

CONTENTS

ビジュアルヘルプ – ImageAnalyzeParticles.....	2
ImageAnalyzeParticles コマンドのヘルプ.....	2

ビジュアルヘルプ – ImageAnalyzeParticles

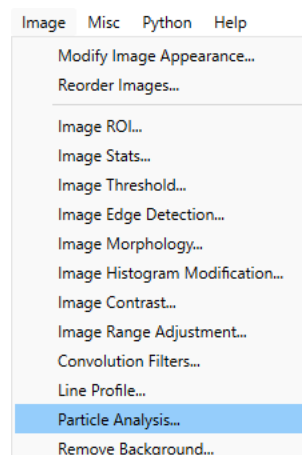
ImageAnalyzeParticles コマンドのヘルプ

ImageAnalyzeParticles [flags] keyword imageMatrix

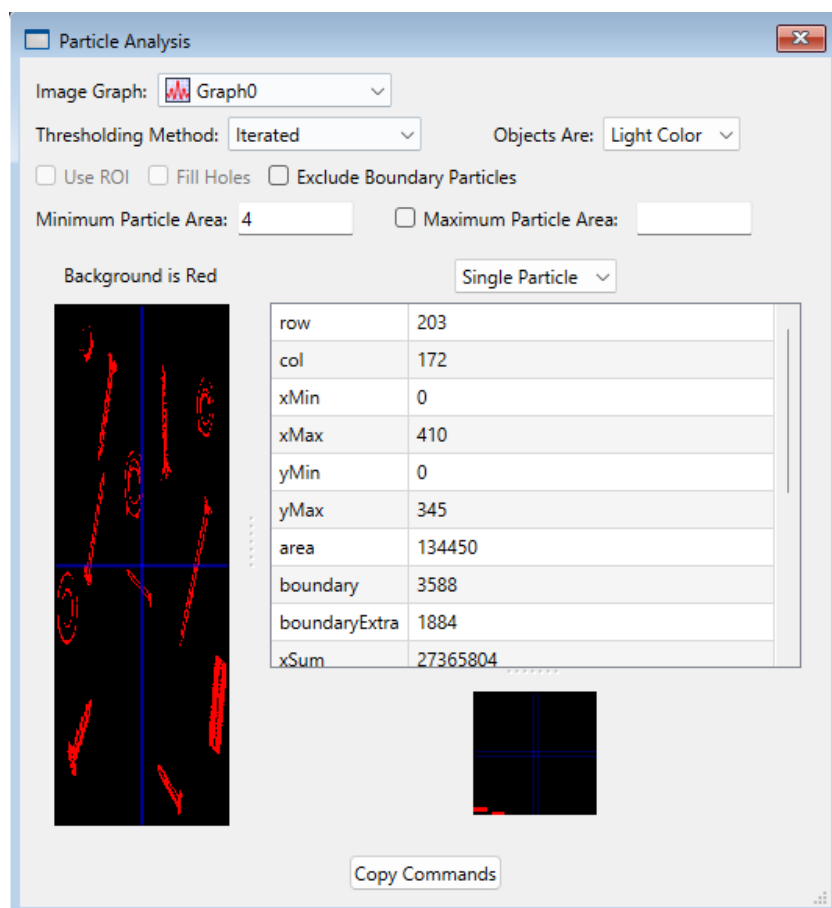
ImageAnalyzeParticles コマンドは、2D または 3D のソースウェーブ画像マトリックスに対して、2 種類の粒子解析コマンドのどちらかを実行します。

ソースウェーブ画像はバイナリ形式、つまり、粒子を 0、背景を 255 で指定する unsigned char 形式である必要があります（データで逆の指定を使っている場合、このコマンドは誤った結果を生成します）。なお、ソース画像内の 0 以外のすべての値は、背景の一部とみなされません。

グレースケール画像は、このコマンドを実行する前に閾値処理を行う必要があります（ImageThreshold コマンドで /I フラグを使う必要がある場合があります）。



メニュー Image→Particle Analysis では視覚的に操作するためのダイアログを表示できます。



注記：

ImageAnalyzeParticles はウェーブスケールを考慮していません。

すべての測定値はピクセル単位であり、すべてのピクセルは正方形であると仮定されています。

パラメーター

keyword は、以下のいずれかの名前です。

mark	<p>/L フラグを使って内部（シード）ピクセルで指定された1つの粒子に対するマスキング画像を作成します。このマスキング画像は、unsigned char 型のウェーブである M_ParticleMarker に格納されます。M_ParticleMarker 内のすべてのポイントは 64 に設定されます（バイナリウェーブでの画像演算では、値 64 が NaN 相当として指定されます）。ただし、粒子内のポイントは 0 に設定されます。このウェーブは、元の画像の上にオーバーレイとして使われるように設計されています（ModifyImage の explicit=1 モードを使用）。</p> <p>このキーワードは、現在 ImageSeedFill に置き換えられています。</p>														
stats	<p>画像内の粒子を計測します。計測結果は、/A フラグで指定された minArea を超える面積を持つすべての粒子について報告されます。計測結果は以下の通りです：</p> <table><tr><td>V_NumParticles</td><td>minArea の閾値を超える粒子の数。</td></tr><tr><td>W_ImageObjArea</td><td>各粒子の面積（ピクセル単位）。</td></tr><tr><td>W_ImageObjPerimeter</td><td>各粒子の周囲長（ピクセル単位）。周囲長の計算には 45 度のエッジに対する推定値が含まれるため、整数以外の値となります。</td></tr><tr><td>W_circularity</td><td>周囲長の二乗と $(4 * \pi * \text{objectArea})$ の比。完全な円の場合、1 に近づきます。</td></tr><tr><td>W_rectangularity</td><td>粒子の面積と、内接する（回転していない）長方形の面積との比。この比は、完全な円形の物体では $n/4$ となり、回転していない長方形では 1 となります。</td></tr><tr><td>W_SpotX/W_SpotY</td><td>各オブジェクトから1つの x, y 座標を取得します。粒子ごとに1つのエントリがあり、これらのエントリは、このコマンドによって生成される他のすべてのウェーブと同じ順序で並んでいます。これらのウェーブから得られる各 (x, y) 座標は、粒子のタグや注釈の位置を定義するために使用できます。また、これらの座標は、関連する mark メソッドや ImageSeedFill コマンドのシードピクセルとしても使用できます。</td></tr><tr><td>W_xmin, W_xmax, W_ymin, W_ymax</td><td>各粒子には、X 軸と Y 軸に沿った内接長方形を定義する1つのポイントが含まれています。</td></tr></table>	V_NumParticles	minArea の閾値を超える粒子の数。	W_ImageObjArea	各粒子の面積（ピクセル単位）。	W_ImageObjPerimeter	各粒子の周囲長（ピクセル単位）。周囲長の計算には 45 度のエッジに対する推定値が含まれるため、整数以外の値となります。	W_circularity	周囲長の二乗と $(4 * \pi * \text{objectArea})$ の比。完全な円の場合、1 に近づきます。	W_rectangularity	粒子の面積と、内接する（回転していない）長方形の面積との比。この比は、完全な円形の物体では $n/4$ となり、回転していない長方形では 1 となります。	W_SpotX/W_SpotY	各オブジェクトから1つの x, y 座標を取得します。粒子ごとに1つのエントリがあり、これらのエントリは、このコマンドによって生成される他のすべてのウェーブと同じ順序で並んでいます。これらのウェーブから得られる各 (x, y) 座標は、粒子のタグや注釈の位置を定義するために使用できます。また、これらの座標は、関連する mark メソッドや ImageSeedFill コマンドのシードピクセルとしても使用できます。	W_xmin, W_xmax, W_ymin, W_ymax	各粒子には、X 軸と Y 軸に沿った内接長方形を定義する1つのポイントが含まれています。
V_NumParticles	minArea の閾値を超える粒子の数。														
W_ImageObjArea	各粒子の面積（ピクセル単位）。														
W_ImageObjPerimeter	各粒子の周囲長（ピクセル単位）。周囲長の計算には 45 度のエッジに対する推定値が含まれるため、整数以外の値となります。														
W_circularity	周囲長の二乗と $(4 * \pi * \text{objectArea})$ の比。完全な円の場合、1 に近づきます。														
W_rectangularity	粒子の面積と、内接する（回転していない）長方形の面積との比。この比は、完全な円形の物体では $n/4$ となり、回転していない長方形では 1 となります。														
W_SpotX/W_SpotY	各オブジェクトから1つの x, y 座標を取得します。粒子ごとに1つのエントリがあり、これらのエントリは、このコマンドによって生成される他のすべてのウェーブと同じ順序で並んでいます。これらのウェーブから得られる各 (x, y) 座標は、粒子のタグや注釈の位置を定義するために使用できます。また、これらの座標は、関連する mark メソッドや ImageSeedFill コマンドのシードピクセルとしても使用できます。														
W_xmin, W_xmax, W_ymin, W_ymax	各粒子には、X 軸と Y 軸に沿った内接長方形を定義する1つのポイントが含まれています。														

/M 指定に応じて、以下の4種類のウェーブのいずれかを作成できます。これらのウェーブは、元の画像の上に重ねて表示されるように設計されています (ModifyImage の explicit=1 モードを使用)。

注記：

これらのウェーブを作成するのにかかる追加時間は、統計データを生成するのにかかる時間と比べれば、ごくわずかです。

M_ParticlePerimeter 粒子境界のマスキング画像。これは、オブジェクトの境界には 0、それ以外のすべてのポイントには 64 が格納された unsigned char 型のウェーブデータです。

M_ParticleArea 粒子が占める領域のマスキング画像。これは unsigned char 型のウェーブデータであり、オブジェクトの境界には 0、それ以外のすべてのポイントには 64 が格納されています。また、/A で指定された最小サイズより小さい粒子は含まれていないという点でも、入力画像とは異なります。

M_Particle 粒子の領域と境界の両方を表すマスキング画像。これは unsigned char 型のウェーブデータであり、オブジェクトの領域には値 16、オブジェクトの境界には値 18、その他のすべてのポイントには値 64 が格納されています。

M_rawMoments 5列のウェーブデータ。1列目は各粒子の x 値の合計、2列目は y 値の合計です。粒子の平均値または「中心」を求めるには、これらの値を対応する面積で割ります。3列目には x^2 の和、4列目には y^2 の和、5列目には $x*y$ の和が含まれます。

imageMatrix が 3D ウェーブの場合、さまざまな結果は1つの 2D ウェーブ M_3DParticleInfo に格納されます。

これは、各粒子につき 1 行 11 列で構成されています。

列は、minRow、maxRow、minCol、maxCol、minLayer、maxLayer、xSeed、ySeed、zSeed、volume、area の順に配置されます。

Edit M_3DParticleInfo.Id を使って、各列を説明する次元ラベル付きのテーブルで結果を表示します。

フラグ

/A=*minArea*

粒子をカウントするための閾値として、最小面積を指定します (例：1 ピクセルの粒子を検出するには、*minArea*=0 を指定します)。最小面積はピクセル単位で測定され、デフォルト値は *minArea*=5 です。

ソースウェーブが 3D の場合、*minArea* は粒子を構成するボクセルの最小数を指定します。

/A は *mark* メソッドと併用しても効果はありません。

<i>/B</i>	境界から 1 ピクセル幅の枠を削除します。これにより、粒子が境界ピクセルを持つことがなくなり（後述の <i>/EBPC</i> を参照）、すべての境界ウェーブが滑らかな輪郭を描くようになります。
<i>/CIRC={minCircularity, maxCircularity}</i>	このフラグを使うと、指定した円形度範囲内の粒子のみがカウントされるように出力をフィルタリングできます。
<i>/D=dataWave</i>	<i>stats</i> キーワードと併用する場合、粒子の強度の最小値、最大値、平均値をサンプリングするウェーブを指定します。 <i>dataWave</i> は、入力バイナリ画像 <i>imageMatrix</i> と同じ次元数でなければなりません。任意の実数型を使用できます。結果は、ウェーブ <i>W_IntMax</i> 、 <i>W_IntMin</i> 、 <i>W_IntAvg</i> として返されます。
<i>/DS3D=dest3DInfo</i>	このフラグを使って、3D 粒子情報ウェーブの保存先を指定します。フラグが指定されていない場合、コマンドはデータを現在のデータフォルダー内のウェーブ <i>M_3DParticleInfo</i> に保存します。
<i>/DSBI=destBIndex</i>	このフラグを使って、境界インデックスウェーブの保存先を指定します。フラグが指定されていない場合、コマンドはインデックスを現在のデータフォルダー内のウェーブ <i>W_BoundaryIndex</i> に保存します。
<i>/DSPA=destPAWave</i>	このフラグを使って、粒子領域ウェーブの保存先を指定します。フラグが指定されていない場合、コマンドはウェーブ <i>M_ParticleArea</i> のデータを現在のデータフォルダーに保存します。
<i>/DSPL=destMPWave</i>	このフラグを使って、粒子のマスキング画像の保存先を指定します。フラグが指定されていない場合、マスキング画像を現在のデータフォルダー内のウェーブ <i>M_Particle</i> に保存します。
<i>/DSPP=destPerimeter</i>	このフラグを使って、粒子境界ウェーブの保存先を指定します。フラグが指定されていない場合、コマンドはデータを現在のデータフォルダー内の <i>M_ParticlePerimeter</i> ウェーブに保存します。
<i>/DSTA=destIOAWave</i>	このフラグを使って、宛先オブジェクト領域のウェーブファイルを指定します。フラグが指定されていない場合、コマンドはデータを現在のデータフォルダー内のウェーブ <i>W_ImageObjArea</i> に保存します。
<i>/DSTC=destCircularity</i>	このフラグを使って、円形ウェーブの保存先を指定します。フラグが指定されていない場合、コマンドはデータを現在のデータフォルダー内の <i>W_circularity</i> ウェーブに保存します。
<i>/DSTM=destMarkerWave</i>	このフラグを使って、保存先のマーカーウェーブを指定します。フラグが指定されていない場合、コマンドはマーカーウェーブを現在のデータフォルダー内の <i>M_ParticleMarker</i> ウェーブに保存します。
<i>/DSTP=destIOPWave</i>	このフラグを使って、宛先オブジェクトの境界ウェーブを指定します。ウェーブが指定されていない場合、コマンドはデータを現

在のデータフォルダー内のウェーブ W_ImageObjPerimeter に保存します。

/DSTR=destRectWave

このフラグを使って、オブジェクトの矩形度ウェーブの保存先を指定します。フラグが指定されていない場合、コマンドはデータを現在のデータフォルダー内のウェーブ W_rectangularity に保存します。

/DSTX=destSpotXW

このフラグを使って、spotX ウェーブの保存先を指定します。フラグが指定されていない場合、コマンドはウェーブ W_SpotX のデータを現在のデータフォルダーに保存します。

/DSTXM=destXMaxW

このフラグを使って、xmax 位置のウェーブの保存先を指定します。フラグが指定されていない場合、コマンドはデータを現在のデータフォルダー内のウェーブ W_xmax に保存します。

/DSTY=destSpotYW

このフラグを使って、spotY ウェーブの保存先を指定します。フラグが指定されていない場合、コマンドはデータを現在のデータフォルダー内の W_SpotY ウェーブに保存します。

/DSRM=destRMWave

このフラグを使って、生のモーメントウェーブの保存先を指定します。フラグが指定されていない場合、コマンドはデータを現在のデータフォルダー内の M_rawMoments ウェーブに保存します。

/DSIN=destIntMin

このフラグを使って、最小粒子強度ウェーブの保存先を指定します。フラグが指定されていない場合、コマンドはデータを現在のデータフォルダー内のウェーブ W_IntMin に保存します。

/DSIA=destIntAvg

このフラグを使って、平均強度ウェーブの保存先を指定します。フラグが指定されていない場合、コマンドはデータを現在のデータフォルダー内のウェーブ W_IntAvg に保存します。

/DSIX=destIntMax

このフラグを使って、最大粒子強度ウェーブの保存先を指定します。フラグが指定されていない場合、コマンドはデータを現在のデータフォルダー内のウェーブ W_IntMax に保存します。

/DSBX=destBoundaryX

このフラグを使って、boundaryX ウェーブの保存先を指定します。フラグが指定されていない場合、コマンドはデータを現在のデータフォルダー内の W_BoundaryX ウェーブに保存します。

/DSBY=destBoundaryY

このフラグを使って、boundaryY ウェーブの保存先を指定します。フラグが指定されていない場合、コマンドはデータを現在のデータフォルダー内の W_BoundaryY ウェーブに保存します。

/DSMM=dest3dMoments

このフラグを使って、3D モーメントのウェーブデータを保存する保存先を指定します。フラグが指定されていない場合、コマンドはウェーブデータ M_Moments を現在のデータフォルダーに保存します。

/DSXN=destXMinWave

このフラグを使って、XMin ウェーブデータの保存先を指定します。フラグが指定されていない場合、コマンドはウェーブ W_xmin のデータを現在のデータフォルダーに保存します。

/DSYN=destYMinWave

このフラグを使って、YMin ウェーブを保存する保存先を指定します。フラグが指定されていない場合、コマンドはウェーブ W_ymin のデータを現在のデータフォルダーに保存します。

/DSYM=destYMaxWave	このフラグを使って、YMax ウェーブの保存先を指定します。フラグが指定されていない場合、コマンドはウェーブ W_ymax のデータを現在のデータフォルダーに保存します。
/E	各粒子に最もよく適合する楕円を計算します。等価な楕円は、まず粒子のモーメント（すなわち、平均 x 値、平均 y 値、平均 x^2 、平均 y^2 、および平均 $x*y$ ）を求め、次に楕円の面積が粒子の面積と等しくなるようにすることで計算されます。 算出された楕円は、ウェーブ M_Moments に保存されます。 imageMatrix が 2D ウェーブの場合、M_Moments に返される結果は列ごとに、楕円の X 座標の中心、楕円の Y 座標の中心、長軸、短軸、長軸が X 方向と成す角度（ラジアン）となります。 imageMatrix が 3D ウェーブの場合、M_Moments 内の結果には、X、Y、Z 成分の和に加え、それらの積のすべての 2 次順列が含まれます。これらは、sumX、sumY、sumZ、sumXX、sumYY、sumZZ、sumXY、sumXZ、sumYZ の順に配置されます。
/EBPC	このフラグを使うと、画像の境界上に 1 つ以上のピクセルを持つ粒子をカウント対象から除外できます。
/F	内部に穴のある 2D 粒子を塗りつぶし、穴を除去するためにその面積を調整します。穴の周囲にある内部境界も削除されます。粒子の境界が、各境界ピクセルを 1 回だけ通過する 1 つの閉じたパスとして横断できない薄い要素で構成されている場合、その粒子は塗りつぶされません。 粒子の塗りつぶしは実行時間を大幅に増加させる可能性があり、画像によっては大量のメモリを必要とする場合があることに注意してください。より効率的なアプローチとしては、二値画像を前処理し、モルフォロジー演算を使って穴を除去することが考えられます。 imageMatrix が 3D ウェーブの場合、このフラグはサポートされません。
/FILL	/FILL オプションを使うと、粒子内部の穴を埋めることができます。算出される面積および周囲長の値は、穴が存在しないものとして計算されます。例えば、粒子の周囲にゼロ値で構成される閉じた輪郭がある場合など、埋め込みアルゴリズムが正常に動作しないことがあります。 /F オプションと /FILL オプションの両方を指定した場合、処理では /FILL のみが使われます。 Igor Pro 7.0 で追加されました。
/FREE	指定されたすべての宛先ウェーブをフリーウェーブとして作成します。このフラグは、ユーザーが宛先を指定せずにデフォルトで作成される出力ウェーブには影響しません。 /FREE は関数内でのみ使用可能であり、かつ /DEST で指定された destWave が単純な名前またはウェーブ参照構造体のフィールドである場合に限り使用できます。 詳細は、ヘルプ Free Waves を参照してください。 Igor Pro 10.0 で追加されました。

/L=(row,col)	mark メソッドに関連して、2D 粒子の位置を指定します。(row, col) は、粒子内の任意のピクセルに対応するシード値です。シードが粒子の境界に属する場合、その粒子は塗りつぶされません。 <i>imageMatrix</i> が 3D ウェーブの場合、このフラグはサポートされません。										
/M=markerVal	2D 画像の stats と組み合わせて使います。ウェーブに関する詳細な説明については、stats キーワードを参照してください。										
	<table border="0"> <tr> <td>markerVal</td> <td>ウェーブ</td> </tr> <tr> <td>0</td> <td>マーカーウェーブなし</td> </tr> <tr> <td>1</td> <td>M_ParticlePerimeter</td> </tr> <tr> <td>2</td> <td>M_ParticleArea</td> </tr> <tr> <td>3</td> <td>M_Particle</td> </tr> </table>	markerVal	ウェーブ	0	マーカーウェーブなし	1	M_ParticlePerimeter	2	M_ParticleArea	3	M_Particle
markerVal	ウェーブ										
0	マーカーウェーブなし										
1	M_ParticlePerimeter										
2	M_ParticleArea										
3	M_Particle										
	<i>imageMatrix</i> が 3D ウェーブの場合、このフラグはサポートされません。										
/MAXA=maxArea	stats キーワードと併用する場合、許容される粒子の面積の上限を指定します。面積はピクセル単位で測定され、 <i>maxArea</i> のデフォルト値は画像のピクセル数です。3D の場合、この最大値はボクセルの数に適用されます。										
/NSW	stats キーワードと組み合わせて使うと、マーカーウェーブ (/M フラグを参照) は生成されますが、粒子の統計ウェーブは生成されません。これにより、粒子が多数含まれる画像の実行時間を短縮できるはずですが。										
/P=plane	3D ウェーブの1つのレイヤーに対して操作を行う時の平面を指定します。										
/PADB	このフラグを stats キーワードと併用すると、画像の周囲に幅 1 ピクセルの背景が追加されます。これにより、画像の境界に接している粒子は、閉じた周囲を持つ内部粒子 (元の画像フレームから 1 ピクセル外側に拡張されたもの) として扱われるようになります。また、境界にあるすべての粒子について、ウェーブ W_ObjPerimeter のウェーブ長が長くなり、これにより円形度などの他の派生パラメーターにも影響が及びます。										
	/PADB は、画像の境界上に位置する粒子に属するすべてのピクセルを考慮するという点で、/B とは異なります。これら 2 つのフラグは排他的です。										
	Igor Pro 7.0 で追加されました。										
/PDLG	進捗ダイアログを表示します。										
	/PDLG は、非常に大きな 3D 画像を処理する時に役立ちます。進行ダイアログにより処理状況が確認でき、ユーザーは操作を中止することも可能です。										
	Igor Pro 9.0 で追加されました。										
/Q	静音モード。履歴エリアに粒子数を表示しません。										
/R=roiWave	関心領域 (ROI) を指定します。ROI は、 <i>imageMatrix</i> と同じ行数と列数を持つ unsigned byte 型 (/b/u) のウェーブによって定義されます。ROI 自体は、 <i>roiWave</i> 内の値が 0 であるエントリ										

またはピクセルによって定義されます。ROI 外のピクセルは、0 以外の任意の値を持つことができます。

ROI は連続している必要はありません。

imageMatrix が 3D ウェーブの場合、*roiWave* は 2D ウェーブ (*imageMatrix* の行数および列数と一致するもの) であるか、または *imageMatrix* と同じ行数、列数、レイヤー数を持つ 3D ウェーブである必要があります。3D の *imageMatrix* に対して 2D の *roiWave* を使う場合、ROI は 3D ウェーブの各レイヤーについて *roiWave* によって定義されるものとみなされます。

2D ROI ウェーブの作成に関する詳細は、ImageGenerateROIMask コマンドのヘルプを参照してください。

/U mark キーワードと併用する場合、ウェーブ M_ParticleMarker をデフォルトの 16 ビットではなく、8 ビットの符号なし整数として保存します。

/W 2D *imageMatrix* ウェーブに対して、境界ウェーブ W_BoundaryX、W_BoundaryY、W_BoundaryIndex を作成します。W_BoundaryX と W_BoundaryY には、粒子の境界に沿ったピクセルが含まれます。各粒子の境界は、両方のウェーブにおいて NaN 要素で終わります。W_BoundaryIndex の各要素は、W_BoundaryX と W_BoundaryY における新しい粒子の開始位置へのインデックスであるため、各粒子の境界を素早く特定することができます。

粒子に穴がある場合、W_BoundaryX と W_BoundaryY のエントリは、外部境界から始まり、その後その粒子のすべての内部境界が続きます。内部境界に対するインデックスエントリは存在しません。

imageMatrix が 3D ウェーブの場合、このフラグはサポートされません。

詳細

粒子解析は、まずデータを元の形式からバイナリ表現に変換することから行われます。

このバイナリ表現では、粒子は 0 で、背景は 0 以外の値で表されます。

アルゴリズムは、粒子に属する最初のピクセルまたはボクセルを検索し、その起点から粒子を拡大させながら、粒子内の面積、周囲長、およびピクセルまたはボクセルの数をカウントしていきます。

追加のフラグを使う場合、アルゴリズムは粒子に属する各ピクセルまたはボクセルについて、追加の量を計算する必要があります。

単に粒子をマスクすることが目的であれば、より効率的なアプローチとして ImageSeedFill コマンドを使う方法があります。

この演算も同様に粒子を追跡しますが、関係のない粒子のプロパティを計算する処理時間を費やすことはありません。

また、ImageSeedFill には、入力ウェーブがバイナリである必要がないという利点もあり、初期の閾値処理にかかる時間を節約できます。

実際、ImageAnalyzeParticles では利用できない適応型/ファジー機能を使っている場合、はるかに優れた結果が得られる可能性があります。

例

グレースケール画像 (プロブ) を適切なバイナリ入力に変換します。

(Image Processing Tutorial を開いて行います)

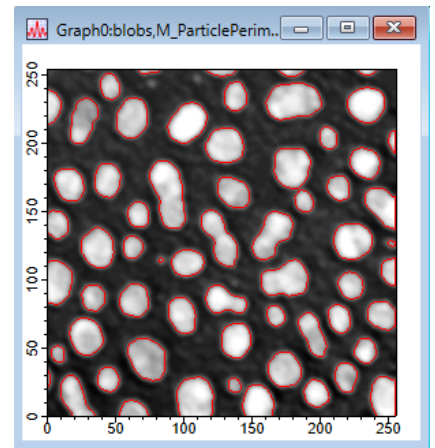
```
ImageThreshold/M=4/Q/I root:images:blobs
```

閾値処理されたブロブ画像の統計情報を取得し、粒子の輪郭に対応する画像マスクウェーブを作成します。

```
ImageAnalyzeParticles/M=1 stats M_ImageThresh
```

ブロブの画像に、輪郭画像の赤いオーバーレイを重ねて表示します。

```
NewImage/F root:images:blobs;  
    AppendImage M_ParticlePerimeter  
ModifyImage M_ParticlePerimeter explicit=1,  
    eval={0,65000,0,0}
```



参照

コマンド ImageThreshold, ImageGenerateROIMask, ImageSeedFill, ModifyImage
ヘルプ Image Processing