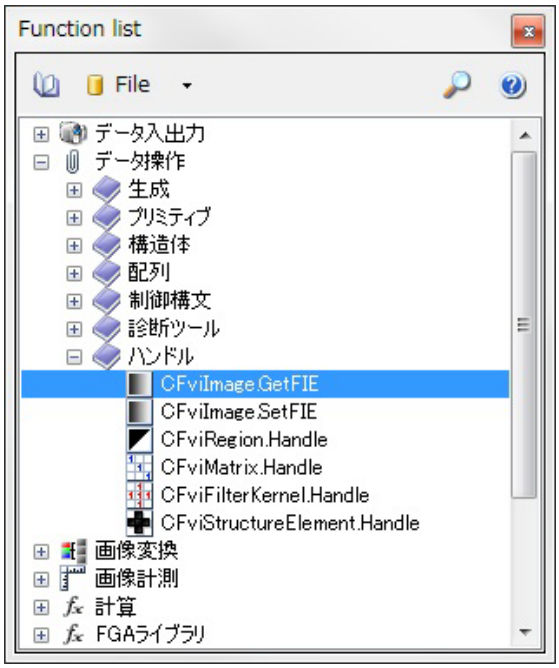
また、FIE ライブラリにおいて、フィルタなどの結果画像が出力される処理においては、出力画像用の画像オブジェクトを新規に生成し、データ出力「hdst」等とリンクする必要があります。

画像オブジェクトを生成する場合、画像サイズ情報が必要となりますので、「ファイル読み込み」にて入力した画像のプロパティを使用して画像サイズ情報を設定し、画像オブジェクトを生成します。

5.1.1 FIE ハンドルの取得

FIE ライブラリのデータ入出力の Type は FIE ハンドルですので、画像オブジェクトから「CfviImage.GetFIE」を使用して取得した FIE ハンドルをリンクする必要があります。

① Function list の「データ操作」、「ハンドル」の順にリストを展開し、「CfviImage.GetFIE」をダブルクリックします。



② TASK タブのワークフローリストの最後の行に「CfviImage.GetFIE」が追加されますので「ファイル読み込み」と「CfviImage.GetFIE#1」をリンクします。

Name	Time (...	Reference	Type
ファイル読み込み#1	104.924		FVIL.IFviFileAccess.Load
CFviImage.GetFIE#1	1.197		FVIL.Data.CFviImage.GetFIE
Image		ファイル読み込み#1.Out	FVIL.Data.CFviImage
Handle	0.806		fvalgcli.FHANDLE

ファイルから読み込んだ画像オブジェクトの FIE ハンドルが取得できました。
また、以下のように「画像オブジェクト生成」とリンクすることで、生成した画像オブジェクトの FIE ハンドルも取得できます。

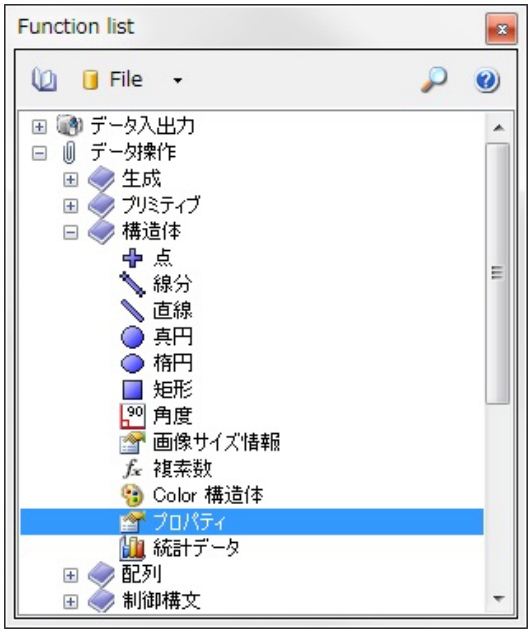
Name	Time (...	Reference	Type
ファイル読み込み#1	104.924		FVIL.IFviFileAccess.Load
プロパティ#1	0.024		HorzSize
プロパティ#2	0.047		VertSize
プロパティ#3	0.062		ImageType
プロパティ#4	0.025		Channel
プロパティ#5	0.034		Depth
プロパティ#6	0.020		ImageInfo
画像サイズ情報#1	1.304		FVIL.ImageSize
画像オブジェクト生成#1	0.276		FVIL.Data.CFviImage
ImageSize		画像サイズ情報#1.this	FVIL.ImageSize
this			FVIL.Data.CFviImage
CFviImage.GetFIE#1	0.017		FVIL.Data.CFviImage.GetFIE
Image		画像オブジェクト生成#1.this	FVIL.Data.CFviImage
Handle	0.005		fvalgcli.FHANDLE

5.1.2 画像オブジェクトの生成

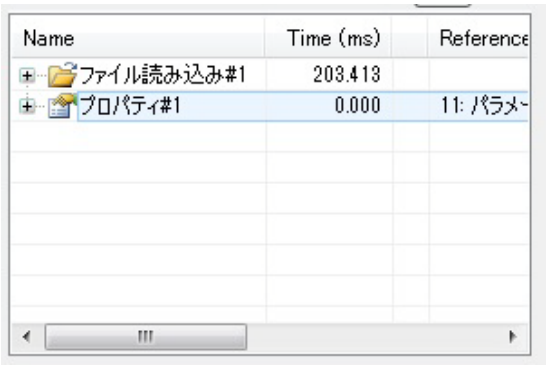
FIE ライブラリのフィルタなど、結果画像が出力される処理においては、出力画像用の画像オブジェクトを新規に生成する必要があります。画像オブジェクトを生成するには、画像サイズ情報が必要になります。

本章では、入力画像の画像サイズ情報を用いて出力画像用の画像オブジェクトの生成手順を解説します。

① 入力画像の画像サイズ情報を取得するため「プロパティ」を用います。Function list の「データ操作」、「構造体」の順にリストを展開し、「プロパティ」をダブルクリックします。

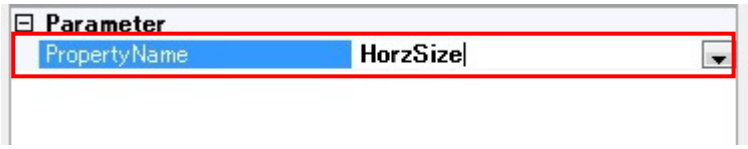


② TASK タブのワークフローリストの最後の行に「プロパティ#1」が追加されます。

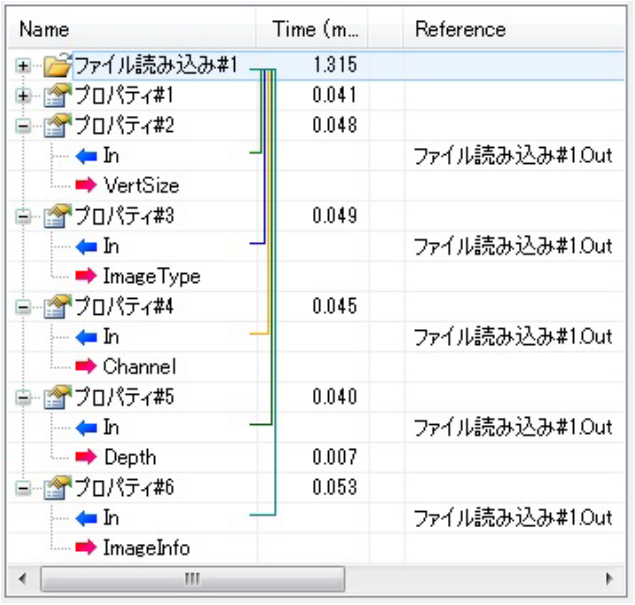


③ 「プロパティ#1」をクリックすると、TASK タブの下側にプロパティグリッドが表示されていますので、「PropertyName」をクリックし、取得したいパラメータを選択します。

ここでは、Parameter の「PropertyName」を「HorzSize」に設定します。

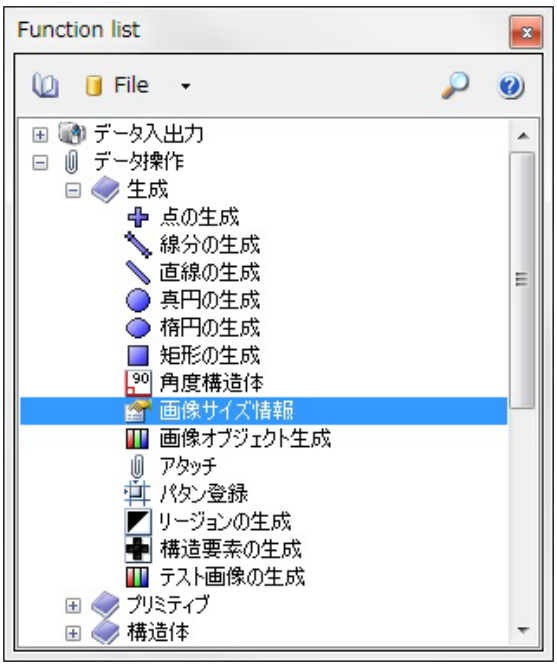


④ 「HorzSize」同様、画像サイズ情報のパラメータである「VertSize」「ImageType」「Channel」「Depth」「ImageInfo」を設定する為に、「プロパティ」を 5 個追加します。
「ファイル読み込み」と各「プロパティ」のリンクを修正します。
そして、各「プロパティ」の「PropertyName」を「VertSize」「ImageType」「Channel」「Depth」「ImageInfo」に設定します。

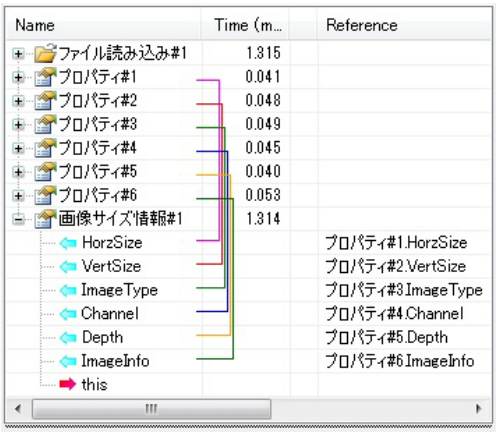


これで、画像オブジェクトを生成するための画像サイズ情報を、入力画像から取得することができました。

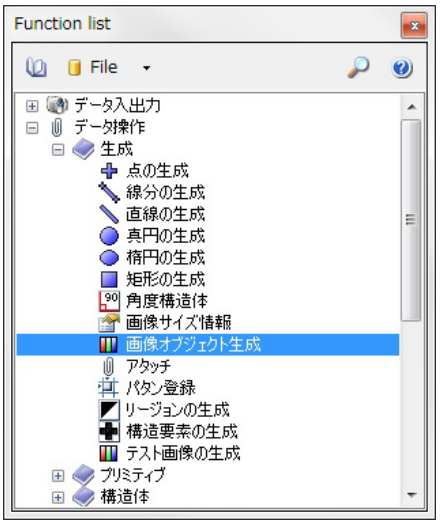
⑤ 入力画像から得られた「画像サイズ情報」を、生成する画像オブジェクトのパラメータとして利用します。Function list の「データ操作」、「生成」の順にリストを展開し、「画像サイズ情報」をダブルクリックします。



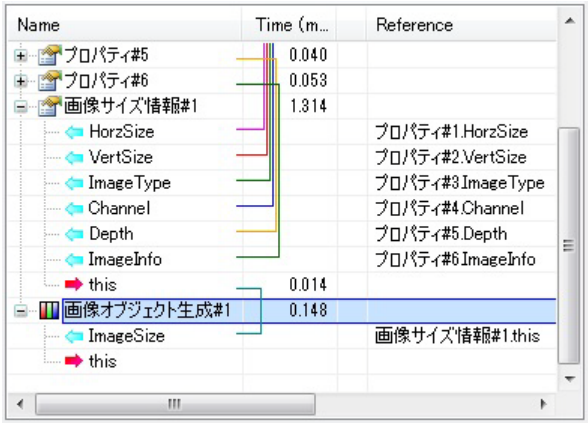
⑥ TASK タブのワークフローリストの最後の行に「画像サイズ情報」が追加されます。
上記で取得したパラメータと「画像サイズ情報」の各パラメータのリンクを修正します。



⑦ 「画像サイズ情報」の設定ができましたので、「画像オブジェクト生成」を追加します。
Function list の「データ操作」、「生成」の順にリストを展開し、「画像オブジェクト生成」をダブルクリックします。



⑧ TASK タブのワークフローリストの最後の行に「画像オブジェクト生成」が追加されます。

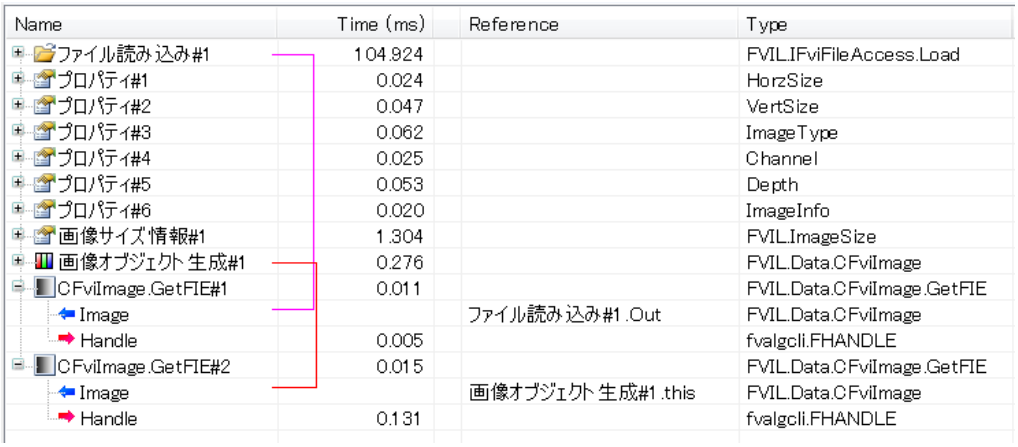


これで画像オブジェクトが生成されました。

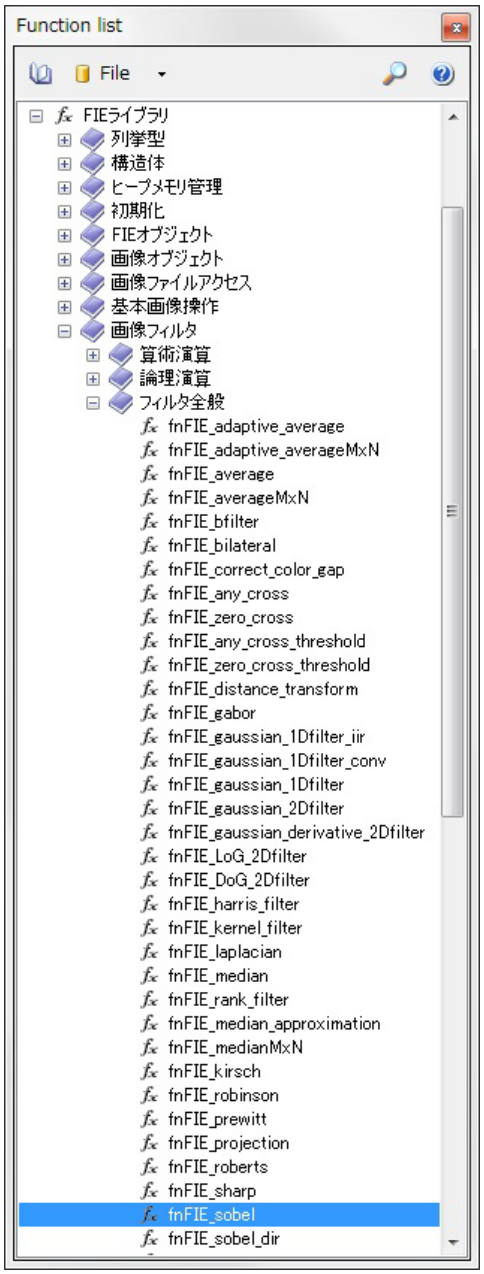
5.2 ソーベルフィルタの実行

前章にて FIE ハンドルの取得方法および画像オブジェクトの生成について解説したので、実際に FIE ライブラリのソーベルフィルタを実行してみましょう。「ファイル読み込み」にて入力した画像に対し、ソーベルフィルタを実行する例で解説します。

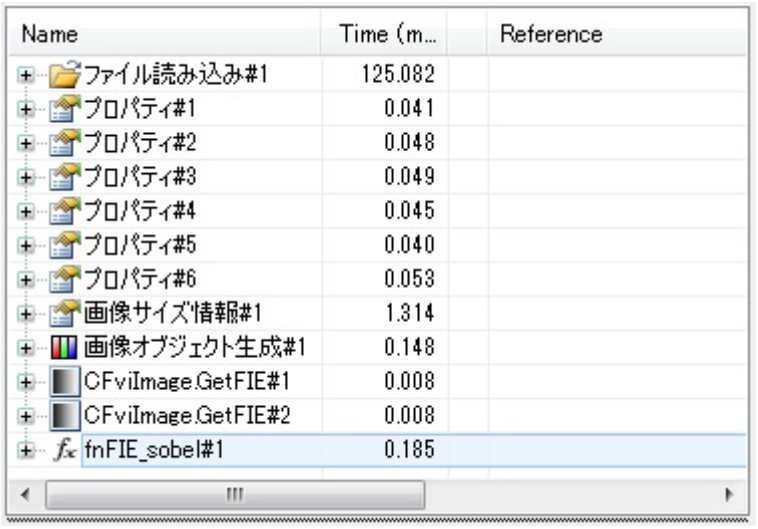
① ソーベルフィルタの実行に先立ち、「ファイル読み込み」、出力用の「画像オブジェクトの生成」および「FIE ハンドルの取得」を設定します。ソーベルフィルタでは入力画像と出力画像の 2 つの画像オブジェクトが必要になりますので、「CFviImage.GetFIE」を 2 つ追加します。「ファイル読み込み」と「CfviImage.GetFIE#1」をリンクし、「画像オブジェクト生成」と「CfviImage.GetFIE#2」をリンクします。



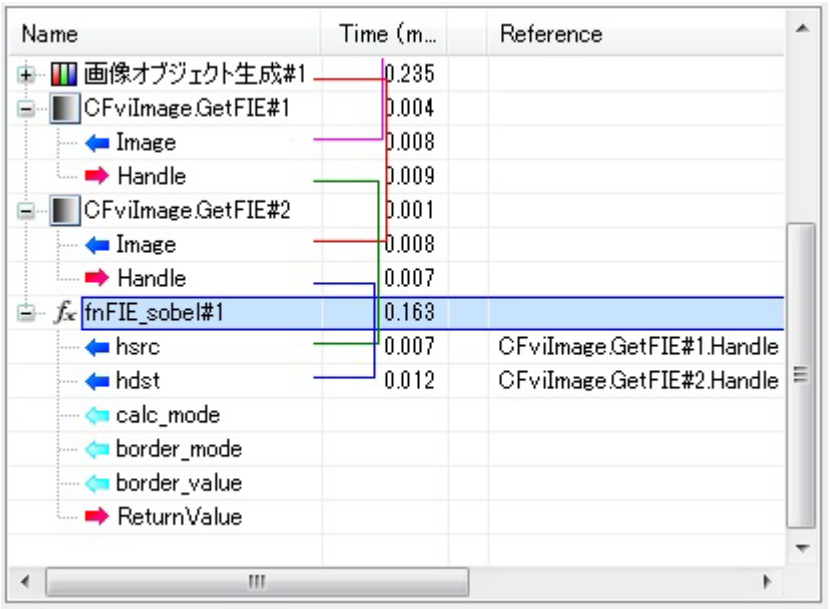
② 続いてソーベルフィルタを設定します。Function list の「FIE ライブラリ」、「画像フィルタ」、「フィルタ全般」の順にリストを展開し、「fnFIE_sobel」をダブルクリックします。



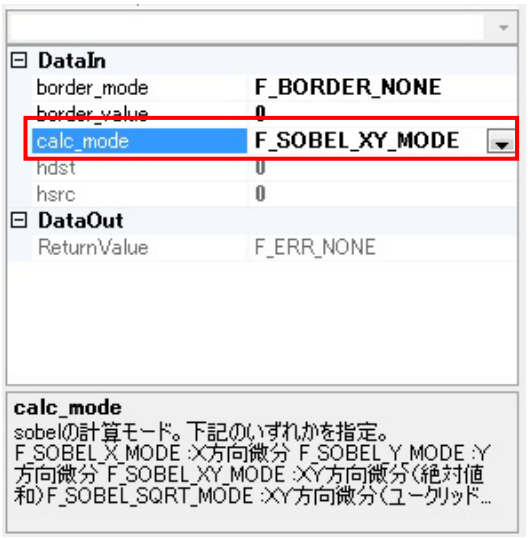
③ TASK タブのワークフローリストの最後の行に「fnFIE_sobel#1」が追加されます。



④ 「CfviImage.GetFIE#1」と「hsrc」を、「CfviImage.GetFIE#2」と「hdst」をリンクします。

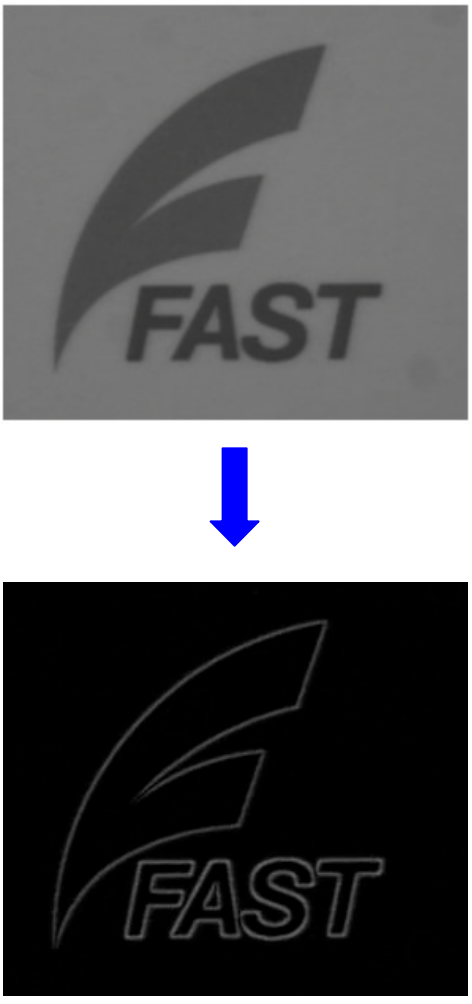


⑤ 「fnFIE_sobel」をクリックすると、TASK タブの下側にプロパティグリッドが表示されていますので、「calc_mode」をクリックし、ソーベルフィルタの計算モードを選択して下さい。
ここでは、上記 2.2 項目の計算モードと合わせる為、「F_SOBEL_XY_MODE」を選択します。



⑥ 実行アイコンをクリックしてみましょう。

ソーベルフィルタが行われた画像が表示されます。



5.3 ベイヤー色合成を設定してみよう

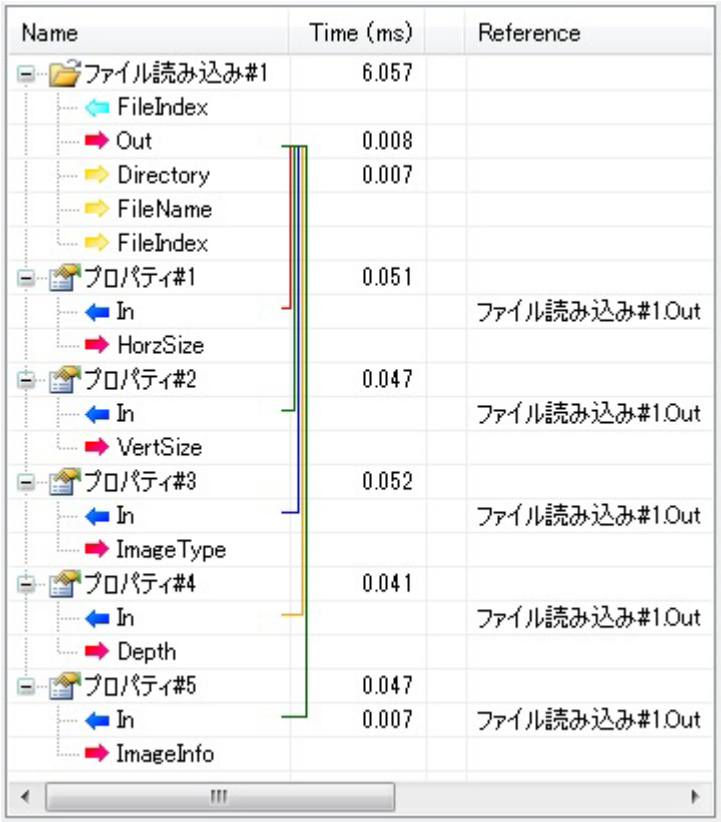
FIE ライブラリにあるベイヤー色合成をしてみましょう。

「ファイル読み込み」、「画像オブジェクトの生成」および「FIE ハンドルの取得」に続いてベイヤー色合成を設定します。

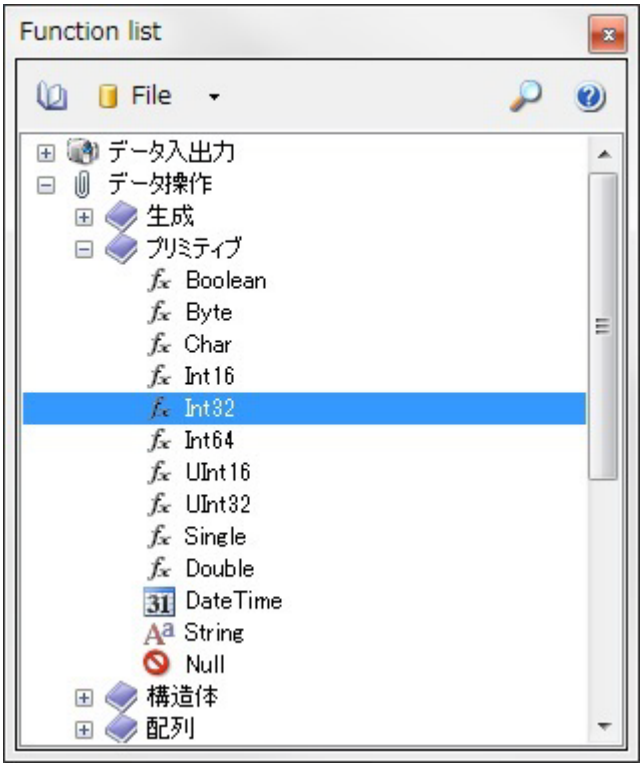
5.3.1 出力画像用の画像オブジェクト(カラー)の生成

ベイヤー色合成を実行する場合もソーベルフィルタ同様に、画像オブジェクトの生成が必要です。ただし、ベイヤー色合成は、濃淡画像(ベイヤー画像)をカラー画像へ変換しますので、カラー画像用の画像オブジェクトが必要になります。その際、カラー画像用に生成する画像オブジェクトのチャネル数は「3」である必要があります。

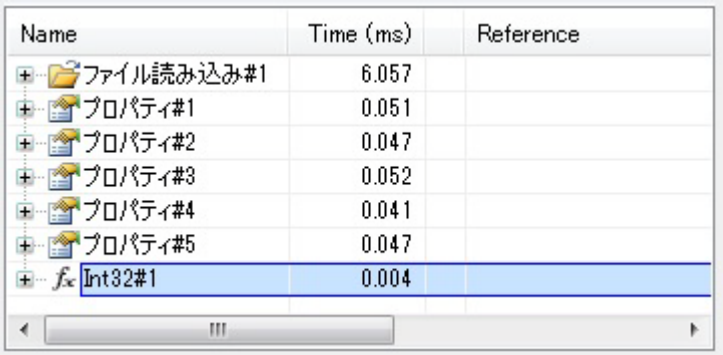
① ソーベルフィルタの手順と同様に「プロパティ」の追加、「ファイル読み込み」と各「プロパティ」のリンクの修正、および各「プロパティ」の「PropertyName」を「HorzSize」「VortSize」「ImageType」「Depth」「ImageInfo」にそれぞれ設定します。



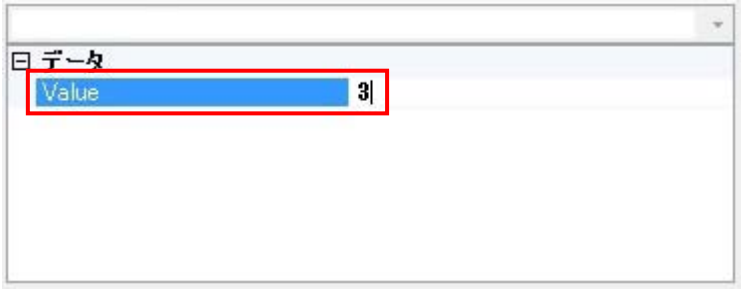
② カラー画像用にチャネル数「Channel」を設定する為に、「Int32」を追加します。
Function list の「データ操作」、「プリミティブ」の順にリストを展開し、「Int32」をダブルクリックします。



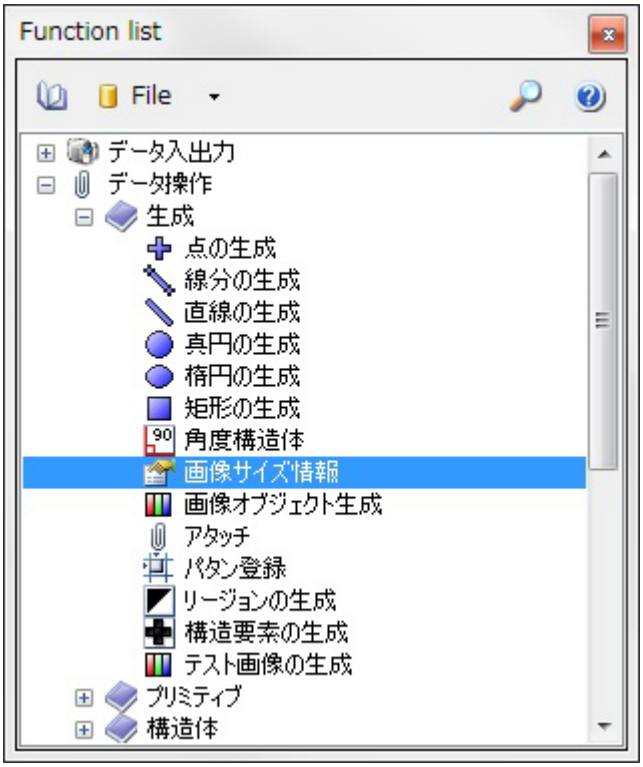
③ TASK タブのワークフローリストの最後の行に「Int32」が追加されます。



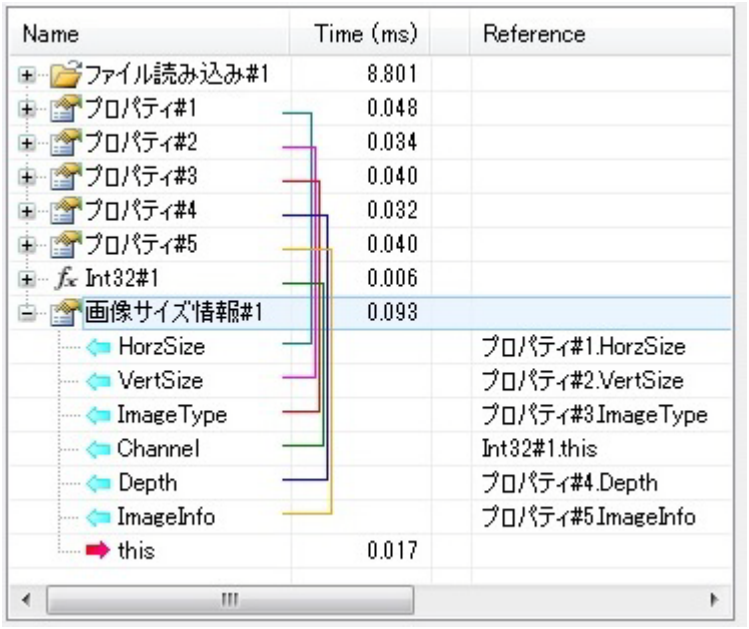
④ ベイヤー色合成における出力画像のチャネル数は、「3」に設定する必要がありますので、
Parameter の「Value」を「3」に変更します。



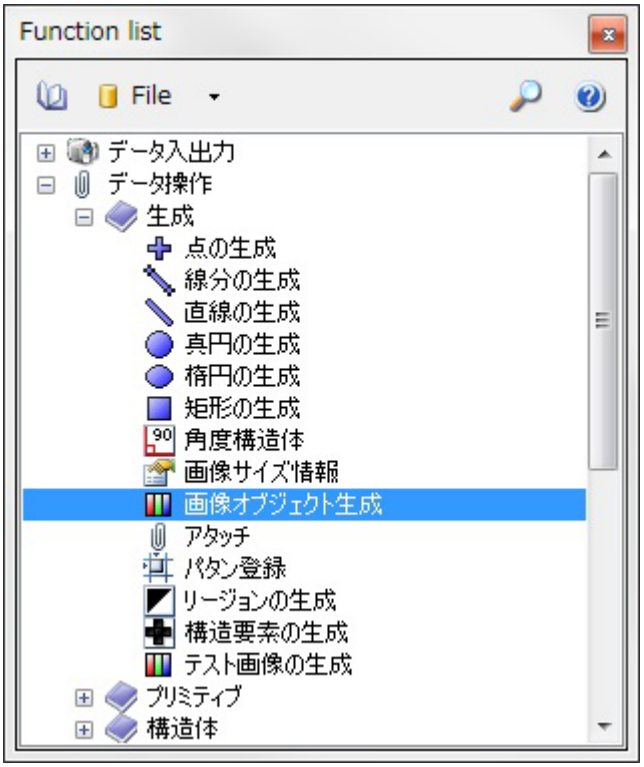
⑤ 「画像サイズ情報」のパラメータの用意ができましたので、「画像サイズ情報」を追加します。
Function list の「データ操作」、「生成」の順にリストを展開し、「画像サイズ情報」を
ダブルクリックします。



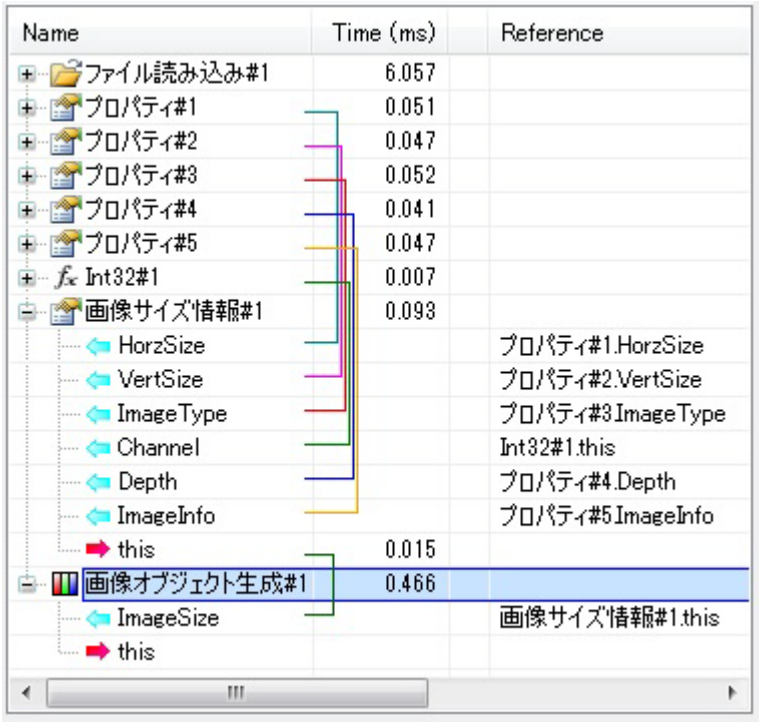
⑥ TASK タブのワークフローリストの最後の行に「画像サイズ情報」が追加されます。
上記で取得したパラメータと「画像サイズ情報」の各パラメータのリンクを修正します。



⑦ 「画像サイズ情報」の設定ができましたので、「画像オブジェクト生成」を追加します。
Function list の「データ操作」、「生成」の順にリストを展開し、「画像オブジェクト生成」をダブルクリックします。

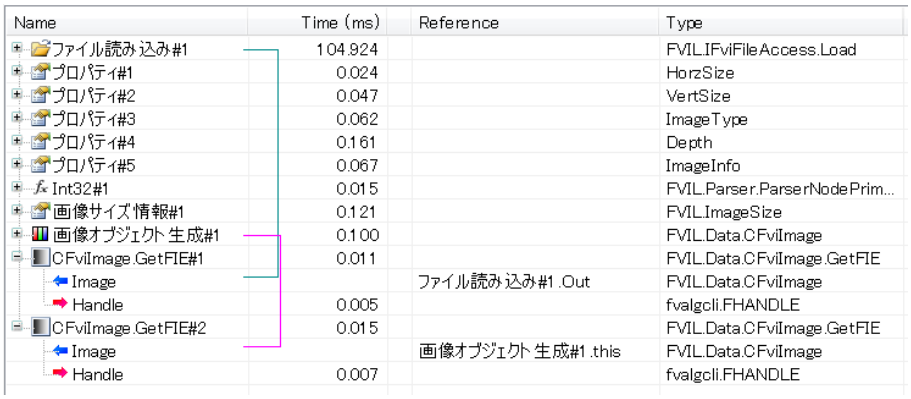


⑧ TASK タブのワークフローリストの最後の行に「画像オブジェクト生成」が追加されます。

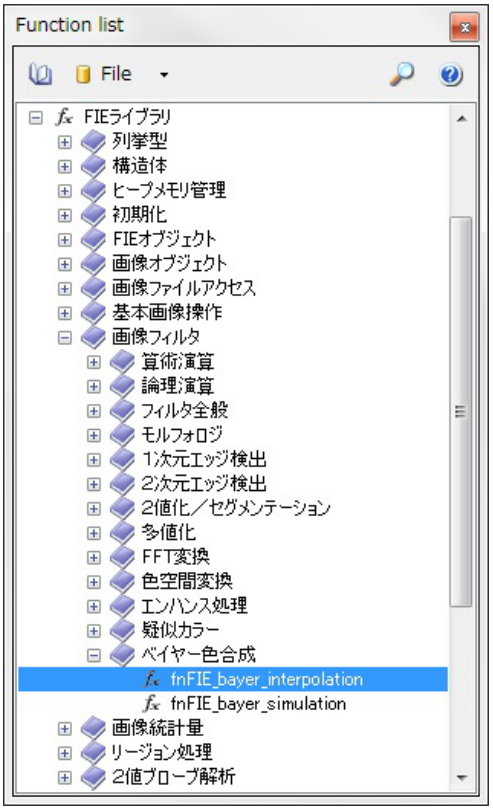


5.3.2 ベイヤー色合成の実行

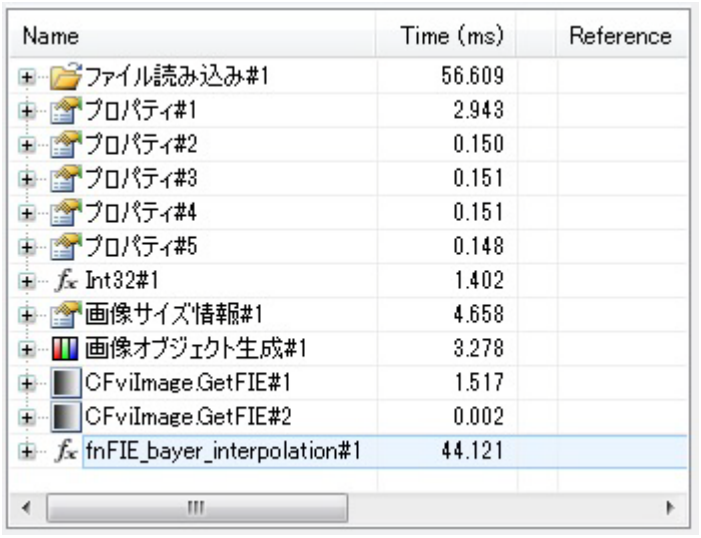
① ベイヤー色合成の実行に先立ち、「FIE ハンドルの取得」を設定します。ベイヤー色合成では入力画像と出力画像の 2 つの画像オブジェクトが必要になりますので、「CFviImage.GetFIE」も 2 つ追加します。「ファイル読み込み」と「CfviImage.GetFIE#1」をリンクし、「画像オブジェクト生成」と「CfviImage.GetFIE#2」をリンクします。



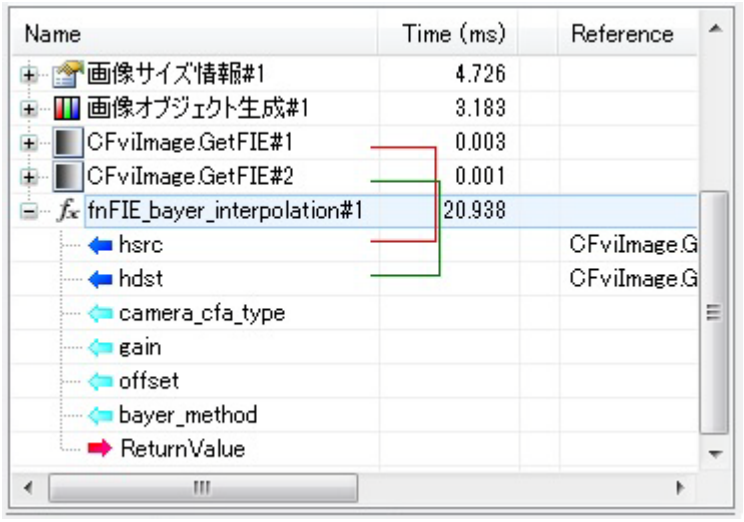
② Function list の「FIE ライブラリ」、「画像フィルタ」、「ベイヤー色合成」の順にリストを展開し、「fnFIE_bayer_interpolation」をダブルクリックします。



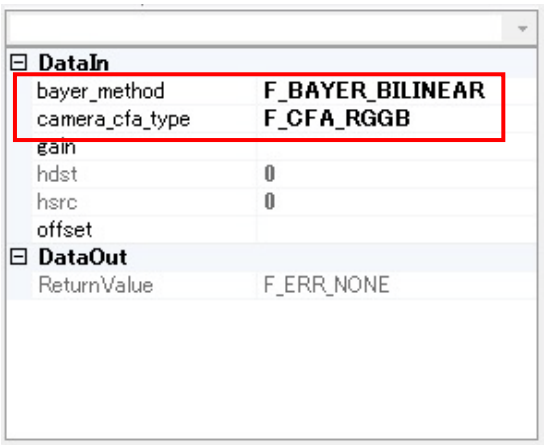
③ TASK タブのワークフローリストの最後の行に「fnFIE_bayer_interpolation」が追加されます。



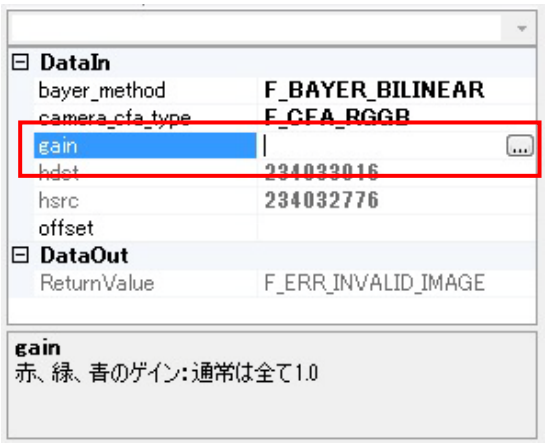
④ 「CfviImage.GetFIE#1」と「hsrc」を、「CfviImage.GetFIE#2」と「hdst」をリンクします。



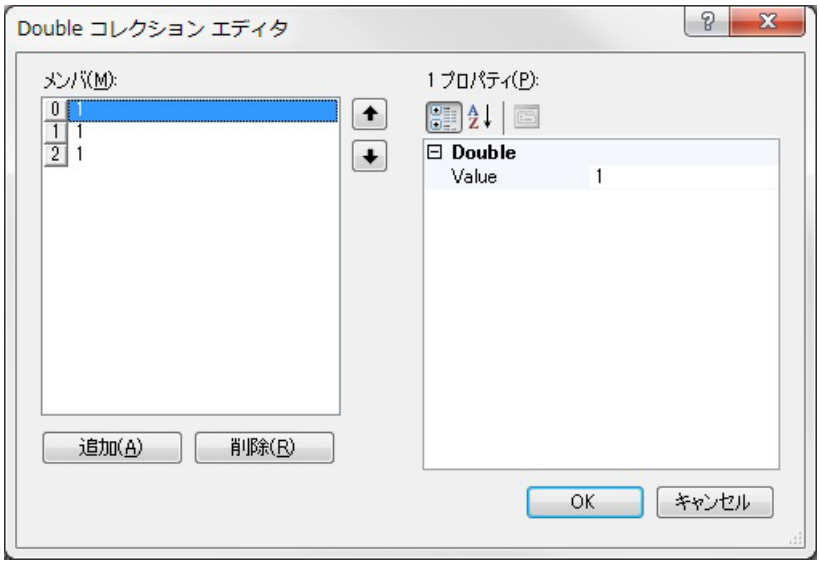
⑤ 「fnFIE_bayer_interpolation」をクリックすると、TASK タブの下側にプロパティグリッドが表示されていますので、「bayer_method」をクリックすると、色合成手法が選択できます。
ここでは、「F_BAYER_BILINEAR」を選択します。
また、「camera_cfa_type」をクリックすると、カメラのカラーフィルタ配列タイプが選択できます。
読み込んだ画像のカラーフィルタ配列タイプに合わせて選択して下さい。



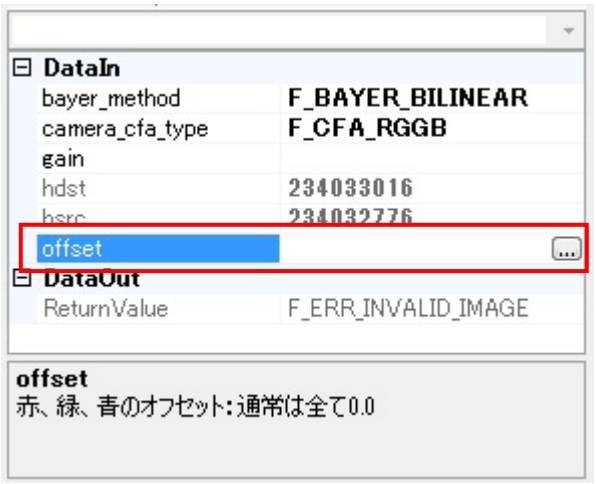
⑥ 「fnFIE_bayer_interpolation」は、「gain」と「offset」を設定する必要があります。
プロパティグリッドの「gain」をクリックし、[...] をクリックして、コレクションエディタを開きます。



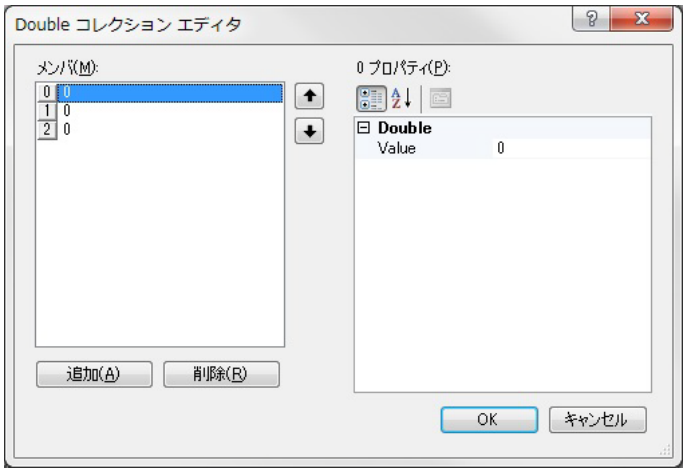
⑦ 「gain」は、赤、緑、青の各色に対応している為、メンバを３個追加します。
また、ゲインの値は通常は全て 1.0 ですので、プロパティの「Value」は全て「1」に変更します。



⑧ プロパティグリッドの「offset」をクリックし、[...] をクリックして、コレクションエディタを開きます。



⑨ 「offset」は、赤、緑、青の各色に対応している為、メンバを 3 個追加します。
また、オフセットの値は通常は全て 0.0 ですので、プロパティの「Value」は全て「0」に変更します。



⑩ 実行アイコンをクリックしてみましょう。

ベイヤー色合成が行われた画像が表示されます。

5.4 プラグイン機能について

プラグイン機能とは、プログラミングでまとまった処理(ParserNodeUnit の派生クラス)を作成し、WIL-Builder に読み込ませて処理することです。
もちろん、このまとまった処理はプログラムで使用することも可能です。

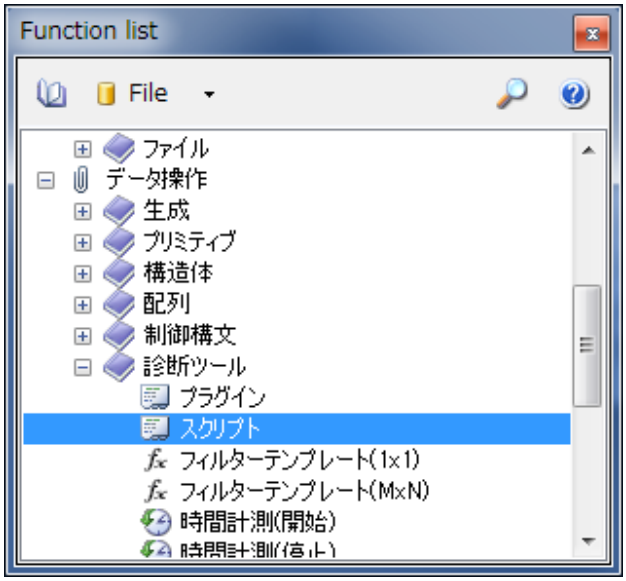
なお、次章で説明いたしますサンプルワークフロー「SAMPLE05」、「SAMPLE11」は、あらかじめプラグインを用意しています。

プラグインの詳細は、「WIL-Builder 説明書」の「5.2.7 Plugin」をご参照ください。

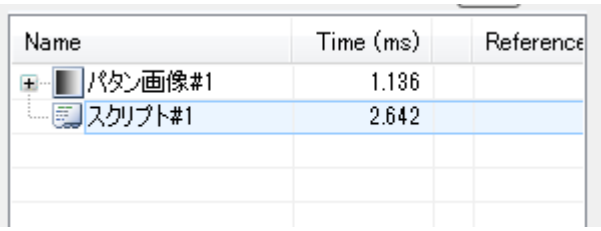
5.5 スクリプト機能について

スクリプト機能とは、プラグイン機能の Execute メソッド内のみ記述する事ができる機能です。
C# または Visual Basic をスクリプト言語として使用する事が出来、独自の処理を簡単に追加する事ができます。
ここでは、登録したパタンの縦、横サイズをコンソールに表示する例を説明いたします。
なお、パタンの登録方法は、説明済みですので省略します。
使用する画像は、「sample00¥sample01.bmp」を使用します。

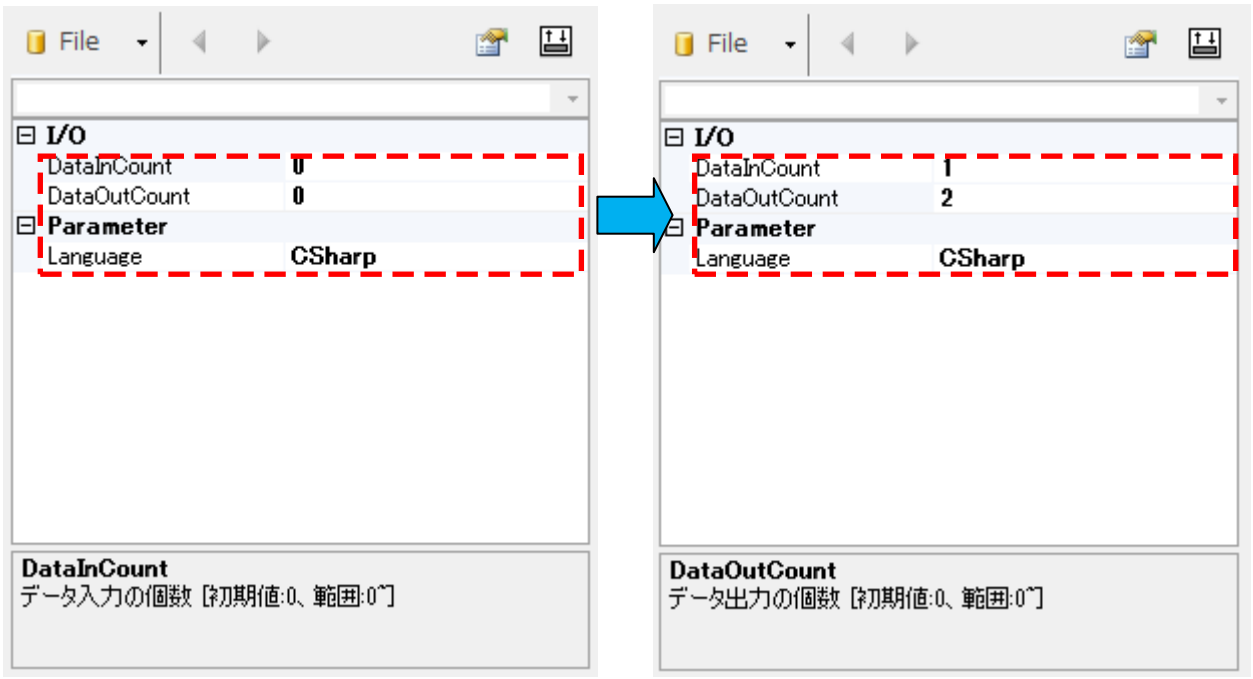
① Function list の「データ操作」、「診断ツール」の順にリストを展開し、「スクリプト」をダブルクリックします。



TASK タブのワークフローリストの最後の行に「スクリプト」が追加されます。

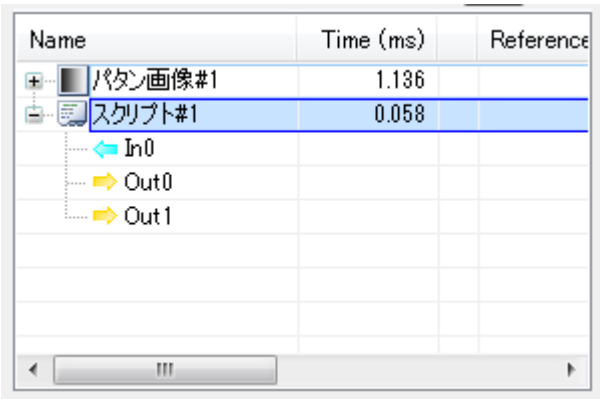


② ワークフローリストの「スクリプト」をクリックすると、TASK タブの下側にプロパティグリッドが表示されていますので、DataInCount を「1」に、DataOutCount を「2」に設定します。

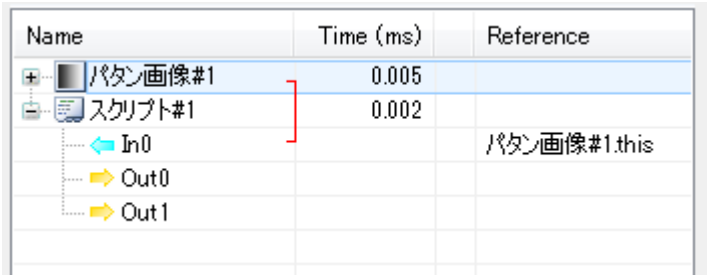


Language に CSharp が設定されている場合は、スクリプトが C# として処理され、VisualBasic が設定されている場合は、スクリプトが Visual Basic として処理されます。

③ 変更後、TASK タブのワークフローリストの最後の行に「スクリプト」をクリックすると下記の様に、3つのデータが追加されます。設定した内容が反映され、In データが「1」、Out データが「2」追加されています。

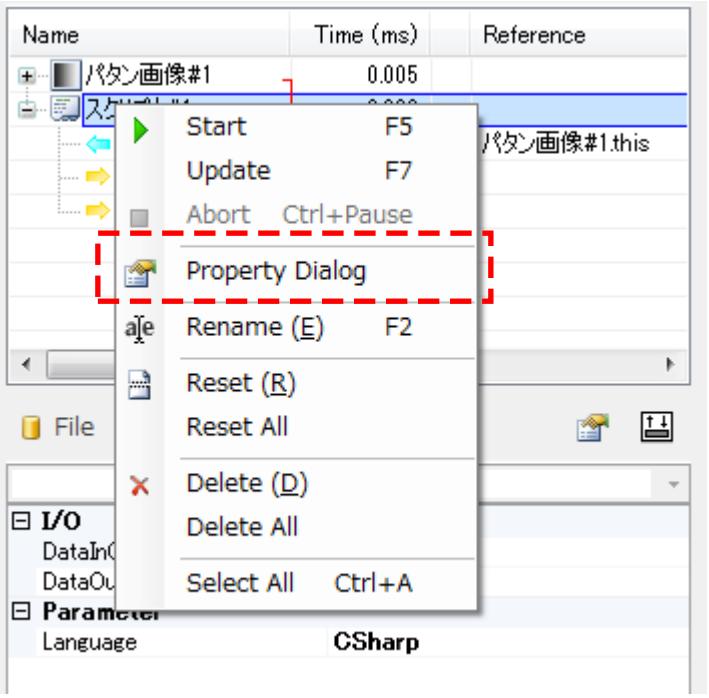


④ パタン画像#1 を In0 にドラッグするとリンクされます。

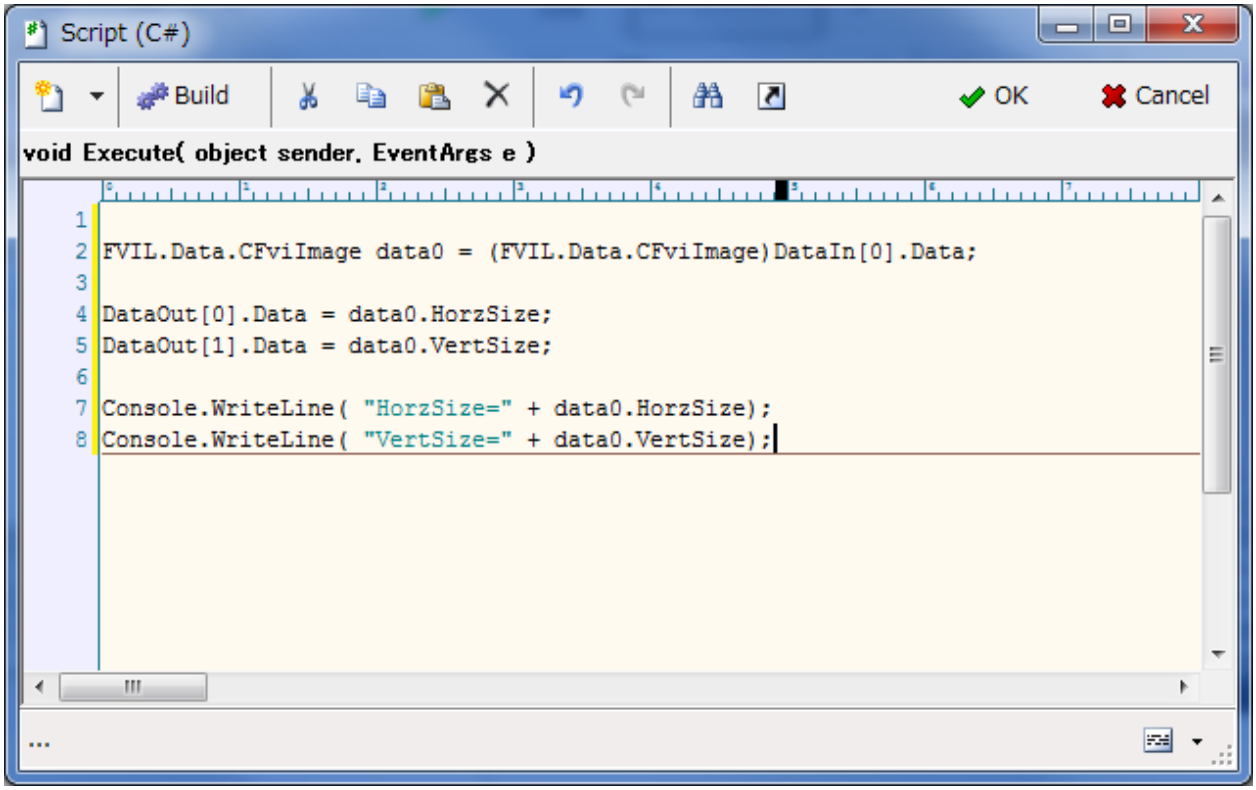


スクリプトの入力データ（In0）にパタンの情報をリンクする事により、スクリプトにパタン情報を渡すことができます。数値情報を渡したい場合は、それをリンクします。

⑤ TASK タブのワークフローリストの最後の行に「スクリプト」を右クリックするとメニューが表示しますので、「Property Dialog」をクリックします。Script エディターが開きます。

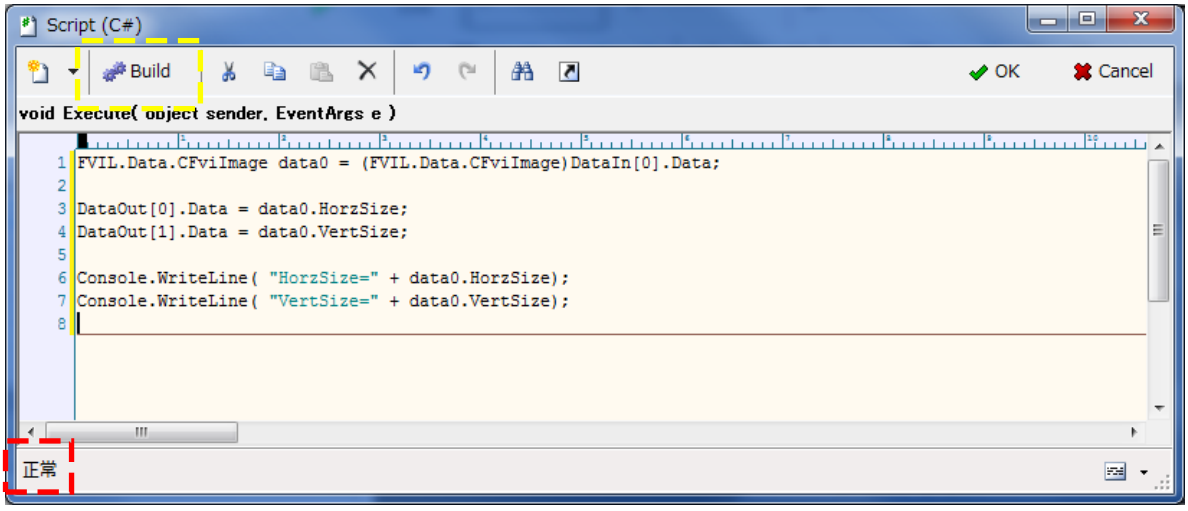


⑥ Script エディターに下記の様に C#でスクリプトを入力します。

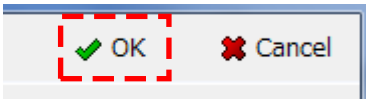


入力データ（In0）にパタンの情報をリンクしましたので、DataIn[0].Data にパタン情報が入っています。それをコピーし、パタンの縦、横サイズを、出力データ DataOut[0].Data にコピーする事で、スクリプトからワークフローに渡すことができます。

⑦ Script エディターの「Build」（黄枠）をクリックし、スクリプトに誤りがなければ、左下に「正常」（赤枠）と表示されます。

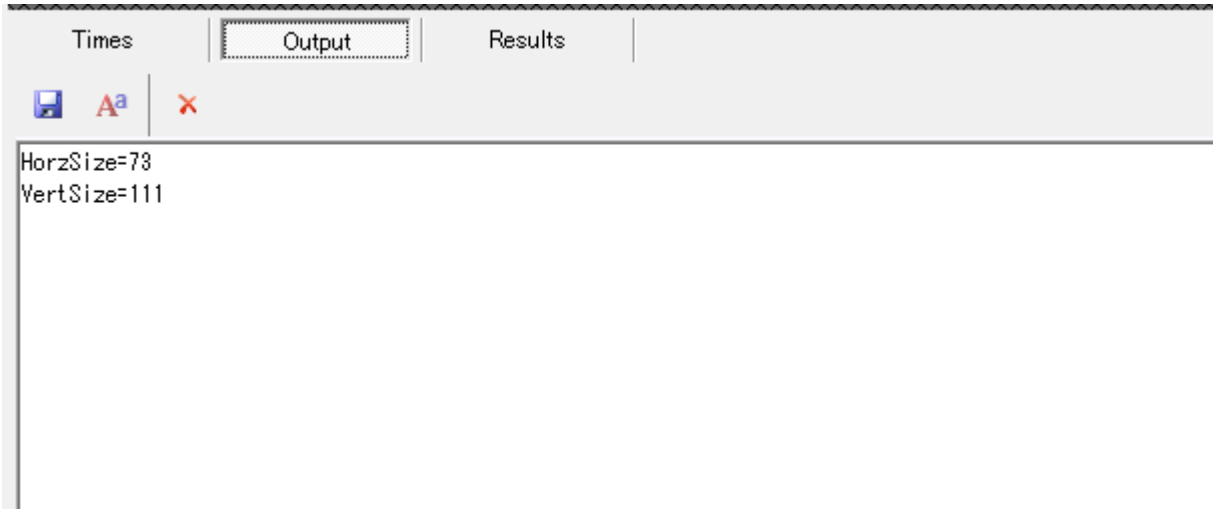


⑧ 「OK」をクリックし、Script エディターを終了します。



⑨ 実行アイコンをクリックしてみましょう。

Result タブの Output タブをクリックすると、コンソールに Script エディターで記載したパタン画像のサイズが表示されます。



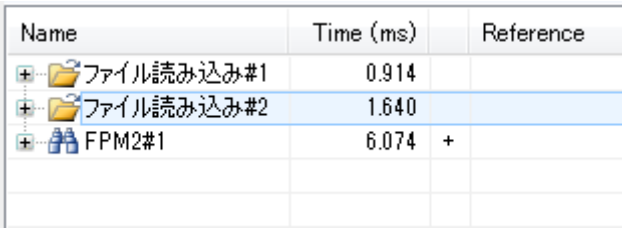
例で登録したパタン画像の横は 73 画素、縦は 111 画素だった事がわかります。

6. サンプルワークフロー解説

WIL-Builder のサンプルワークフローについて、簡単に説明します。

6.1 SAMPLE00

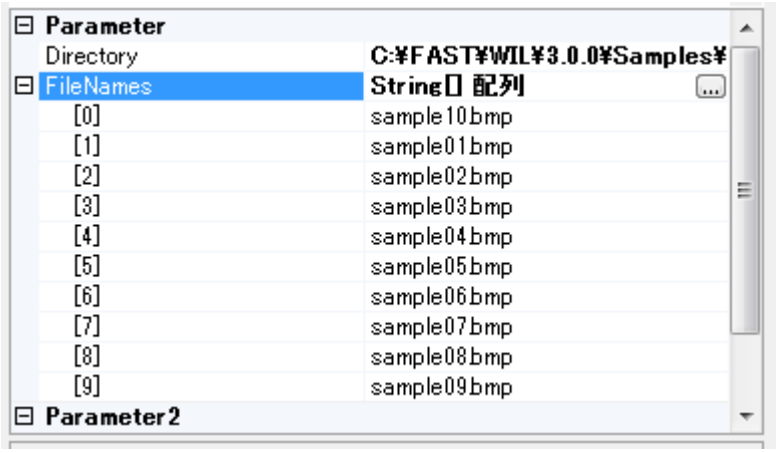
SAMPLE00 を説明します。

機能	FPM 特徴点応用マッチング
内容	ファイルから読み込んだ画像の中から、あらかじめ登録していたパタンを FPM 特徴点応用マッチングを用いてサーチします。
TASK タブの ワークフロー	

POINT　ファイル読み込み#1 について(FVIL.IFviFileAccess.Load)

プロパティグリッドの Parameter2「FileIndexMode」が「Sequential」になっている場合、実行アイコンをクリック毎に読み込みファイルをインデックス順にロードします。

「FileNames」を展開するとファイル名が羅列されている。

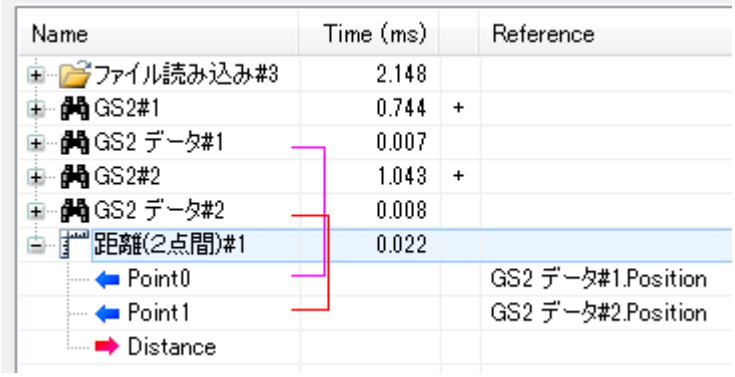


6.2 SAMPLE01

SAMPLE01 を説明します。

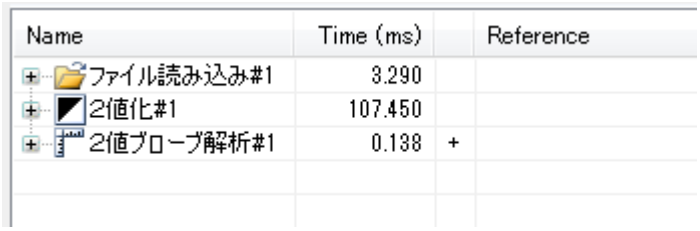
機能	グレイサーチ
内容	ファイルから読み込んだ画像の中から、2 つのマークそれぞれに対して正規化相関サーチを行い、2 つの回答位置の間の距離を計測します。
TASK タブのワークフロー	

POINT 「GS2」のサーチ結果を「GS2 データ」から取得し、Point にリンクします。

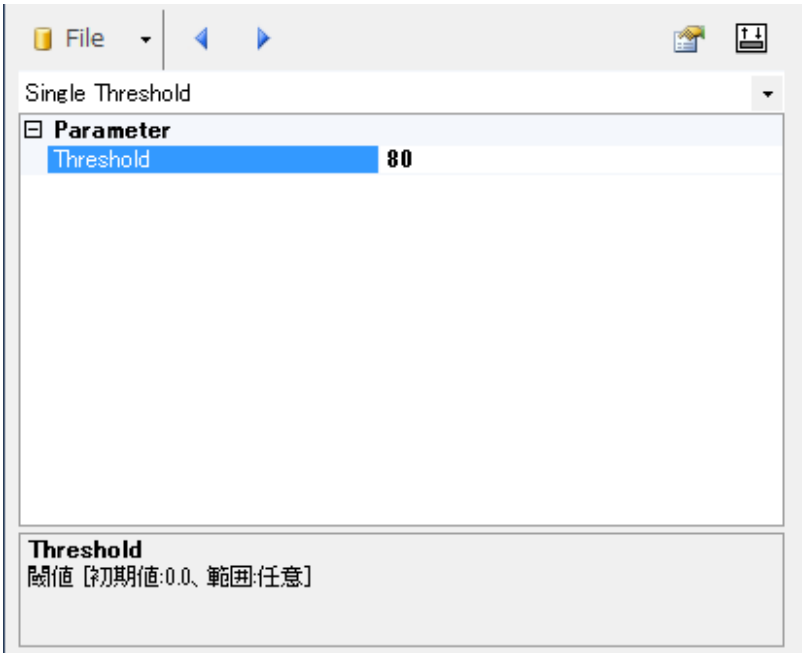


6.3 SAMPLE02

SAMPLE02 を説明します。

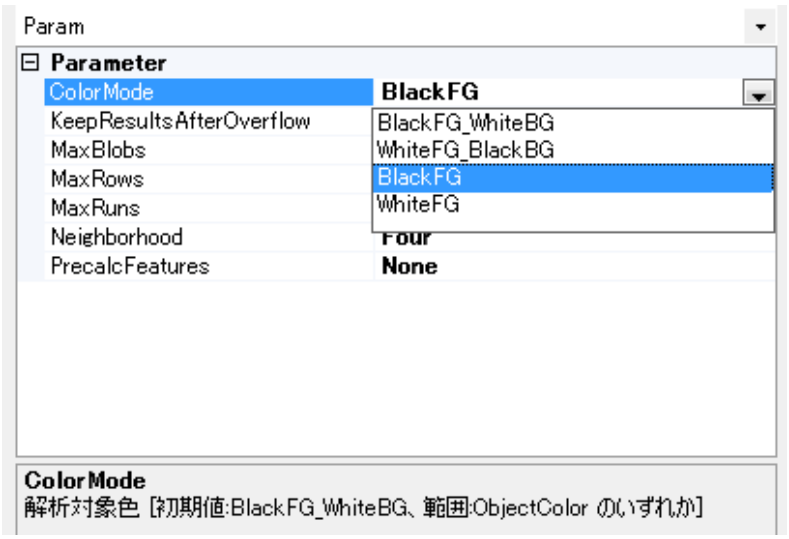
機能	2 値ブローブ解析
内容	ファイルから読み込んだ画像を固定 2 値化で 2 値化し、2 値ブローブ解析を実行し、ブローブを計測します。
TASK タブのワークフロー	

POINT　「2 値化」により濃淡画像を白と黒の 2 値画像に変換します。



※対象ブローブが 2 値化できる最適なノードを選択してください。

POINT　「2 値ブローブ解析」の Parameter「ColorMode」により、計測したい色を指定します。

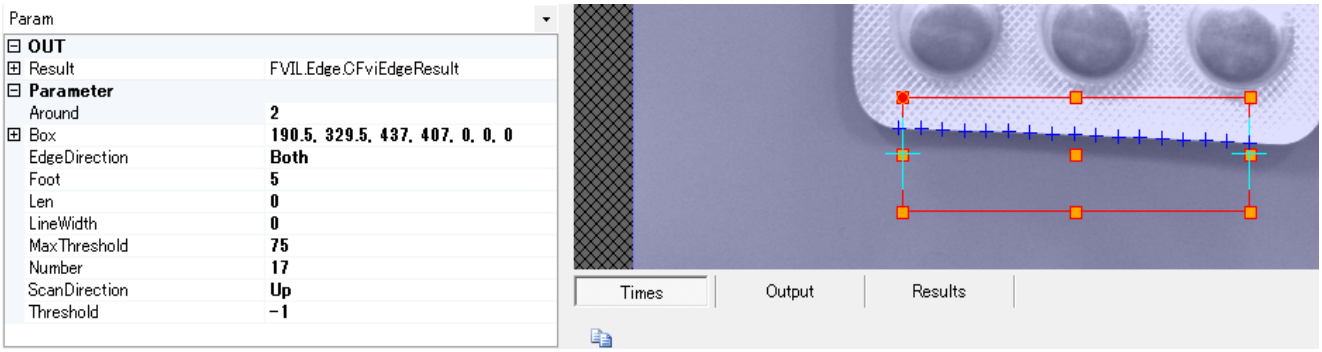


6.4 SAMPLE03

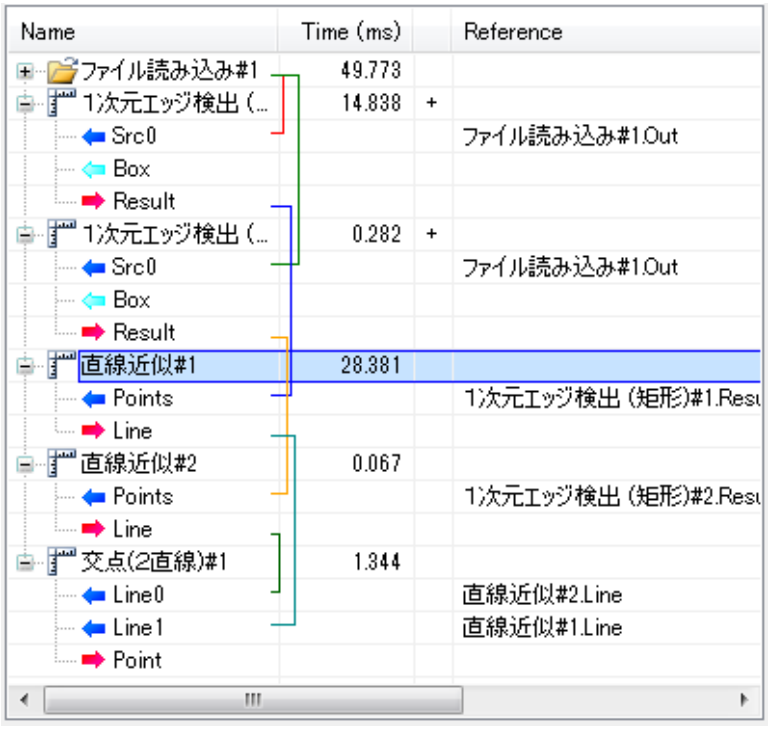
SAMPLE03 を説明します。

機能	コーナー座標計測(R 計測)
内容	ファイルから読み込んだ画像から、指定した 2 か所の範囲から 1 次元エッジを検出し、求めた近似直線から交点を検出します。
TASK タブのワークフロー	

POINT　「1 次元エッジ検出」の検出領域は WOI を用いてマウスで設定できます。



POINT　AUTO　Link により、データの受け渡しが、図のように関連付けされています。

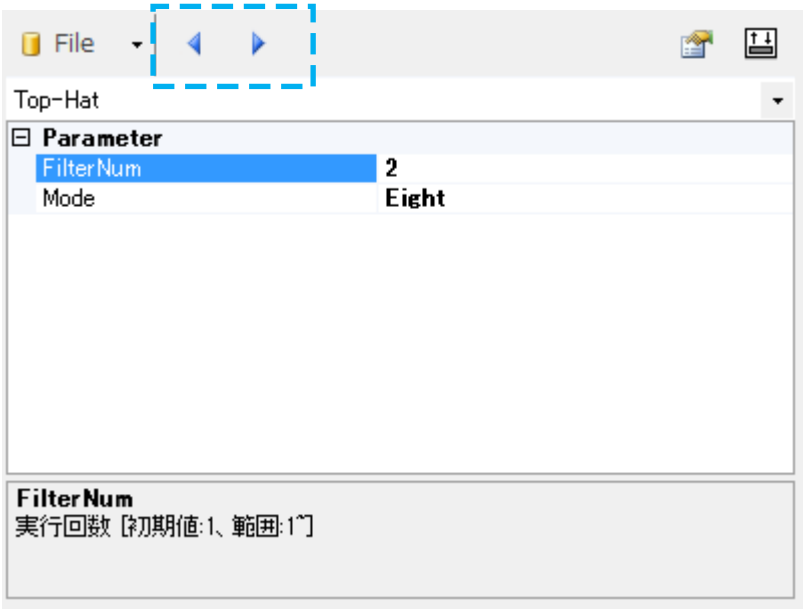


6.5 SAMPLE04

SAMPLE04 を説明します。

機能	モルフォロジを用いたキズ計測
内容	ファイルから読み込んだ画像からにモルフォロジを用いて対象物上のキズを消した後、元画像と画像間差分を行いキズのみ抽出します。明欠陥を抽出する場合は Top-Hat、暗欠陥を抽出する場合は Bottom-Hat を使用します。これらの他に minmax 差分を使用した方法もあります。
TASK タブのワークフロー	

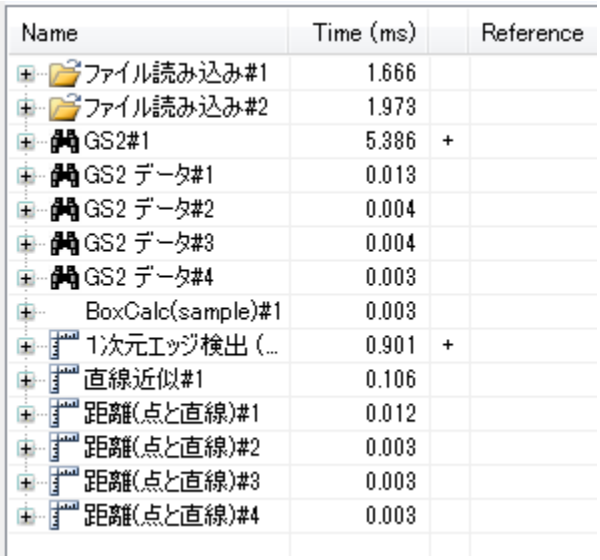
POINT 「モルフォロジ」は複数の機能を持つノードがありますので、青いカーソルで機能の指標を切り替えることができます。 青いカーソルは機能の指標をシーケンスに増減します。
（左：１つ減少、右：１つ増加）



※最適なノードを選択してください。

6.6 SAMPLE05

SAMPLE05 を説明します。

機能	リード長計測
内容	ファイルから読み込んだ画像から正規化相関サーチの回答位置をもとに箱指定エッジ計測、ロバスト推定の直線抽出を行い、直線を抽出します。その後、求めた直線と正規化相関サーチの回答位置の距離を計測します。
TASK タブのワークフロー	

Name	Time (ms)		Reference
ファイル読み込み#1	1.666		
ファイル読み込み#2	1.973		
GS2#1	5.386	+	
GS2 データ#1	0.013		
GS2 データ#2	0.004		
GS2 データ#3	0.004		
GS2 データ#4	0.003		
BoxCalc(sample)#1	0.003		
1次元エッジ検出 (...	0.901	+	
直線近似#1	0.106		
距離(点と直線)#1	0.012		
距離(点と直線)#2	0.003		
距離(点と直線)#3	0.003		
距離(点と直線)#4	0.003		

POINT　プラグイン「BoxCalc」内で、サーチ結果によりエッジ抽出の領域をオフセットする機能を実現しています。同じフォルダにある「sample05_nouse.bff」は、プラグイン「BoxCalc」を使用せず、Function だけで同じ機能を実現していますが、複雑になります。

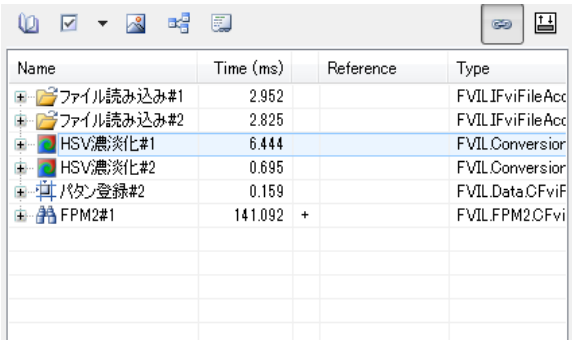
※あらかじめ SAMPLE05 と同じフォルダにある「BoxCalc.dll」はプラグインなので、データディレクトリ(マイ ドキュメントの WIL-Builder 3.0.0)にコピーしておく必要があります。
詳細は、「WIL-Builder 説明書」の「5.2.7 Plugin」をご参照ください。

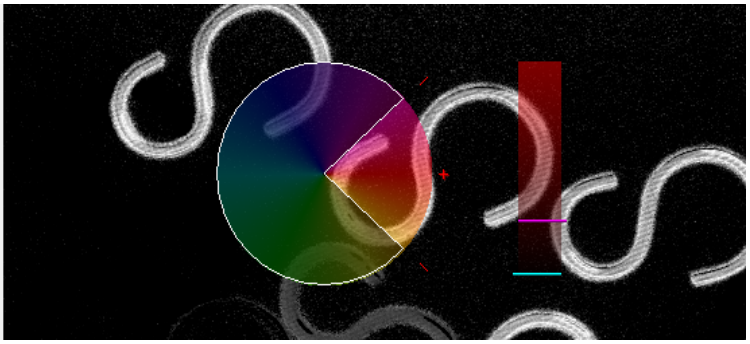
6.7 SAMPLE06

SAMPLE06 を説明します。

機能	カラーサーチ
内容	ファイルから読み込んだ画像とパタン画像から濃淡画像に変換し、FPM 特徴点応用マッチングを用いてサーチします。
TASK タブのワークフロー	

POINT　「HSV 濃淡化」Parameter である色相の基準や範囲、彩度の上下限はマウスで操作することが可能です。



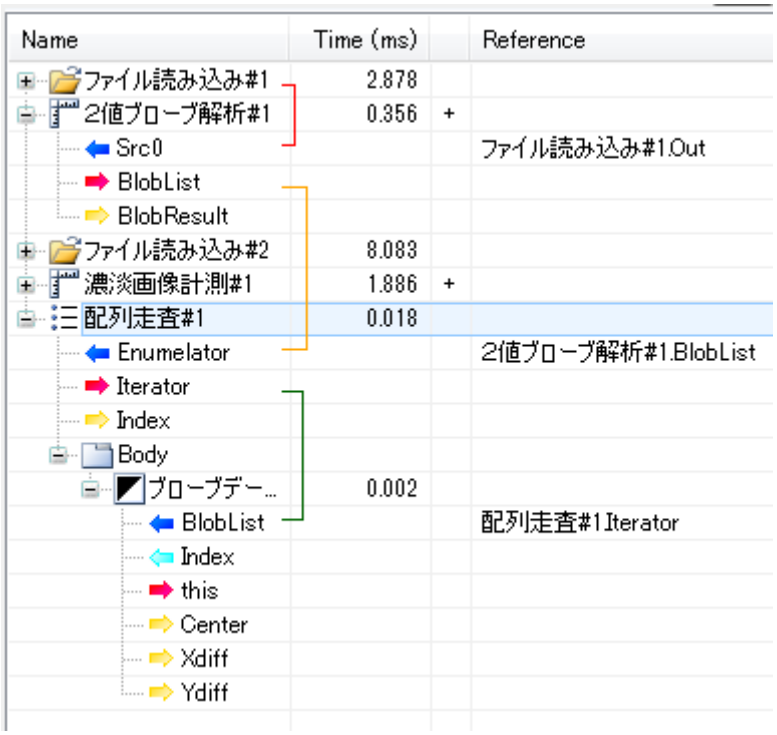


6.8 SAMPLE07

SAMPLE07 を説明します。

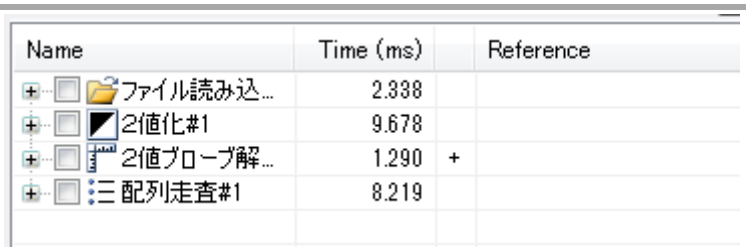
機能	2 値ブローブ解析からリージョン処理
内容	ファイルから読み込んだ 2 値画像から 2 値ブローブ解析を行い、その解析データをもとに、濃淡画像からリージョン処理を行い、濃淡画像から特徴量を計測します。またブローブ毎の中心位置も表示します。
TASK タブのワークフロー	

POINT　「配列走査」に「2 値ブローブ解析」の結果を渡し、検出個数分ループします。
「Body」部分を個数分処理します。

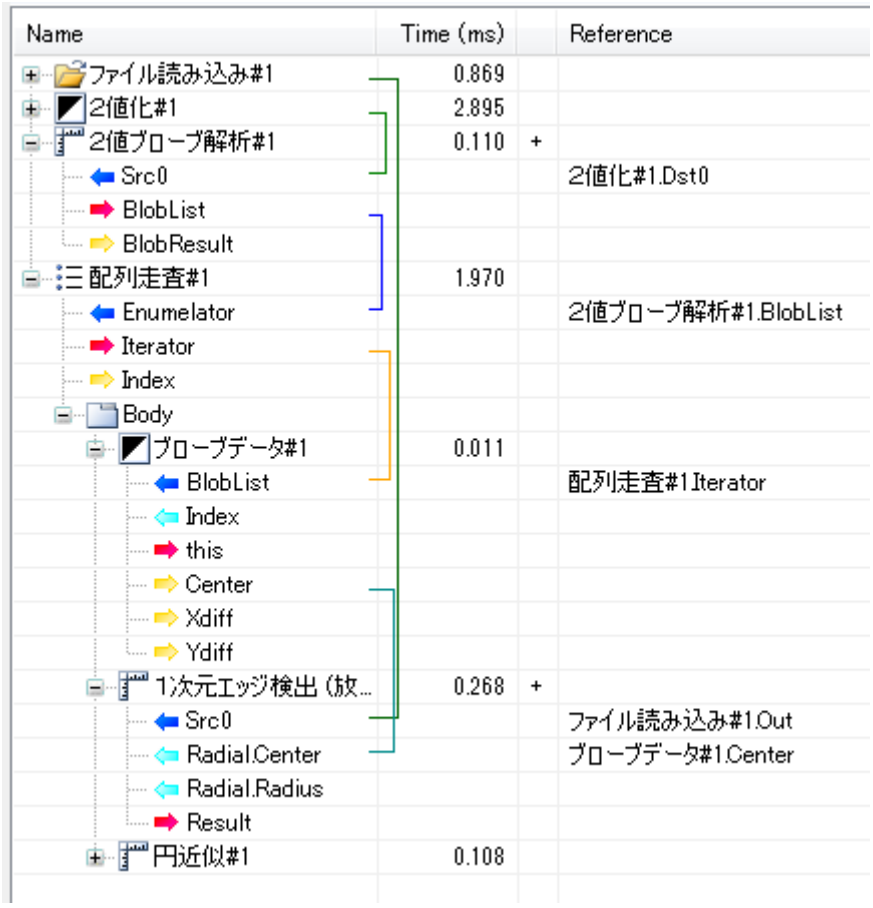


6.9 SAMPLE08

SAMPLE08 を説明します。

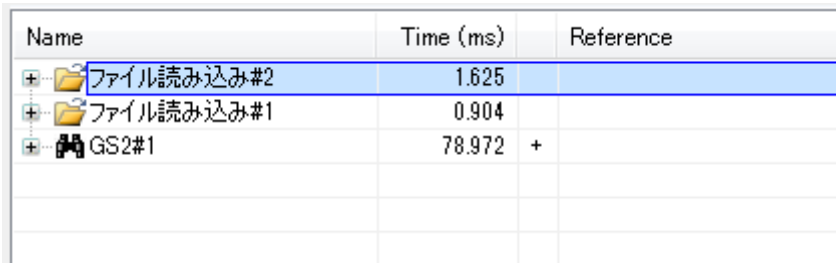
機能	2 値ブローブ解析
内容	ファイルから読み込んだ画像を 2 値化し、2 値ブローブ解析で得られたブローブの重心から放射状に 1 次元エッジ検出し、その得られた座標から円近似を行います。
TASK タブのワークフロー	

POINT　「配列走査」に「2 値ブローブ解析」の結果を渡し、1 次元エッジ検出を行い、そのエッジ群から円近似を行います。



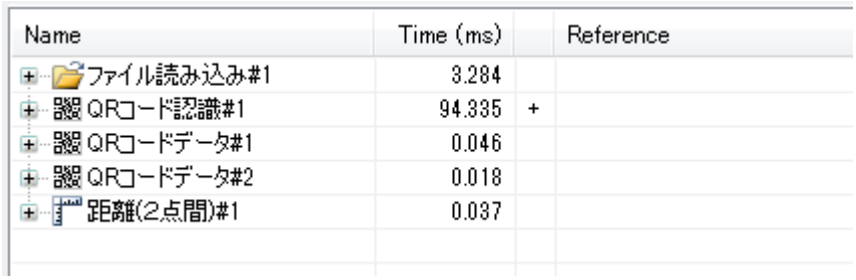
6.10 SAMPLE09

SAMPLE09 を説明します。

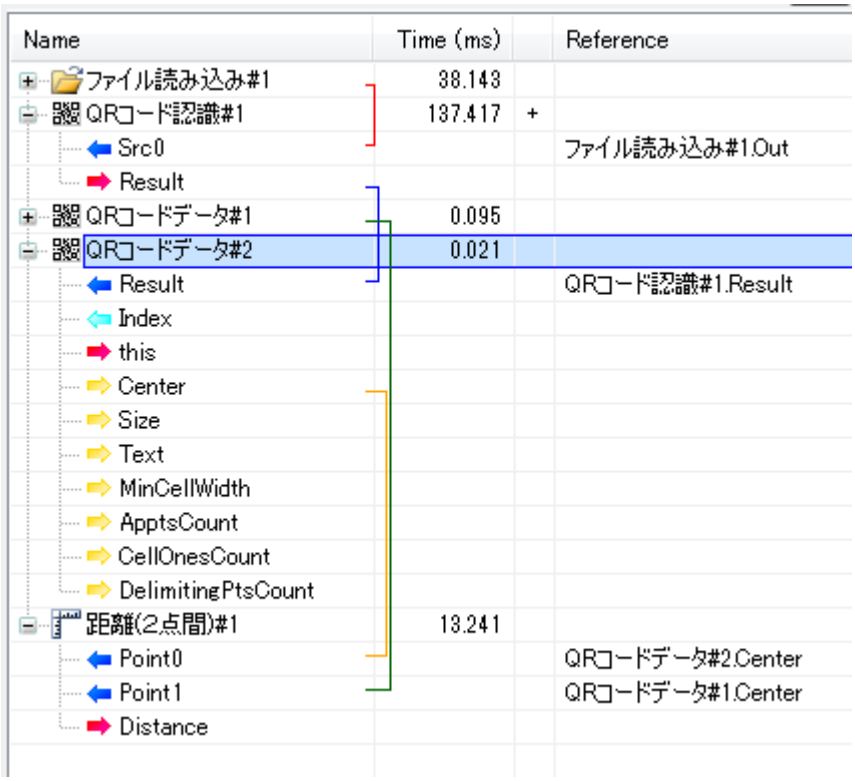
機能	グレイサーチ
内容	ファイルから読み込んだ画像の中から、あらかじめ登録していたパタンをサーチします。
TASK タブの ワークフロー	

6.11 SAMPLE10

SAMPLE10 を説明します。

機能	QR コード認識
内容	ファイルから読み込んだ画像の中から、QR コード認識を行い、2 個の QR コードの距離を計測します。
TASK タブのワークフロー	

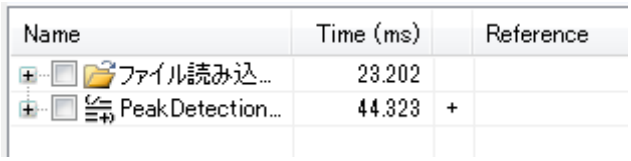
POINT　「QR コードデータ」の Center を「距離(2 点間)」へ渡し、2 点間の距離を算出します。Center は Point 型なので、キャストは必要ありません。



※同じフォルダの QR_SerialPort.bff は、カメラから画像を取り込み、QR コード認識を行い、デコードした文字列をシリアル通信で送信するサンプルです。

6.12 SAMPLE11

SAMPLE11 を説明します。

機能	濃度投影によるピーク個数を出力
内容	入力画像の濃度投影を行い、その結果からピーク探索を行って、ピーク個数を出力します。濃度投影は水平方向のみに対応しています。また、このサンプルでは プラグイン（PeakDetection.dll）を使用しています。 PeakDetection.dll は、処理結果のリストビュー表示を実装しています。 (FVIL.Parser.IParserNodeStatisticsCtrl)
TASK タブの ワークフロー	

POINT　プラグイン「BoxCalc」内で、サーチ結果によりエッジ抽出の領域をオフセットする機能を実現しています。同じフォルダにある「sample05_nouse.bff」は、プラグイン「BoxCalc」を使用せず、Function だけで実現しています。

※あらかじめ SAMPLE11 と同じフォルダにある「PeakDetection.dll」はプラグインなので、データディレクトリ(マイ ドキュメントの WIL-Builder 3.0.0)にコピーしておく必要があります。
詳細は、「WIL-Builder 説明書」の「5.2.7 Plugin」をご参照ください。

7. 困ったときは

WIL のセットアップや動作確認で問題が発生した場合には以下の内容をご確認ください。ご確認いただいても解消されない場合は、ユーザ・サポートまでお問い合わせください。

7.1 トラブルシューティング

(1) WIL-Builder が起動できない

WIL-Builder、WILsample (C#)、WILsample (VB) 等、 .NET Framework を使用したサンプルアプリケーションは .NET Framework 2.0 (SP2) がインストールされている必要があります。
また、弊社が提供する FVIL アセンブリを GAC (グローバルアセンブリキャッシュ) に登録する必要が有ります。

GACに関しては、リリースノート「GAC の登録と解除」の項をご参照ください。

7.2 ユーザ・サポートについて

弊社製品につきましてのお問い合わせはユーザ・サポートにて承ります。
ご質問の内容に下記の必要事項をお書き添えいただき、e-mail、FAX もしくは TEL にてお問い合わせください。

・御社名
・部署名
・お名前
・ユーザ登録番号
・機種名またはライブラリ名
・システムおよびライブラリのバージョン

e-mail でのお問い合わせ：support@fast-corp.co.jp
FAX でのお問い合わせ：046-272-8692
TEL でのお問い合わせ：046-272-8691
受付時間：月曜日～金曜日(祝祭日および弊社指定の休日を除く)
9:00～12:00　13:00～17:00

◆ソースファイルの添付についてのお願い◆

作成されたプログラムがハングアップしてしまう等のトラブルを抱えたお客様より、ソースファイルが添付された e-mail をいただくことがありますが、まずは以下のような詳しい状況をご説明いただき、ユーザ・サポートまでご相談ください。サポートの過程で弊社にてソースファイルの解析が必要だと判断した場合には、ソースファイルの添付をお願いしております。

・お使いのライブラリ製品の名称
・お使いのシステム、ライブラリのバージョン
・プログラムのどの部分(どのライブラリ)でハングアップするのか細かい情報
・問題の部分に関連のあるライブラリなどの戻り値は正常終了しているか
・発生頻度

WIL-Builder　チュートリアル

2017 年 10 月 第 3 版発行

発行所　　株式会社ファースト

本　　社　〒242-0001　神奈川県大和市下鶴間 2791-5

ユーザ・サポート　FAX　046-272-8692　　TEL　046-272-8691
E-mail: support@fast-corp.co.jp

B-002799