W_xmin, W_xmax, W_ymin, W_ymax

Contain a single point for each particle defining an inscribing rectangular box with axes along the X and Y directions.

One of the following waves can be created depending on the /M specification. The waves are designed to be used as an overlay on the original image (using the explicit=1 mode of **ModifyImage**). **Note**: the additional time required to create these waves is negligible compared with the time it takes to generate the stats data.

M_ParticlePerimeter

Masking image of particle boundaries. It is an unsigned char wave that contains 0 values for the object boundaries and 64 for all other points.

M_ParticleArea

Masking image of the area occupied by the particles. It is an unsigned char wave containing 0 values for the object boundaries and 64 for all other points. It is also different from the input image in that particles smaller than the minimum size, specified by /A, are absent.

M_Particle

Image of both the area and the boundary of the particles. It is an unsigned char wave that contains the value 16 for object area, the value 18 for the object boundaries and the value 64 for all other points.

M_rawMoments

Contains five columns. The first column is the raw sum of the x values for each particle, and the second column contains the sum of the y values. To obtain the average or “center” of a particle divide these values by the corresponding area. The third column contains the sum of x^2, the fourth column the sum of y^2, and the fifth column the sum of x*y. The entries of this wave are used in calculating a fit to an ellipse (using the /E flag).

When *imageMatrix* is a 3D wave, the different results are packed into a single 2D wave M_3DParticleInfo, which consists of one row and 11 columns for each particle. Columns are arranged in the following order: minRow, maxRow, minCol, maxCol, minLayer, maxLayer, xSeed, ySeed, zSeed, volume, and area. Use `Edit M_3DParticleInfo.ld` to display the results in a table with dimension labels describing the different columns.

Examples

Convert a grayscale image (blobs) into a proper binary input:

```
ImageThreshold/M=4/Q/I blobs
```

Get the statistics on the thresholded image of blobs and create an image mask output wave for the perimeter of the particles:

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

Display an image of the blobs with a red overlay of the perimeter image:

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

See Also

The **ImageThreshold**, **ImageGenerateROIMask**, **ImageSeedFill**, and **ModifyImage** operations. For more usage details see **Particle Analysis** on page III-375.

ImageBlend

`ImageBlend [/A=`*`alpha`*` /W=`*`alphaWave`*`] `*`srcWaveA`*`, `*`srcWaveB`*` [, `*`destWave`*`]`

The ImageBlend operation takes two RGB images (3D waves) in *srcWaveA* and *srcWaveB* and computes the alpha blending so that

destWave = *srcWaveA* * (1 - *alpha*) + *srcWaveB* * *alpha*

for each color component. If *destWave* is not specified or does not already exist, the result is saved in the current data folder in the wave M_alphaBlend.

The source and destination waves must be of the same data types and the same dimensions. The *alphaWave*, if used, must be a single precision (SP) float wave and it must have the same number of rows and columns as the source waves.