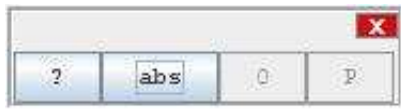


Operation: abs



Author:

Uses the method abs of the class ImageProcessor from ImageJ.

Example



Description

Replace each pixel value of the image with its absolute value.

Options

The operation has no options.

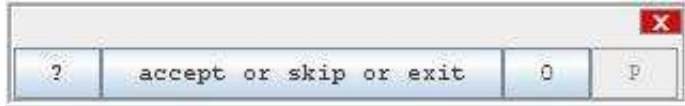
Parameter

The only parameter is the input image.

Results

The result image is a copy of the input image with each pixel value replaced by its absolute value.

Operation: accept or skip or exit



Author:

Volker Backer

Example

count objects			
?	foreach image in list do	0	P
?	open image	0	P
?	show image	0	P
?	std dev around median threshold	0	P
?	invert image	0	P
?	find objects	0	P
?	objects to point selection	0	P
?	accept or skip or exit	0	P
?	measure	0	P
?	report point measurements	0	P
?	close image	0	P
?	foreach image end	0	P

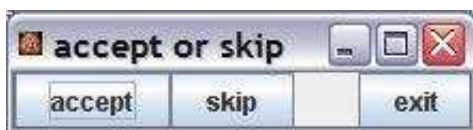


image	number of objects	X	Y	Z	Value	folder
A4 dapi 1.tif	10	296,7	236,8	0	154,3	C:\Docum

Description

Pause the application until the user presses one of the buttons accept, skip or exit. If accept is pressed the application continues with the next operation. If skip is pressed the application skips a configurable number of operations. If exit is pressed the application is stopped.

Options



number of operations: The number of operations that will be skipped when the button skip is pressed.

Parameter

The operation has no parameters.

Results

The operation has no result.

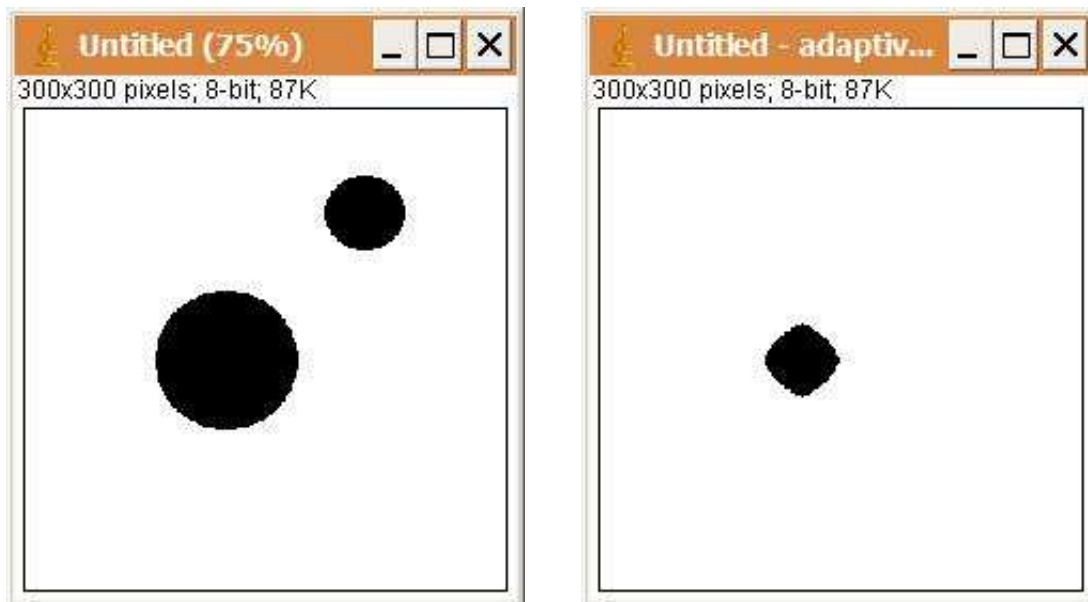
Operation: adaptive erode



Author:

Volker Bäcker

Example



Description

The operation erodes a binary image until the first object totally disappears.

Options

The operation has no options.

Parameter

The only parameter is the input image. The input image must be a binary image.

Results

The only result is the result image.

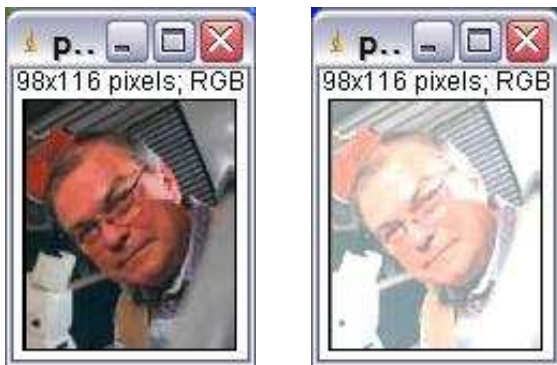
Operation: add



Author:

Uses the method add of the class ImageProcessor from ImageJ.

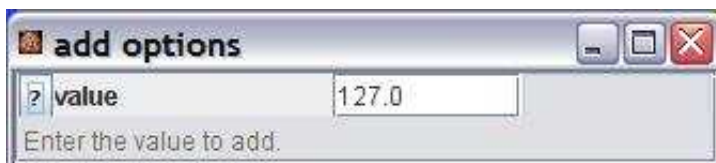
Example



Description

The operation adds a constant value to the value of each pixel in the image.

Options



Description

value: The value that will be added to the value of each pixel.

Parameter

The only parameter is the input image.

Results

The only result is the result image.

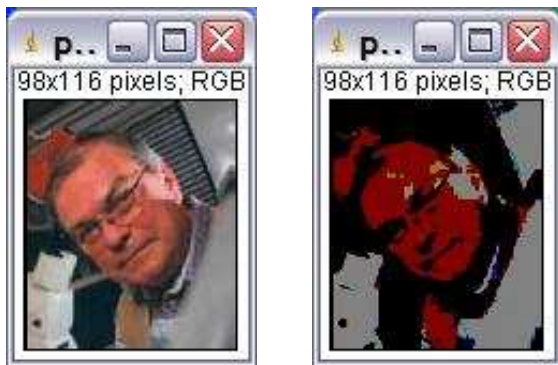
Operation: and



Author:

Uses the method and of the class ImageProcessor from ImageJ.

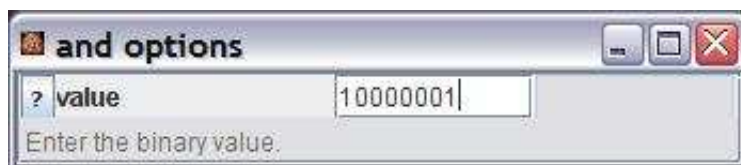
Example



Description

The operation replaces the value of each pixel in the image with the result of a binary and of the value and the input value.

Options



value: The value in binary form.

Parameter

The only parameter is the input image.

Results

The only result is the result image.

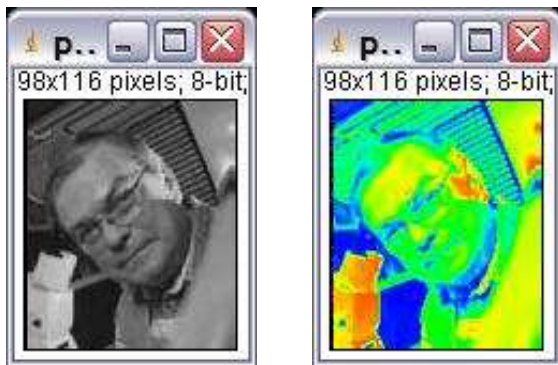
Operation: apply lut



Author:

The operation uses the LutLoader from ImageJ.

Example



Description

The operation applies a lookup table (lut) to a greyscale image. A lookup table maps intensity values to colors. An image with a lookup-table is displayed with the colors defined by the lookup-table.

Options



lookup tables: Select a lookup table from a list of available lookup tables. Some of the lookup tables are intern, others can be found in the folder `_lut`.

Parameter

The only parameter is the input image. The input image must not be an RGB image.

Results

The only result is the result image.

Operation: auto threshold



Author:

The operation uses the methods `threshold` and `getAutoThreshold` from the imageJ class `ImageProcessor`.

Example



Description

The threshold operation sets all intensities above the threshold value to the maximum (255) and all below to the minimum (0). The threshold value is computed iteratively. The iteration stops when the threshold is above the composite average of the two classes. See imageJ documentation for details.

Options

none

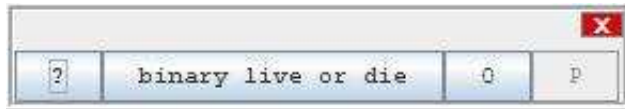
Parameter

The input image must be a 8bit greyscale image.

Results

The result is a 8bit greyscale image that contains only the intensities 0 and 255.

Operation: binary live or die



Author:

Volker Bäcker

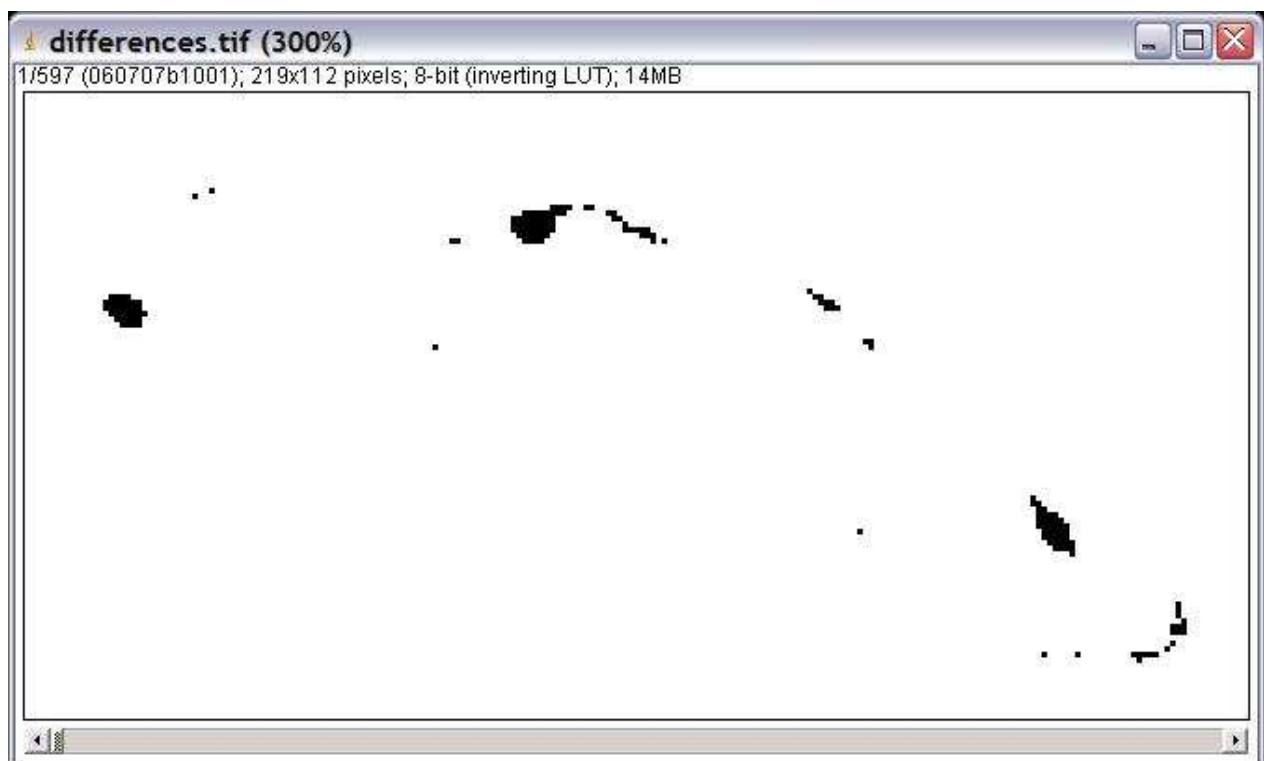
The idea comes from:

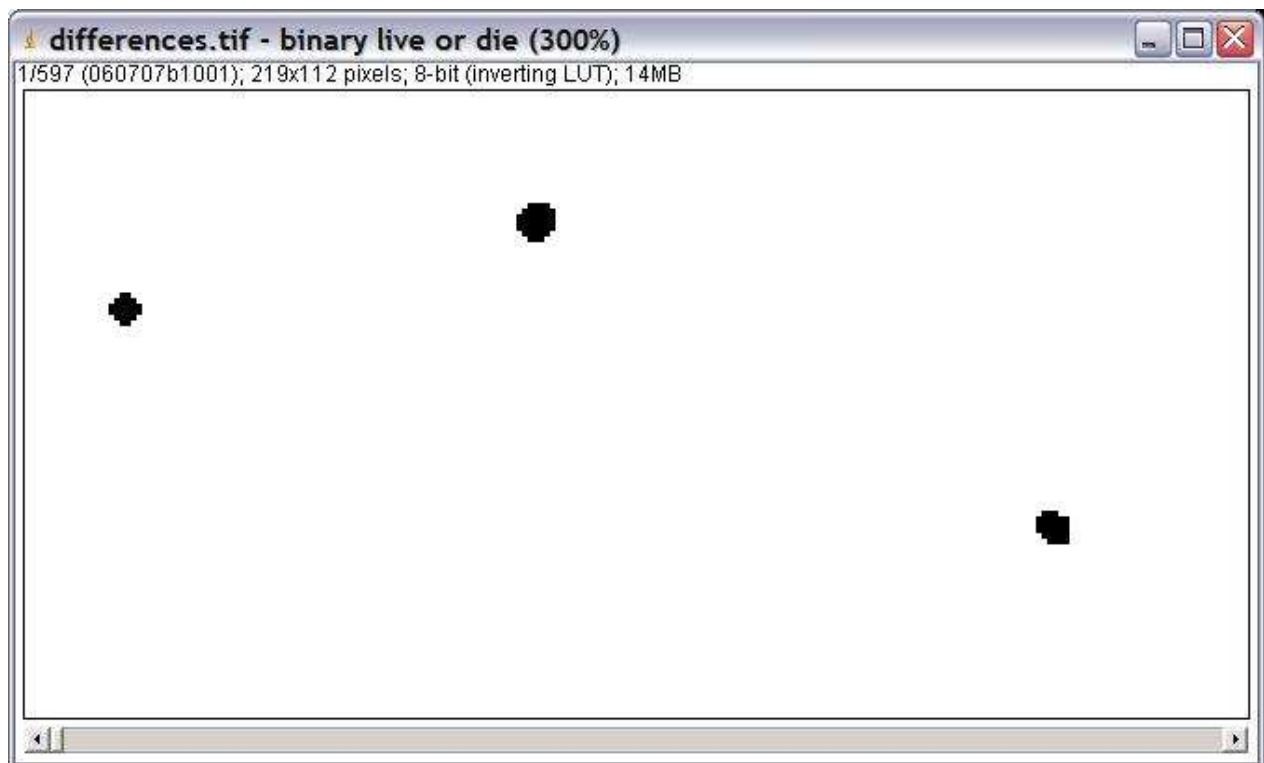
Leighton T. Izu, W. Gil Wier and C. William Balke

Theoretical Analysis of the Ca21 Spark Amplitude Distribution

Biophysical Journal Volume 75 September 1998 1144–1162

Example

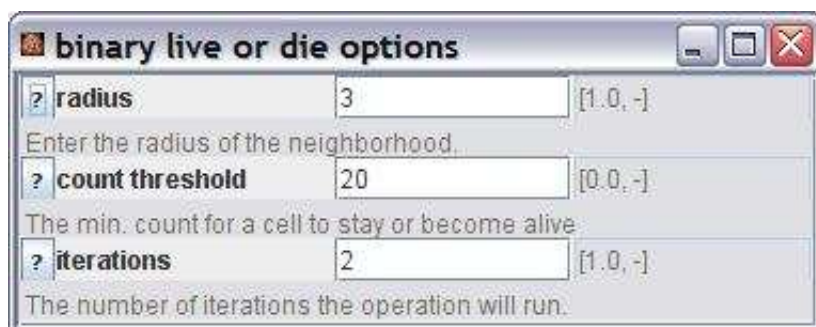




Description

The operation implements a totalistic cellular automaton rule. It runs a given number of iterations. In each iteration each pixel is replaced by 255 if the number of pixels in his neighborhood is not smaller than the threshold and by 0 otherwise.

Options



radius: The radius of the neighborhood. A radius 3 means for example that the neighborhood has a size of 7x7 pixel.

count threshold: The minimum number of pixels in the neighborhood that must be alive (255) for the central pixel to become or stay alive.

iterations: The number of iterations the operation runs. In each iteration the values of the preceding iteration are used to compute the new result.

Parameter

The only parameter is the input image. It should be an 8-bit mask image, containing only the intensities 0 and 255.

Results

The result is an 8-bit mask image.

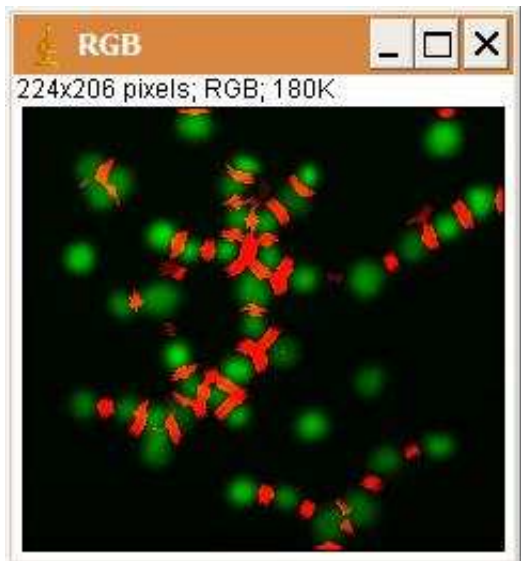
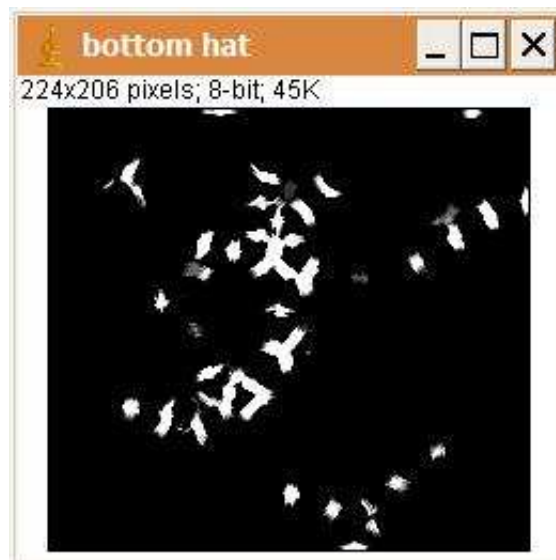
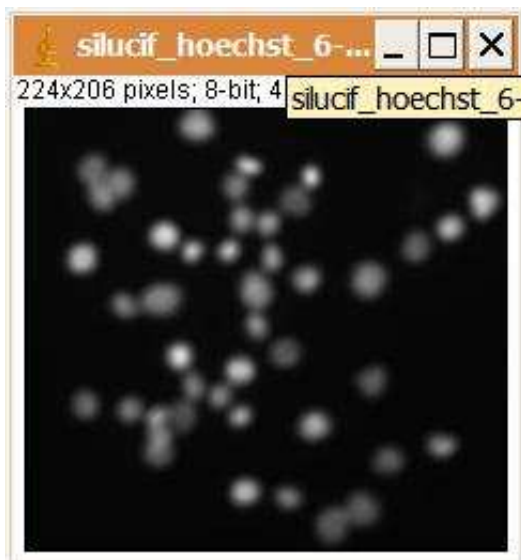
Operation: bottom hat gray



Author:

The operation calls the Grayscale Morphology plugin by Dimitar Prodanov.

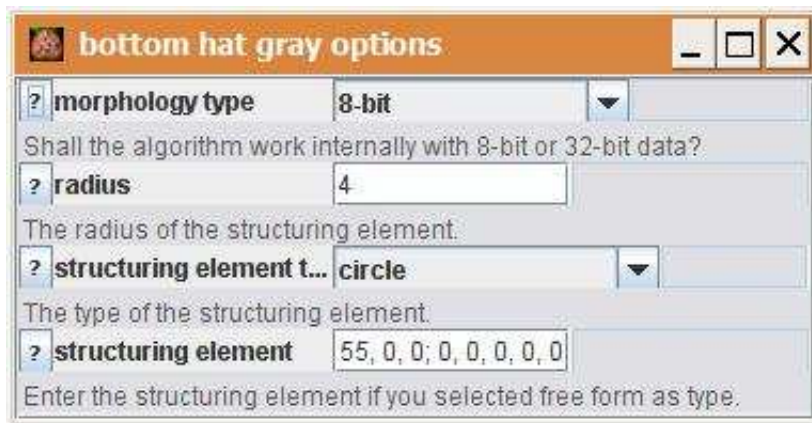
Example



Description

The operation detects regions that connect objects. This is done by calculating the difference of the input image and the result of the close operation.

Options



morphology type: Choose whether the 8-bit or the 32-bit morphology plugins are used.

radius: The radius of the structuring element.

structuring element type: Select the form of the structuring element. If free form is selected the structuring element can be entered manually.

structuring element: Enter the structuring element directly. This is used when free form is selected in the field structuring element type.

Parameter

The only parameter is the input image.

Results

The only result is the result image. The result image is a 32-bit or an 8-bit image, depending on the morphology type used.

Operation: clear image



Author:

The operation uses the class Filler from ImageJ.

Example



Description

The operation copies the image or the current selection and fills the result with the background color.

Options

The operation has no options.

Parameter

The only parameter is the input image.

Results

The result is a copy of the input image or of the current selection, filled with the background color.

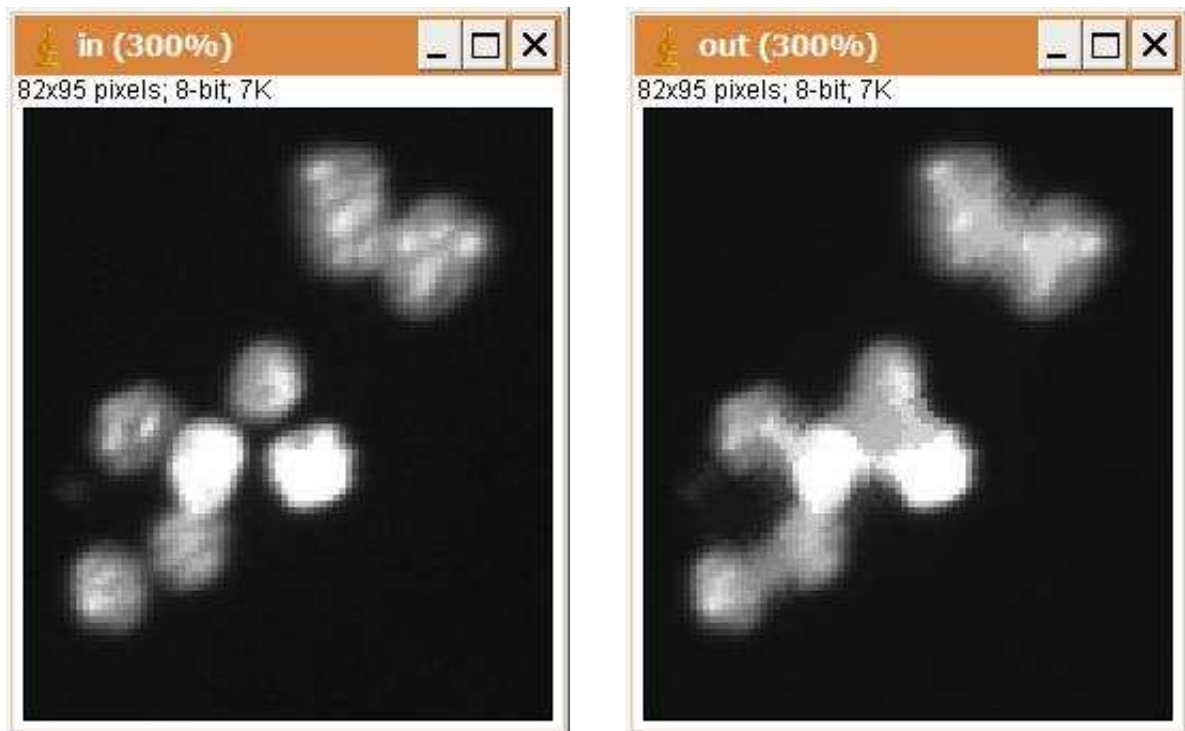
Operation: close gray



Author:

The operation calls the Grayscale Morphology plugin by Dimitar Prodanov.

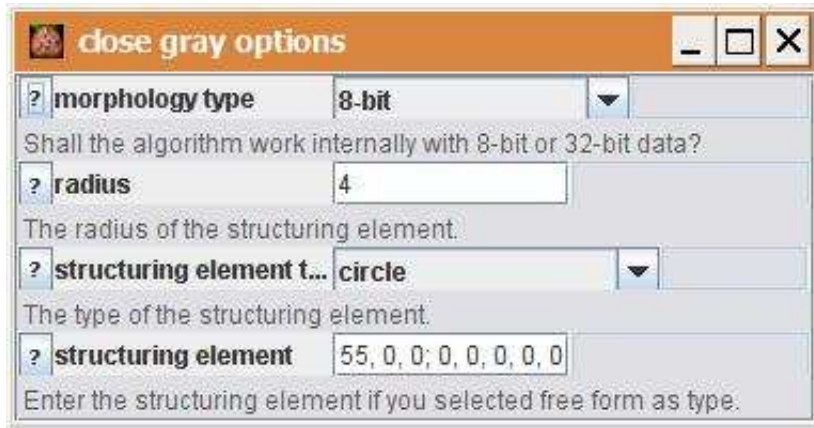
Example



Description

The operation closes the greyscale image with a structuring element. It shrinks background holes in foreground regions and keeps the original outer shape. It keeps background holes that have similar shape as the structuring element or that can completely contain the structuring element and eliminates others. A close is a dilation followed by an erosion.

Options



morphology type: Choose whether the 8-bit or the 32-bit morphology plugins are used.

radius: The radius of the structuring element.

structuring element type: Select the form of the structuring element. If free form is selected the structuring element can be entered manually.

structuring element: Enter the structuring element directly. This is used when free form is selected in the field structuring element type.

Parameter

The only parameter is the input image.

Results

The only result is the result image. The result image is a 32-bit or an 8-bit image, depending on the morphology type used.

Operation: close image



Author:

Volker Bäcker

Example



Description

Close the window of the input image.

Options

The operation has no options.

Parameter

The only parameter is the input image.

Results

The operation has no results.

Operation: close session



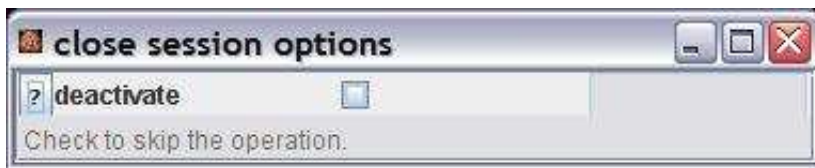
Author:

Volker Bäcker

Description

Closes the session of the user. On window systems the name of a command or script that closes the session can be configured in `mri_cia_config.txt` under the key "logoff command". If the key doesn't exist the command `shutdown -l -f` is used. On unix the command `/bin/bash logoff` is used. On Mac a script with the name `logout.scpt` is called. If the option deactivate is checked the operation is ignored.

Options



deactivate: If the option is checked, the operation does nothing.

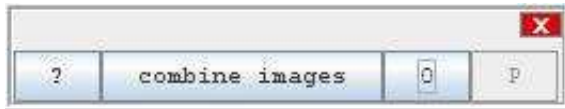
Parameter

The operation has no parameters.

Results

The operations has no results.

Operation: combine images

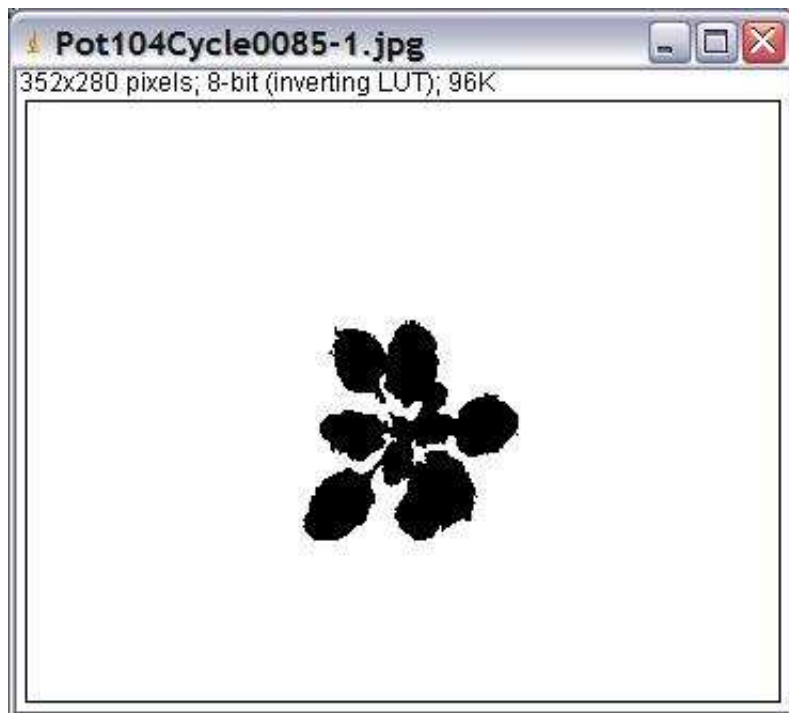


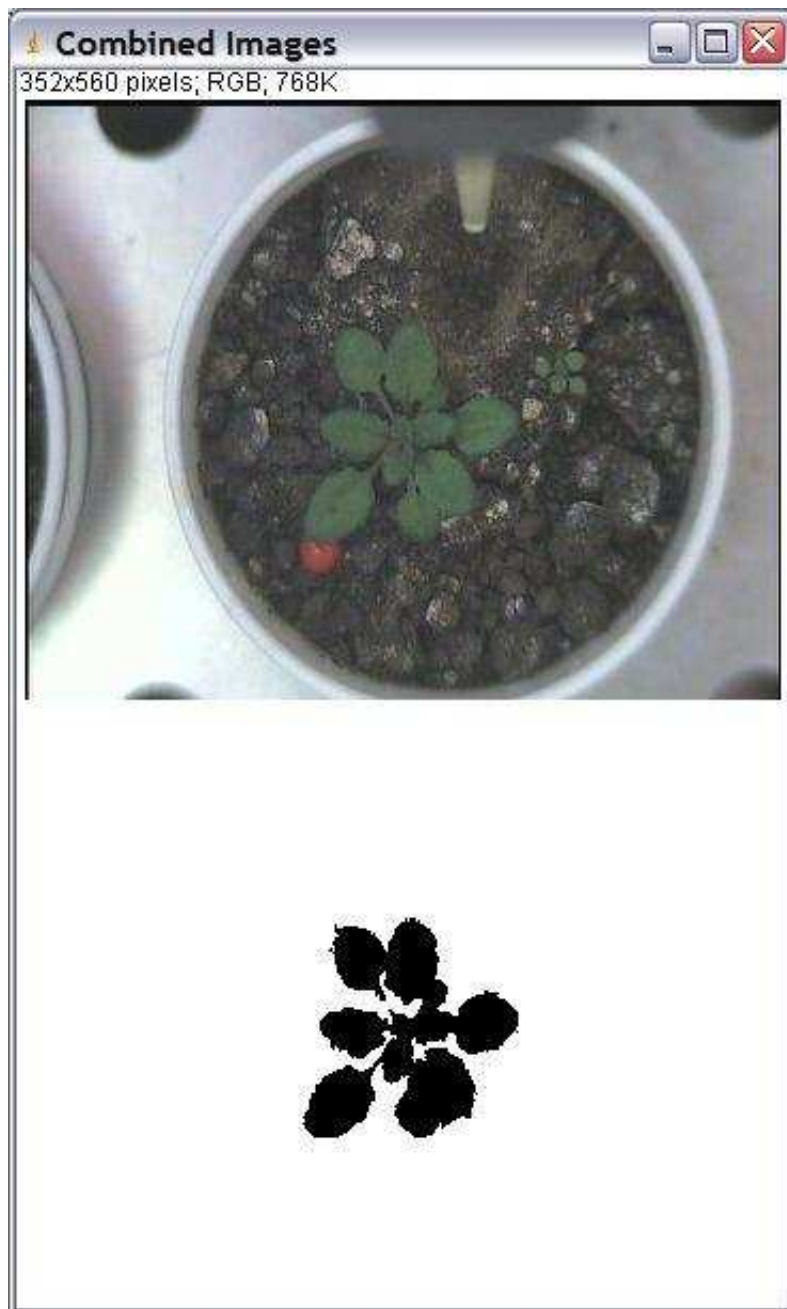
Author:

Uses the Stack_Combiner plugin from Wayne Rasband.

Example







Description

The operation creates a new image containing the two input images either one above the other or one next to the other.

Options



combine vertically: If checked the input images will be one above the other otherwise the input images will be next to each other.

Parameter

The two input images. Both images must be of the same type.

Results

The only result is the result image.

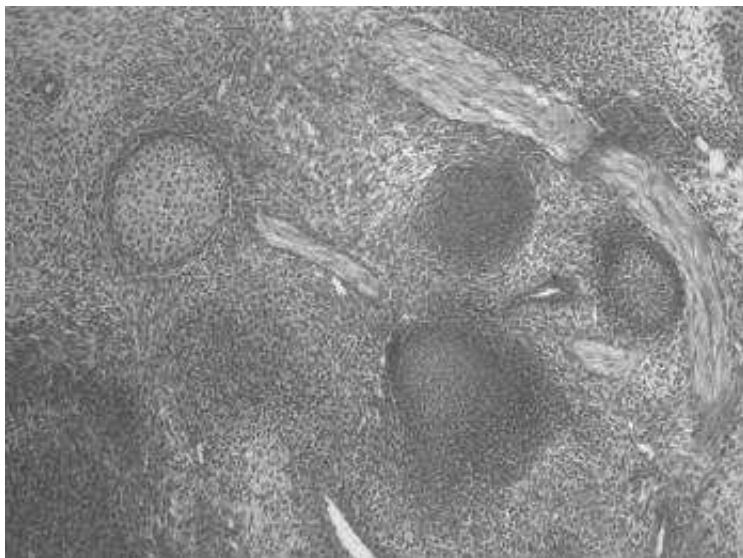
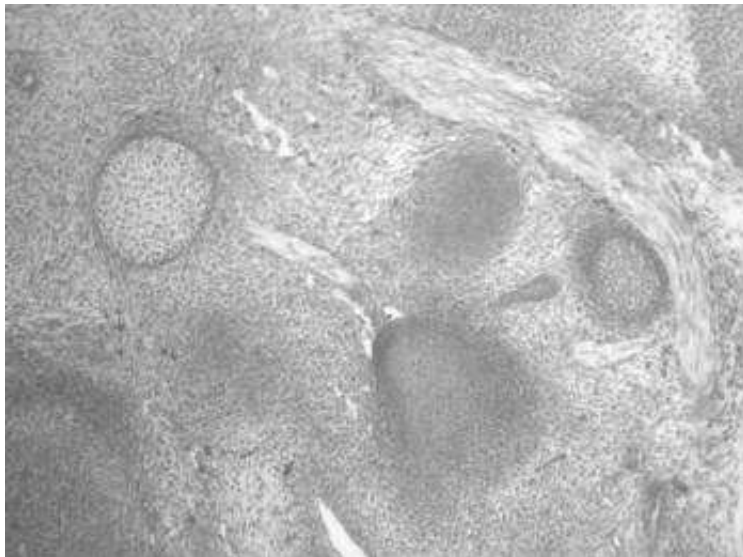
Operation: compute difference

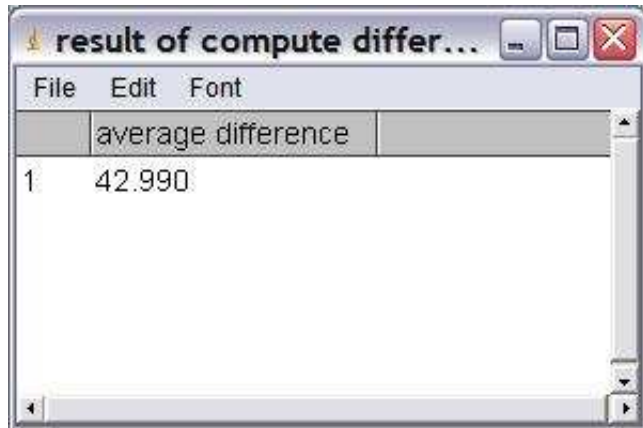


Author:

Volker Backer

Example:





Description

Calculate the average difference between two images. The two images must have the same size. The images are converted to 8-bit images before the computation is done. Pixels that are zero in one of the images are skipped. If you run the operation directly (not from an application), the first image is the active image. To give the user the possibility to select the second image the operation opens the wait button and stops until the user presses it.

Options

The operation has no options.

Parameter

1. (ImagePlus) first image: a 2-dimensional image
 2. (ImagePlus) second image: a 2-dimensional image
- The two images must have the same size.

Results

1. (ResultsTable) difference: a results table containing the average difference between the two images.

Operation: compute moments



Author:

Volker Bäcker

Example



	m00	m10	m01	m20	m02	m11	m30
1	511085	-8.845973E-10	-3.137302E-9	2.29334224E8	1.2841252E8	-9318851.525	5.9897146E8

Description

Computes the central moments up to the third order and 7 features build from the central moments that are invariant against translation and rotation. Computes either the moments for the whole image or within the rectangular selection.

You can find the definition of the features in:

Seminarband der Projektgruppe 294 - BAMBUS - Baukasten für die Analyse und Modellierung von Bildobjekten aufgrund unscharfen Wissens, University Dortmund, Internal Reports, 1996, page 72 (or in this document: mustererkennung.pdf). They have originally been taken from:

Heinrich Niemann, Klassifikation von Mustern, Springer-Verlag Berlin Heidelberg, 1983

Options

The operation has no options.

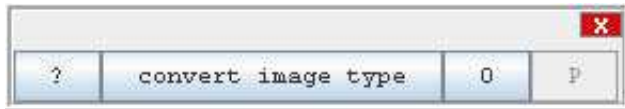
Parameter

The only parameter is the input image. If there is a ROI on the image the bounding box of the ROI is taken into account.

Results

A results table with the central moments up to the third order and the 7 rotation and translation invariant features calculated from them.

Operation: convert image type



Author:

The operation uses the Class Converter from ImageJ.

Description

Converts the image type of the input image to the type selected in the operators options.

Depending on the type of the input image not all conversions are possible. See ImageJ documentation, menu "image" for details.

Supported Conversions:

8-bit -> 16-bit*

8-bit -> 32-bit*

8-bit -> RGB Color*

16-bit -> 8-bit*

16-bit -> 32-bit*

16-bit -> RGB Color*

32-bit -> 8-bit*

32-bit -> 16-bit

32-bit -> RGB Color*

8-bit Color -> 8-bit (grayscale)*

8-bit Color -> RGB Color

RGB Color -> 8-bit (grayscale)*

RGB Color -> 8-bit Color*

RGB Color -> RGB Stack

RGB Color -> HSB Stack

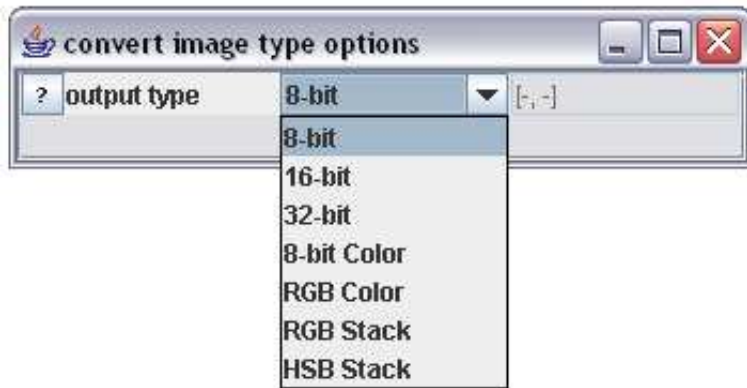
RGB Stack -> RGB Color

HSB Stack -> RGB Color

* works with stacks

Options

output type: choose the type of the result image



Parameter

The only parameter is the input image.

Results

The only result is the converted image.

Operation: convolve



Author:

The operation uses the Class Convolver from ImageJ.

Example

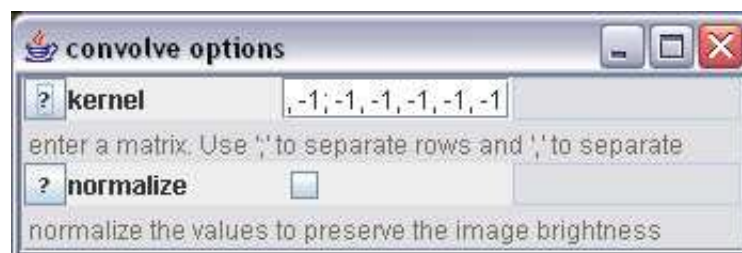


The image has been convolved with the default kernel of size 5x5 with value 24 for the central position and -1 for all others.

Description

The kernel is applied to each pixel of the image. The new value of each pixel is the sum of the values of the neighbors each multiplied by the corresponding coefficient in the kernel. See ImageJ documentation, menu "process>>filters" for details.

Options



kernel: The number of rows and columns of the kernel must be odd. Each row must have the same number of elements.

normalize: If normalize is chosen the values in the matrix are divided by their sum to preserve the brightness of the image.

Parameter

The only parameter is the input image.

Results

The only result is the convolved image.

Operation: copy image



Author:

Uses the class Duplicater from ImageJ.

Description

Creates a copy of the input image.

Options

The operation has no options

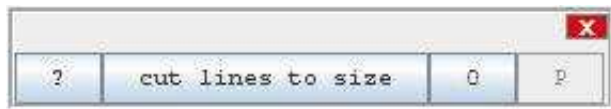
Parameter

The only parameter is the input image. The input image can be of any type.

Results

The result is a copy of the input image.

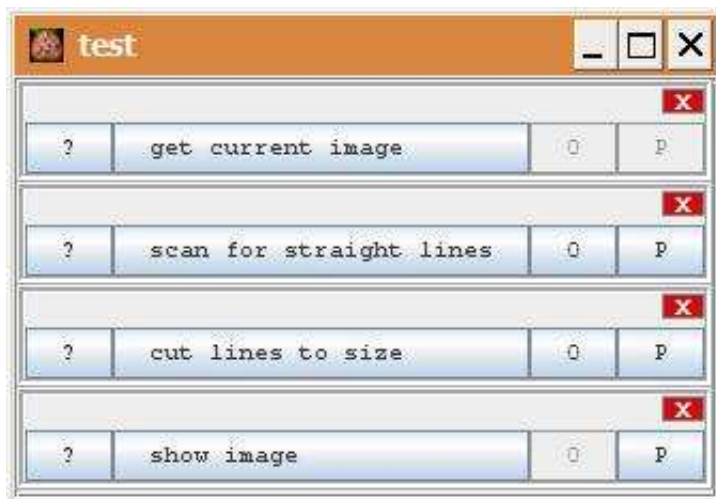
Operation: cut lines to size

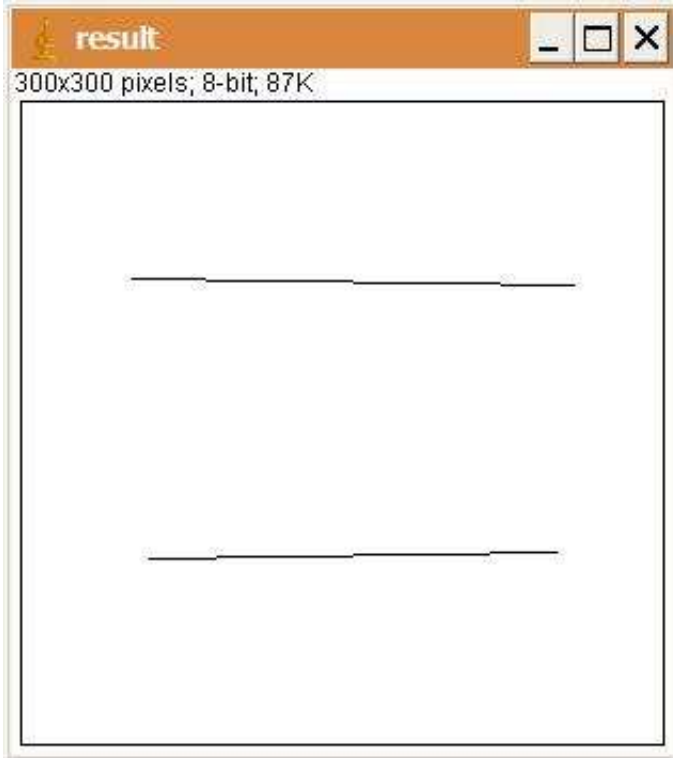
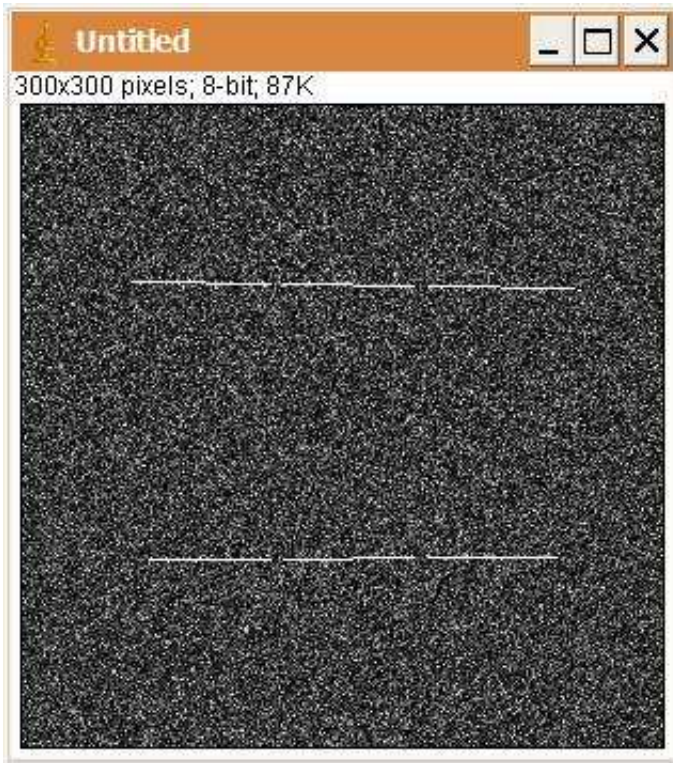


Author:

Volker Bäcker

Example

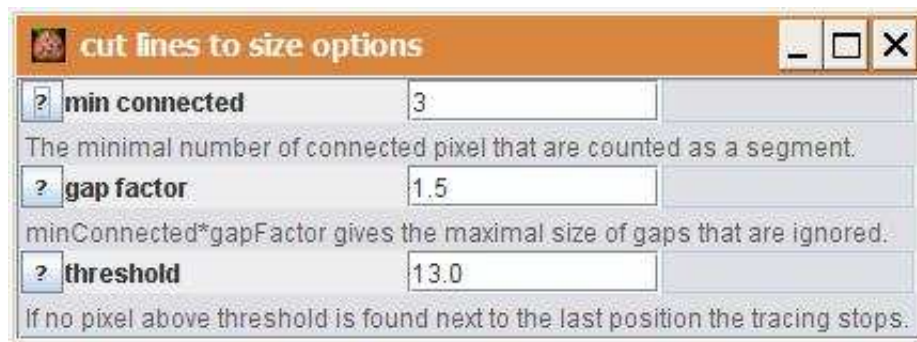




Description

The operation scans the image along the given lines. Stop when no point above threshold is found within a range of 6 pixels perpendicular to the line. Segments along a line are merged if the gap size is smaller than $\text{minConnected} * \text{gapFactor}$ and if the length of the segment is at least minConnected .

Options



min connected: The minimal number of connected pixel that are counted as a segment.

gap factor: gap factor multiplied with min connected gives the maximal length of gaps that will be ignored.

threshold: The intensity threshold for the tracing of the line. If no pixel above threshold is found near the last position the tracing stops.

Parameter

inputImage (ImagePlus): A greyscale input image.

lines (ArrayList<Line2D>): A list of lines.

Results

result (ImagePlus): An image representing the line segments found.

resultLines (ArrayList): A list of line segments.

Operation: despeckle



Author:

The operation uses the class RankFilter from ImageJ.

Example



The source image has been despeckled with the radius 1.

Description

<describe it here>

Options

<screenshot options>

<anchor><option1>: <short description>

<anchor><option2>: <short description>

<anchor><option3>: <short description>

...

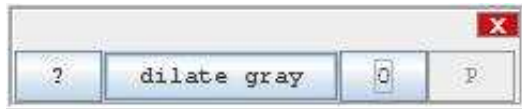
Parameter

<describe the parameters>

Results

<describe the results>

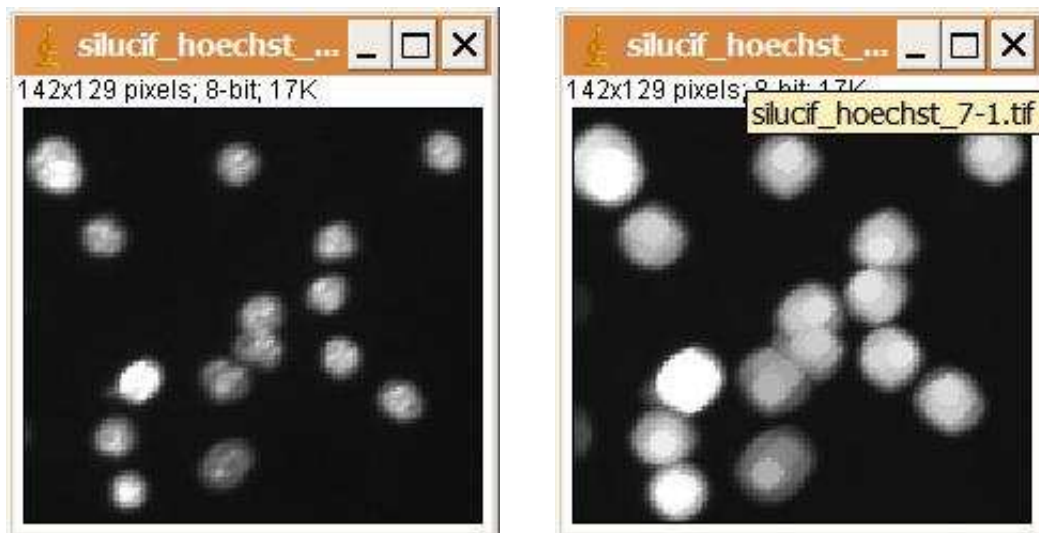
Operation: dilate gray



Author:

The operation calls the Grayscale Morphology plugin by Dimitar Prodanov.

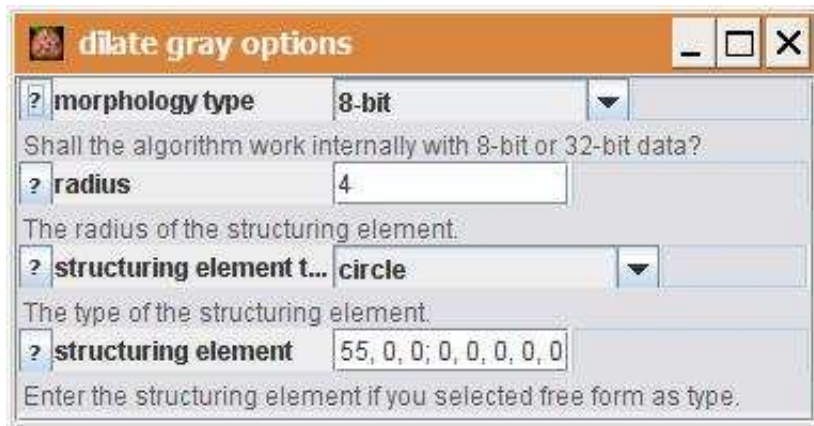
Example



Description

The operation enlarges foreground regions by testing if the intersection of the region and the structuring element is not empty.

Options



morphology type: Choose whether the 8-bit or the 32-bit morphology plugins are used.

radius: The radius of the structuring element.

structuring element type: Select the form of the structuring element. If free form is selected the structuring element can be entered manually.

structuring element: Enter the structuring element directly. This is used when free form is selected in the field structuring element type

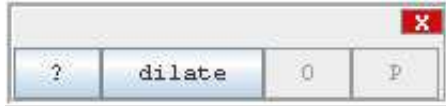
Parameter

The only parameter is the input image.

Results

The only result is the result image. The result image is a 32-bit or an 8-bit image, depending on the morphology type used.

Operation: dilate



Author:

The operation uses the method `dilate` from the ImageJ class `ImageProcessor`. See also ImageJ documentation.

Example



image

shows the result of the dilate operation for a binary and an rgb

Description

Replaces each pixel with the minimum (darkest) value in a 3x3 neighborhood, thus enlarging dark regions. This is the same as using the minimum operation with a radius 1.

Options

The operation has no options.

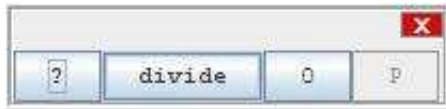
Parameter

The only parameter is the input image. It must be of type 8bit or rgb.

Results

The only result is the result image.

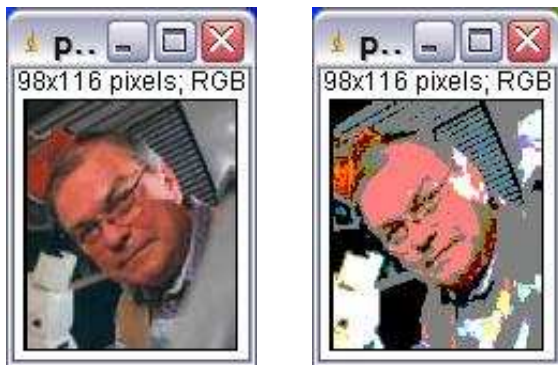
Operation: divide



Author:

Uses the method multiply of the class ImageProcessor from ImageJ.

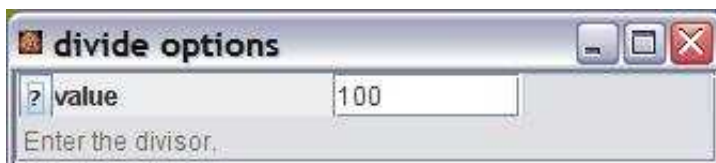
Example



Description

The operation divides the value of each pixel in the image by a constant value.

Options



value: The constant value. The value of each pixel in the image will be divided by this value.

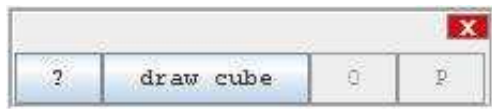
Parameter

The only parameter is the input image.

Results

The only result is the result image.

Operation: draw cube



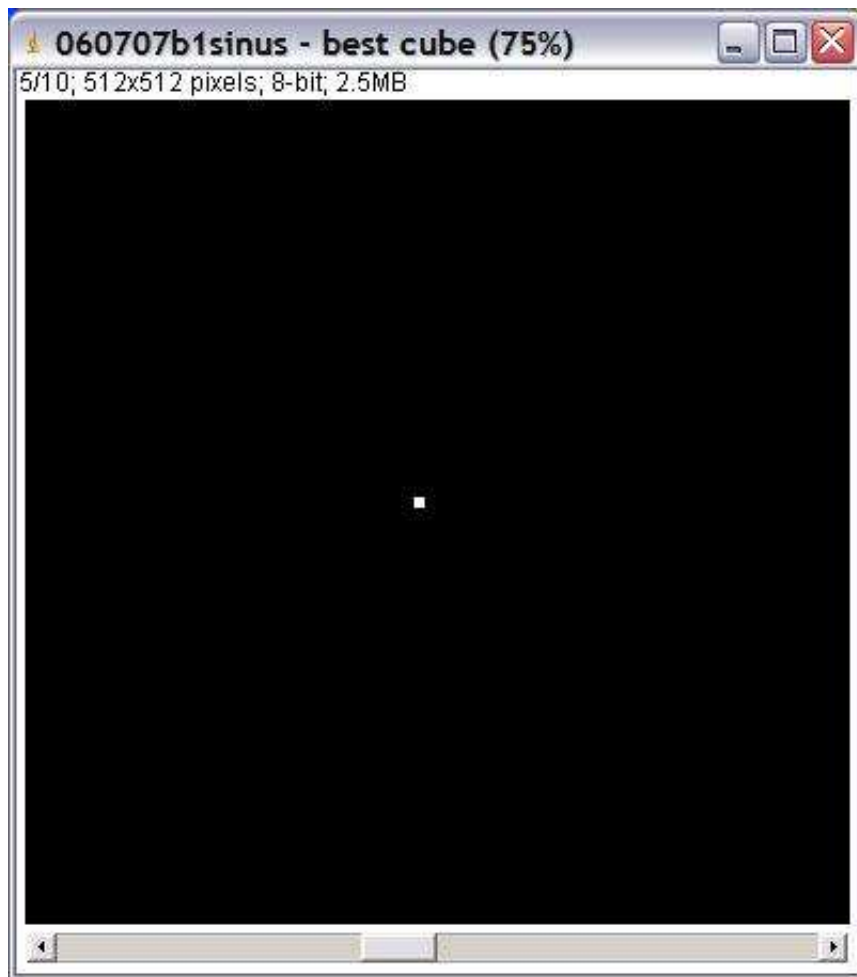
Author:

Volker Bäcker

Example:







Description

The input is a results table with 4 values in the columns x, y, z and radius. The result is a stack with a drawing of a solid cube of size $(2*\text{radius}+1 \times 2*\text{radius}+1 \times 2*\text{radius}+1)$, that has its center at x, y, z.

Options

The operation has no options.

Parameter

inputImage (ImagePlus): The input image. The result stack will be of the same dimensions as the input image.

cube (ResultsTable): A ResultsTable with one line of values and the columns x, y, z and radius.

Results

A drawing of the cube on an image stack of the same dimensions as the input image.

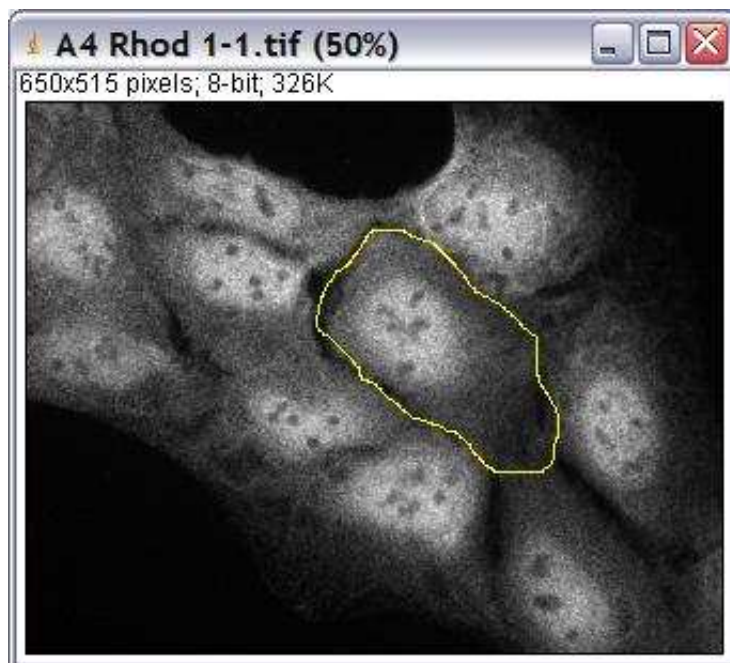
Operation: draw

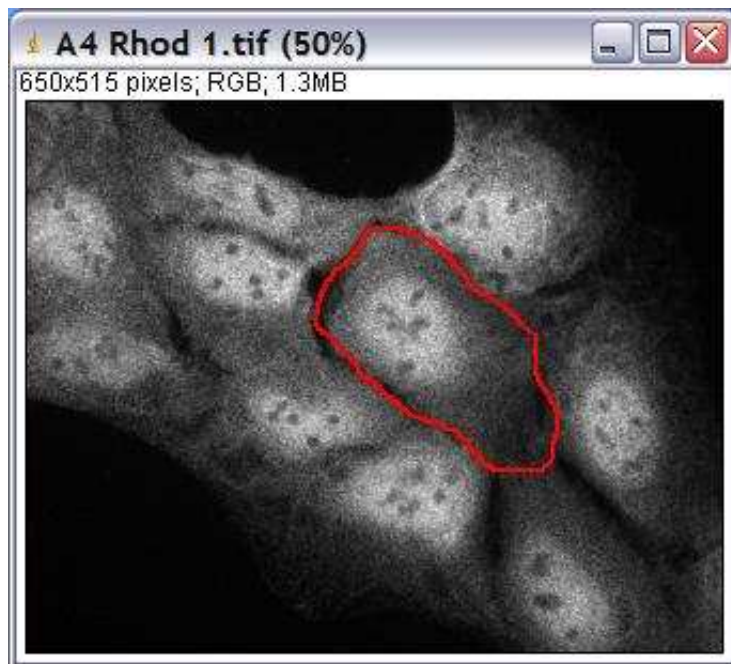


Author:

Uses the class Filler from ImageJ.

Example





Description

The operation draws the current selection on the input image. The current foreground color and the line width configured from ImageJ are used.

Options

The operation has no options.

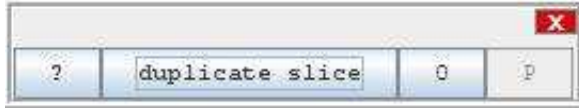
Parameter

The only parameter is the input image. There must be a selection (roi) on the input image.

Results

The result is a reference to the modified input image.

Operation: duplicate slice



Author:

<who wrote it>

Description

<describe it here>

Options

<screenshot options>

<anchor><option1>: <short description>

<anchor><option2>: <short description>

<anchor><option3>: <short description>

...

Parameter

<describe the parameters>

Results

<describe the results>

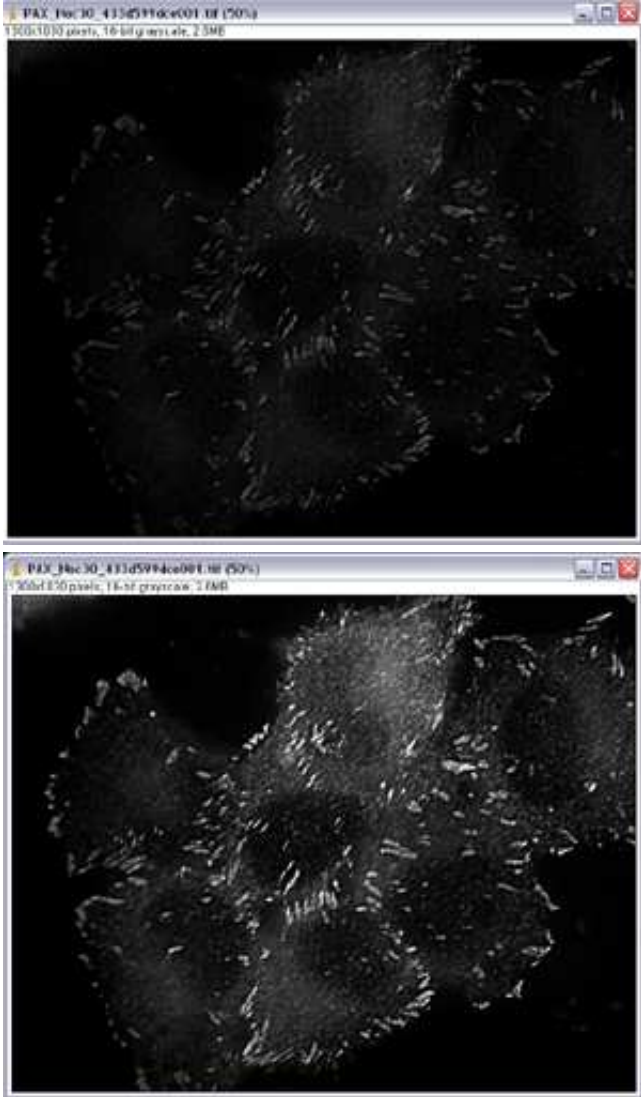
Operation: enhance contrast



Author:

The operation uses the Class ContrastEnhancer from ImageJ.

Example

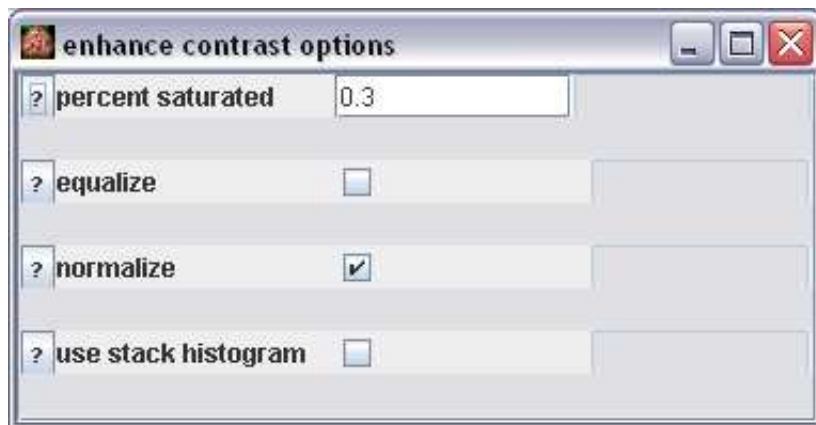


normalize with max. 0.3% of pixels saturated

Description

Enhances image contrast by using either histogram stretching or histogram equalization. See ImageJ documentation, menu "process" for details.

Options



percent saturated: the maximum percent of pixel that will be saturated after the normalization.

equalize: use histogram equalization (percent saturated will be ignored).

normalize: use histogram stretching.

use stack histogram: in case of volume image use the histogram of the volume instead of the histogram of the current slide.

Parameter

The only parameter is the input image.

Results

The only result is the enhanced image.

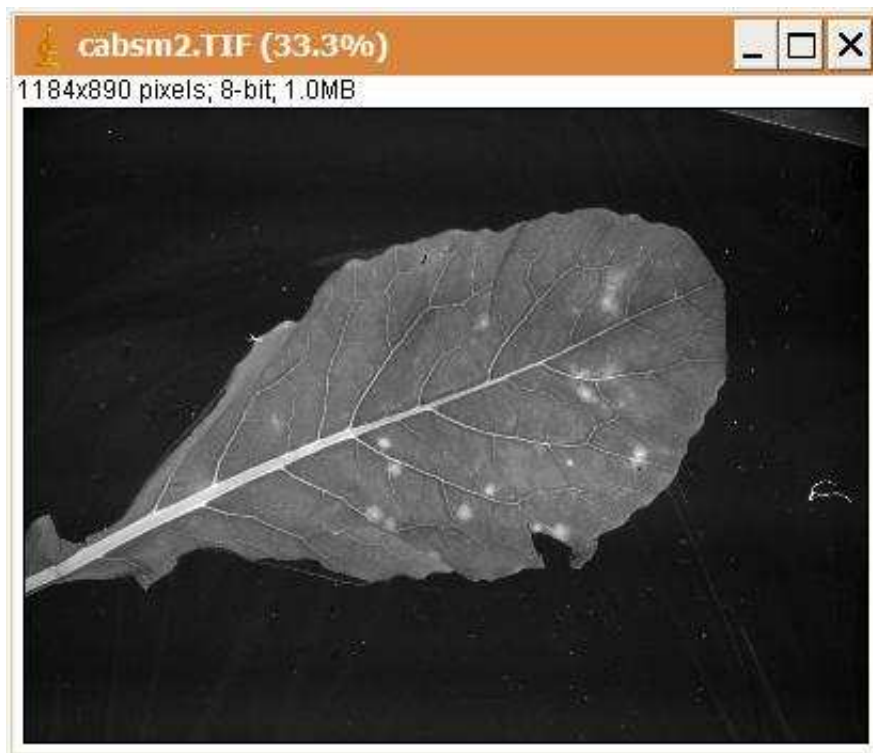
Operation: enhance spots

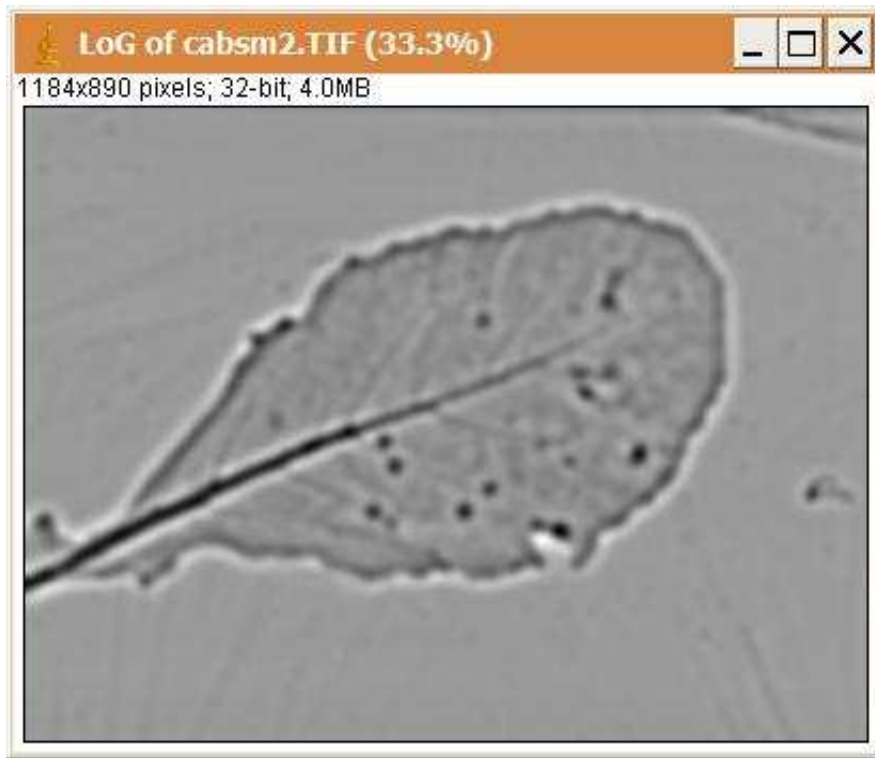


Author:

Calls the plugin LoG 3D by Daniel Sage.

Example





Description

A spot detector based on 3D Laplacian of Gaussian or Mexican Hat.

Options



sigma x: The standard deviation in x-direction.

sigma y: The standard deviation in y-direction.

sigma z: The standard deviation in z-direction.

volume: If volume is checked the stack is processed as a volume otherwise it is processed slice by slice.

Parameter

The only parameter is the input image. The input image must be a greyscale image.

Results

The only result is the result image. The result image is a 32-bit image.

Operation: entropy threshold



Author:

The operation uses the imageJ plugin "Maximum Entropy Threshold" written by Jerek Sacha.

Example



Description

A threshold value is computed from the histogram in a way that maximizes the inter-class entropy. Intensities above the threshold value are set to the maximum (255) and intensities below to the minimum (0).

Options

none

Parameter

The only parameter is the input image. This must be an 8bit greyscale image.

Results

The result image as an 8 bit greyscale image that contains only the intensities 0 and 255.

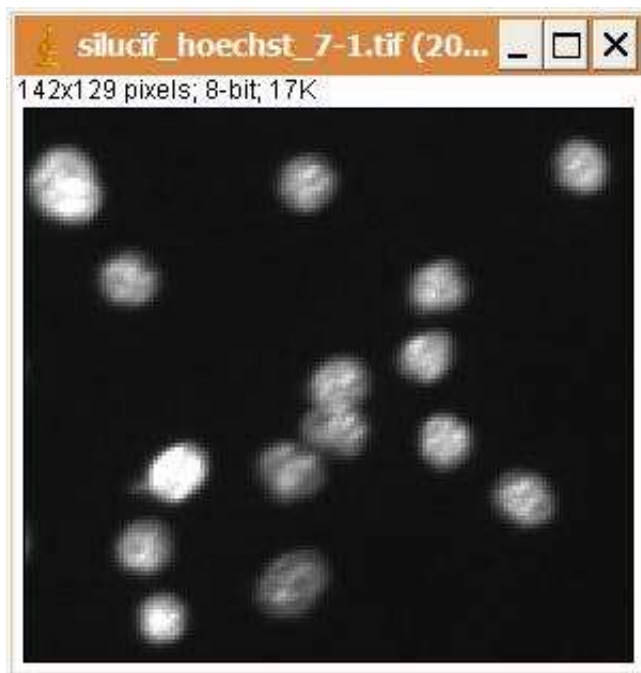
Operation: erode gray

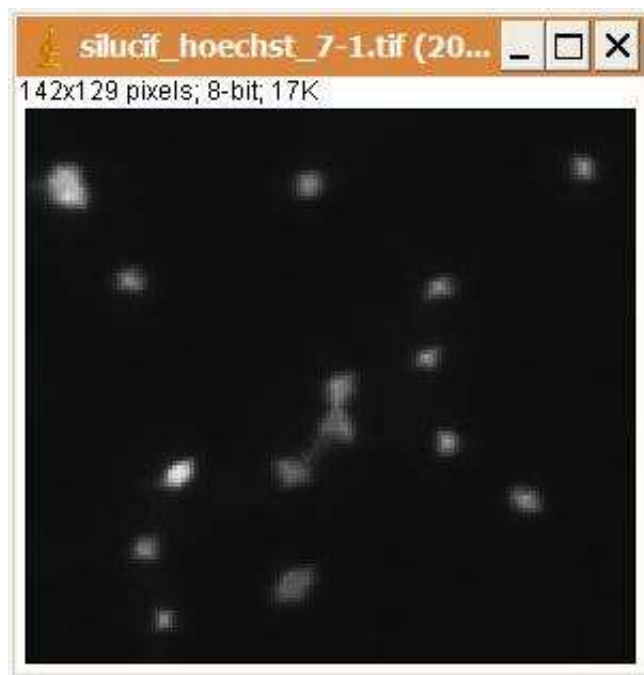


Author:

The operation calls the Grayscale Morphology plugin by Dimitar Prodanov.

Example

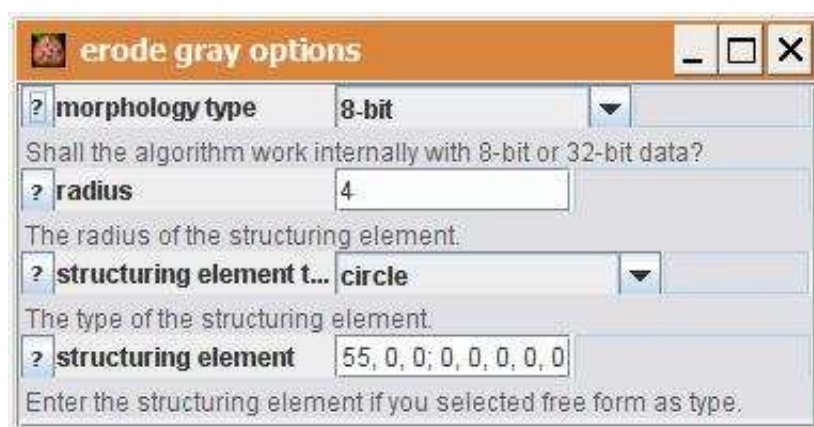




Description

The operation shrinks foreground regions by testing if the structuring element lies totally within the foreground for each position. The only result is the result image. The result image is a 32-bit or an 8-bit image, depending on the morphology type used.

Options



morphology type: Choose whether the 8-bit or the 32-bit morphology plugins are used.

radius: The radius of the structuring element.

structuring element type: Select the form of the structuring element. If free form is selected the structuring element can be entered manually.

structuring element: Enter the structuring element directly. This is used when free form is selected in the

field structuring element type.

Parameter

The only parameter is the input image.

Results

The only result is the result image. The result image is a 32-bit or an 8-bit image, depending on the morphology type used.

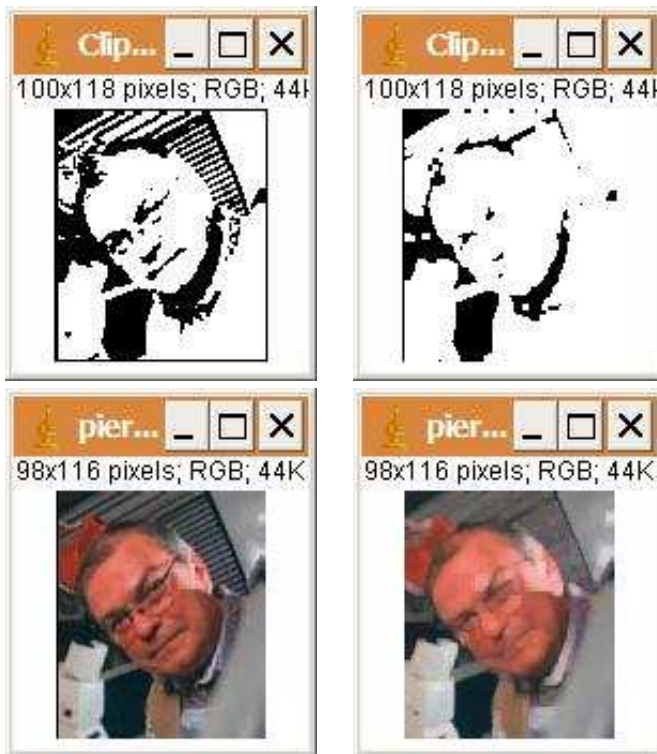
Operation: erode



Author:

The operation uses the method `erode` from the ImageJ class `ImageProcessor`. See also [ImageJ documentation](#).

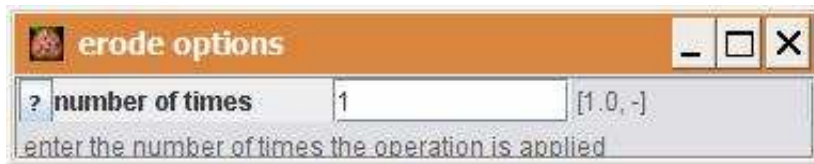
Example



Description

The operation replaces each pixel with the maximum (brightest) value in a 3x3 neighborhood, thus enlarging bright regions. This is the same as using the maximum operation with a radius 1. The operation can be applied repeatedly on the same image for a given number of times.

Options



number of times: The number of times the operation is applied to the input image.

Parameter

The only parameter is the input image.

Results

The onlt result is the result image.

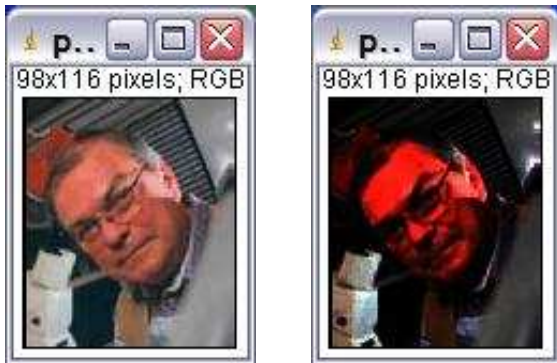
Operation: exp



Author:

Uses the method `exp` of the class `ImageProcessor` from `ImageJ`.

Example



Description

The operation replaces each intensity value i in the image with $\exp(i)$. Values will be scaled and truncated for image types other than 32-bit.

Options

The operation has no options.

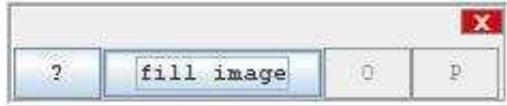
Parameter

The only parameter is the input image.

Results

The only result is the result image.

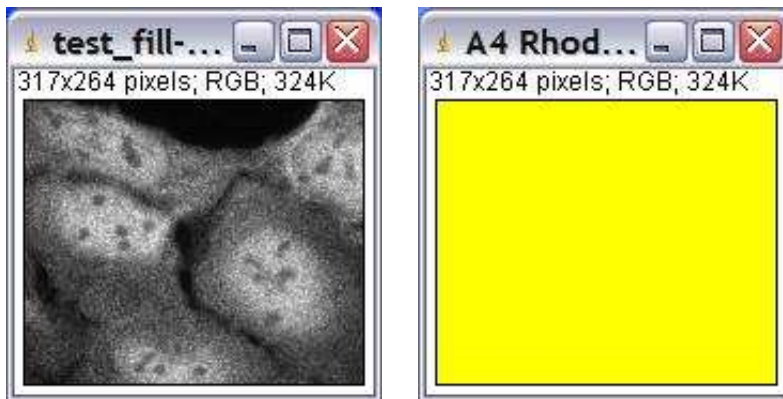
Operation: fill image



Author:

Uses the class Filler from ImageJ.

Example



Description

Answer a copy of the current image or selection filled with the current foreground color.

Options

The operation has no options.

Parameter

The only parameter is the input image.

Results

The result is always a copy of the input image or selection filled with the current foreground color.

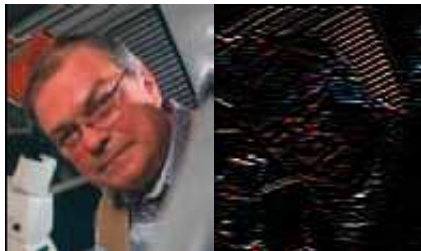
Operation: filter horizontal lines



Author:

The operation is a convolve operation with an appropriate default kernel.

Example

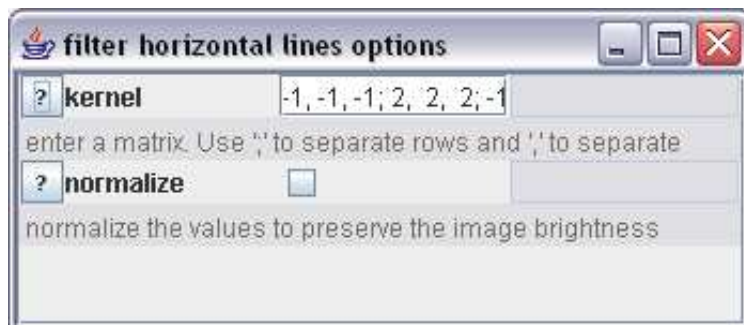


Description

This is a convolve operation with the kernel

-1	-1	-1
2	2	2
-1	-1	-1

Options



kernel:

The number of rows and columns of the kernel must be odd. Each row must have the same number of elements.

normalize:

If normalize is chosen the values in the matrix are divided by their sum to preserve the brightness of the image.

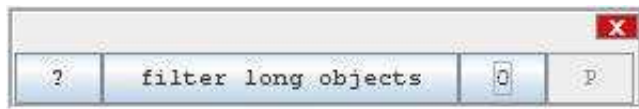
Parameter

The only parameter is the input image.

Results

The only result is the convolved image.

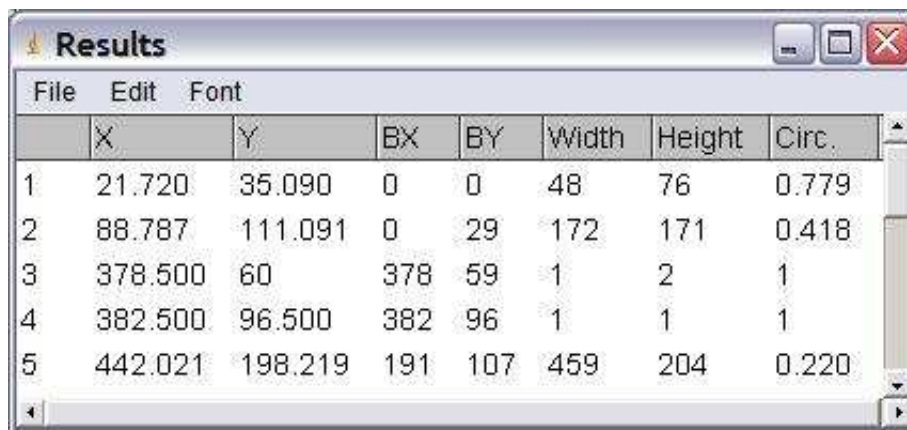
Operation: filter long objects



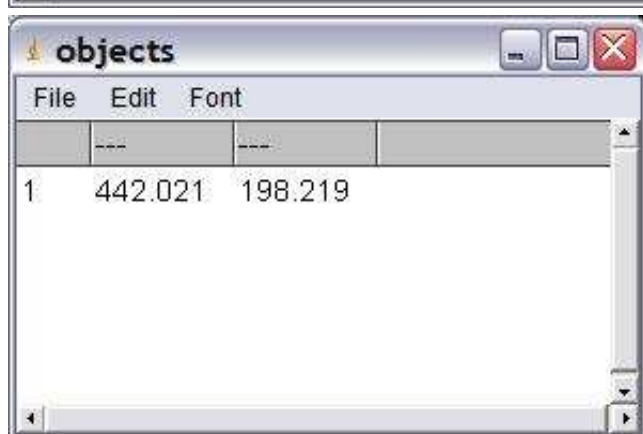
Author:

Volker Baecker

Example



	X	Y	BX	BY	Width	Height	Circ.
1	21.720	35.090	0	0	48	76	0.779
2	88.787	111.091	0	29	172	171	0.418
3	378.500	60	378	59	1	2	1
4	382.500	96.500	382	96	1	1	1
5	442.021	198.219	191	107	459	204	0.220

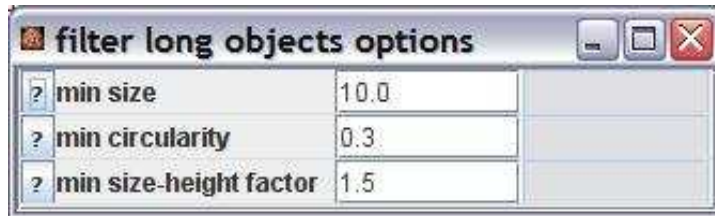


	X	Y
1	442.021	198.219

Description

The operation filters objects from a results table by width of the bounding rectangle, circularity and ratio of width and height of the bounding rectangle (height times factor > width). The input table must contain the measurements: centroid, bounding box and circularity. The result table will contain the centroids of the objects that fulfill the conditions.

Options



min size: The minimal width of the bounding box

min circularity: The minimal circularity.

min size-height factor: The width of the bounding box must at least be factor times longer than the height.

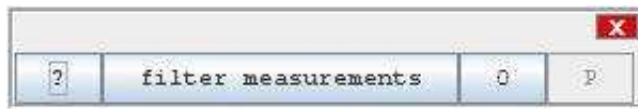
Parameter

The only parameter is the input results table.

Results

The output is the filtered results table.

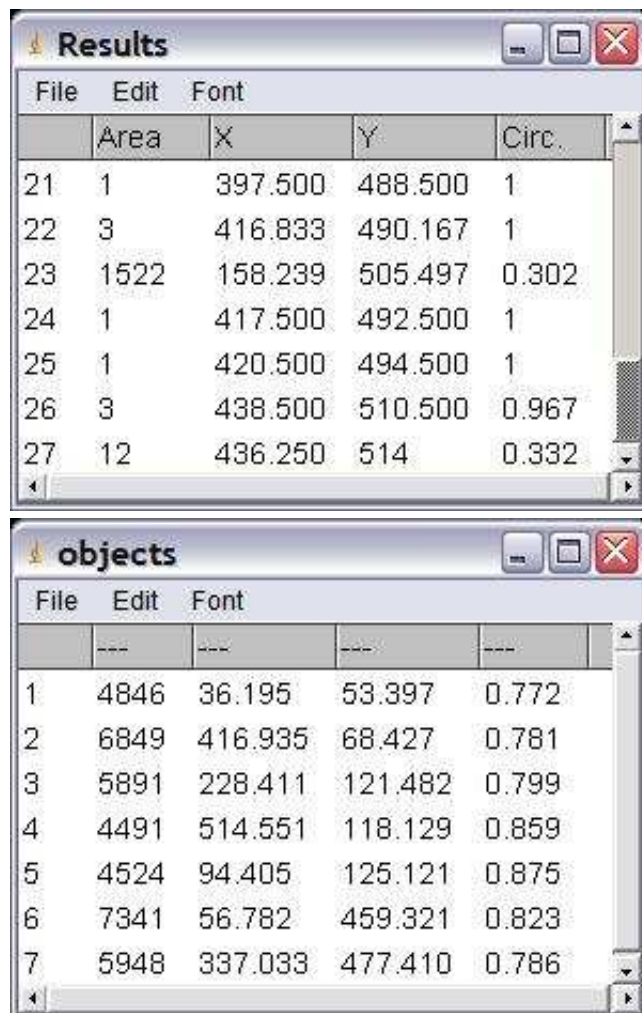
Operation: filter measurements



Author:

Volker Baecker

Example



Two screenshots of software windows. The top window, titled "Results", displays a table with 7 rows and 5 columns. The bottom window, titled "objects", displays a table with 7 rows and 5 columns. Both windows have a menu bar with "File", "Edit", and "Font" options.

	Area	X	Y	Circ.
21	1	397.500	488.500	1
22	3	416.833	490.167	1
23	1522	158.239	505.497	0.302
24	1	417.500	492.500	1
25	1	420.500	494.500	1
26	3	438.500	510.500	0.967
27	12	436.250	514	0.332

	---	---	---	---
1	4846	36.195	53.397	0.772
2	6849	416.935	68.427	0.781
3	5891	228.411	121.482	0.799
4	4491	514.551	118.129	0.859
5	4524	94.405	125.121	0.875
6	7341	56.782	459.321	0.823
7	5948	337.033	477.410	0.786

Description

Filters the measurements of a results table using minimal and maximal values for each measurement.

Options

filter measurements options		
? min area	4000	
? max area	8000	
? min mean		
? max mean		
? min std. dev.		
? max std. dev.		
? min mode		
? max mode		
? min min intensity		
? max min intensity		
? min max intensity		
? max max intensity		
? min centroid x		
? max centroid x		
? min centroid y		
? max centroid y		
? min center of mass x		
? max center of mass x		
? min center of mass y		
? max center of mass y		
? min perimeter		
? max perimeter		
? min bounding box x		
? max bounding box x		
? min bounding box y		
? max bounding box y		
? min bounding box wi...		
? max bounding box wi...		
? min bounding box hei...		
? max bounding box h...		
? min major		
? max major		
? min minor		
? max minor		
? min angle		
? max angle		
? min circularity	0.7	
? max circularity	1	
? min feret		
? max feret		
? min int. density		
? max int. density		
? min median		
? max median		
? min skew		
? max skew		
? min kurt		
? max kurt		

The actual value must be bigger than the minimal value and smaller than the maximal value. Only non-empty fields are taken into account.

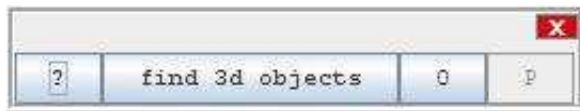
Parameter

The only parameter is the input results table.

Results

The result is the filtered results table;

Operation: find 3d objects

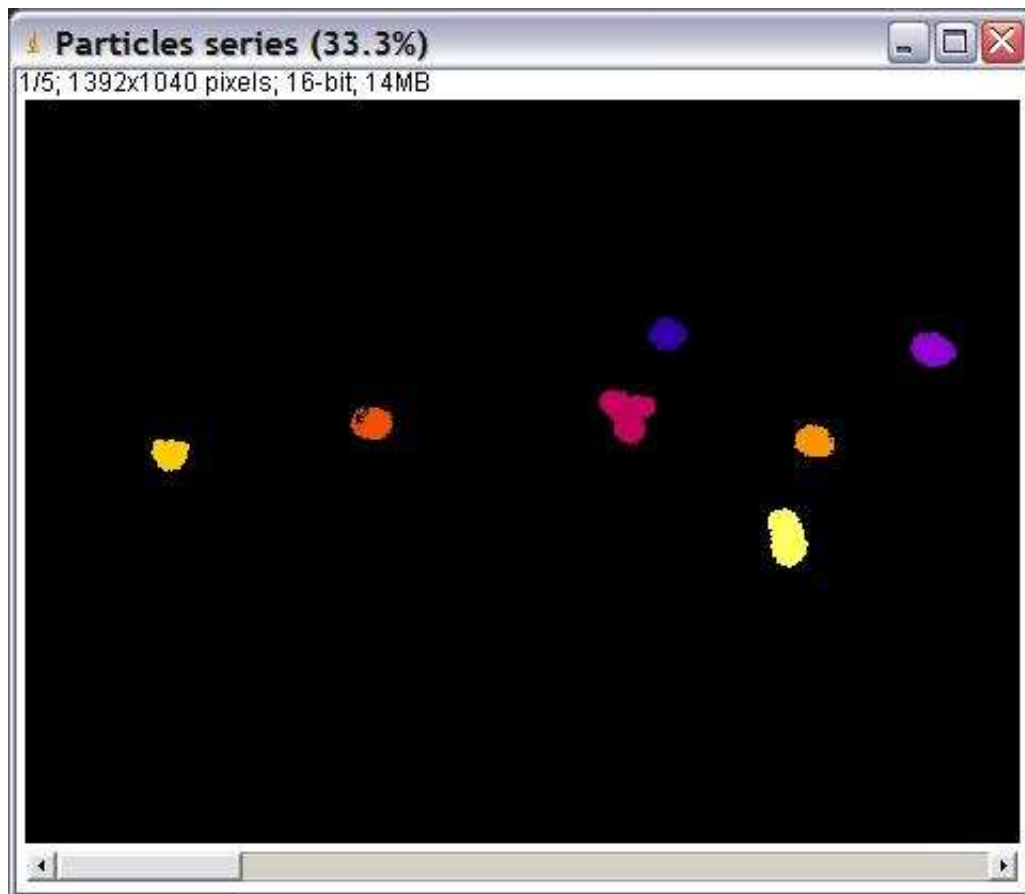


Author:

The operation calls the 3D objects counter plugin from Fabrice Cordelières and Jonathan Jackson.

Example



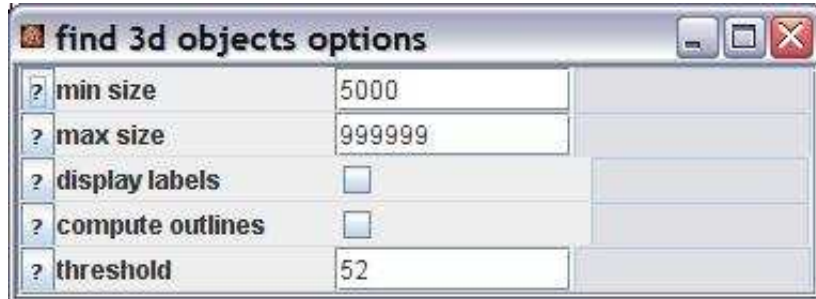


Results									
File	Edit	Font							
	Volume	Surface	Intensity	Centre X	Centre Y	Centre Z	Centre int X	Centre int Y	Centre int Z
1	8069	4726	64.846	1229.714	259.258	2.786	1230.009	259.375	2.715
2	5205	4391	57.764	897.734	324.376	2.150	897.483	324.270	2.126
3	8748	5471	64.752	1272.338	346.523	2.688	1272.541	346.351	2.608
4	11955	9023	60.074	844.449	439.465	2.473	844.539	440.135	2.432
5	5289	4000	65.163	489.549	452.356	2.372	490.710	452.368	2.335
6	5719	4293	59.538	1102.485	477.393	2.264	1102.114	477.394	2.236
7	6827	4413	62.650	200.880	493.721	2.655	201.517	493.938	2.617
8	13494	7963	64.888	1067.290	611.581	2.801	1067.516	612.398	2.720

Description

Counts the number of 3D objects in a stack and displays the volume, the surface, the centre of mass and the centre of intensity for each object.

Options



min size: The minimal size of objects that will be counted.

max size: The maximal size of objects that will be counted.

display labels: When checked the object numbers will be displayed in the result images.

compute outlines: When checked an image of the outlines (surfaces) of the objects is computed.

threshold: Only intensity values above the threshold value will be taken into account.

Parameter

The only parameter is the input image. The image must be an 8 bit or 16 bit greyscale image.

Results

Measurements (ResultsTable) : The results table with the measurements of the volume, the surface, the centre of mass and the centre of intensity for each object.

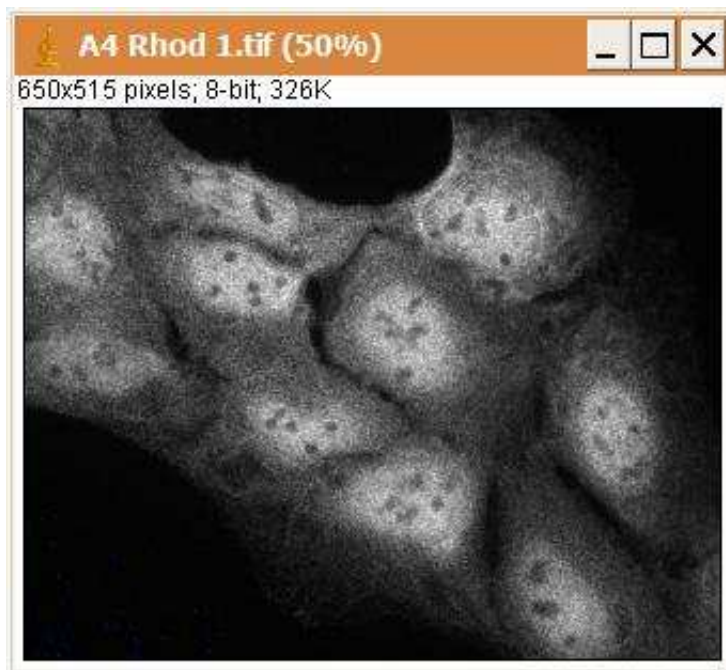
Operation: find and subtract background

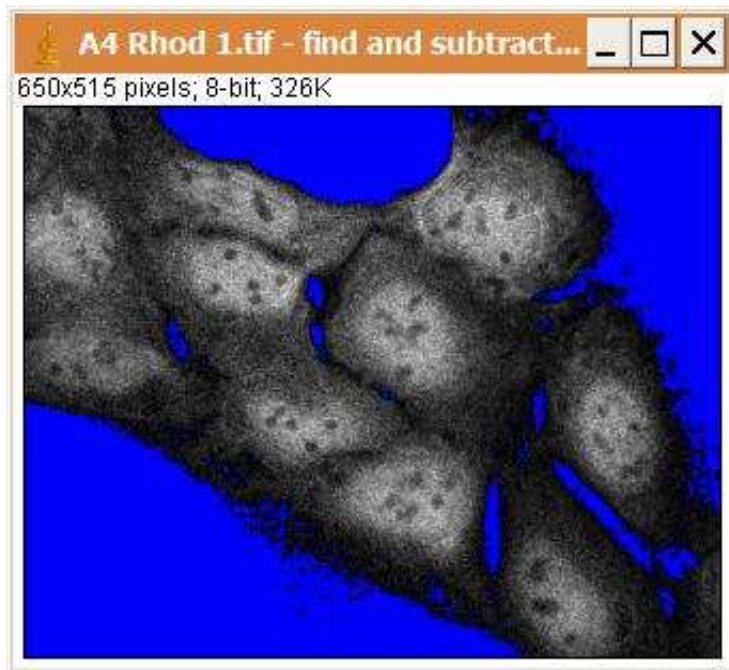


Author:

Volker Bäcker

Example

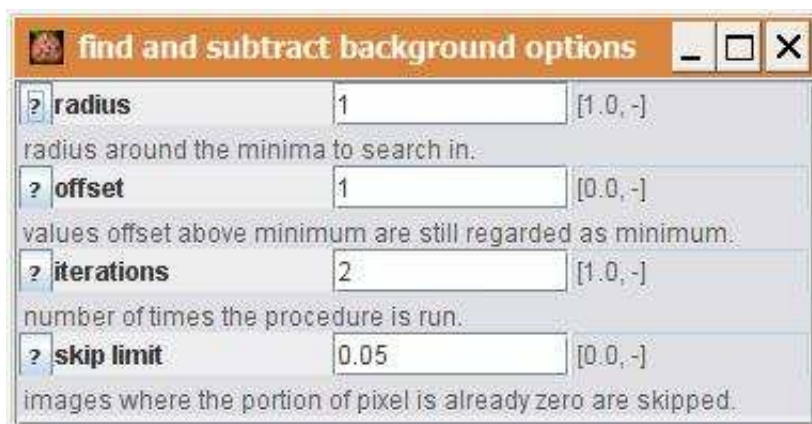




Description

The operation searches the biggest value around the minima of the image and subtracts it from the image.

Options



radius: The radius that defines the area around the minima in which the maximum background value is searched.

offset: Values upto offset above the minimum are still regarded as minimum.

iterations: The number of times the operation is run..

skip limit: Images for which the given portion of pixels is already zero are skipped.

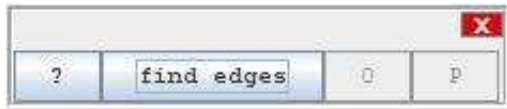
Parameter

The only parameter is the input image.

Results

The only result is the result image.

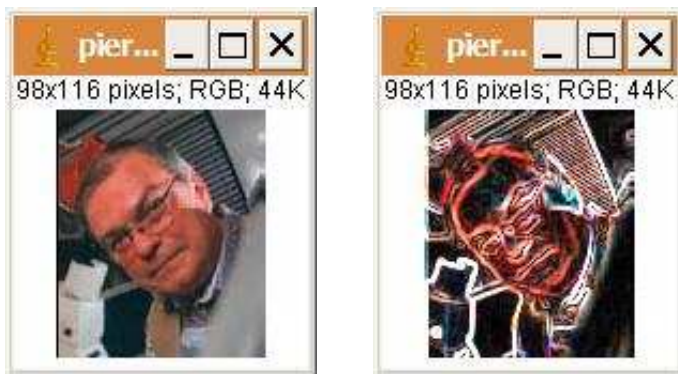
Operation: find edges



Author:

The operation calls the Find Edges command from ImageJ.

Example



Description

The operation uses a Sobel edge detector to highlight sharp changes in intensity in the input image.

Options

The operation has no options.

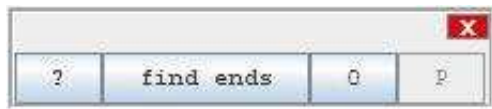
Parameter

The only parameter is the input image.

Results

The only result is the result image.

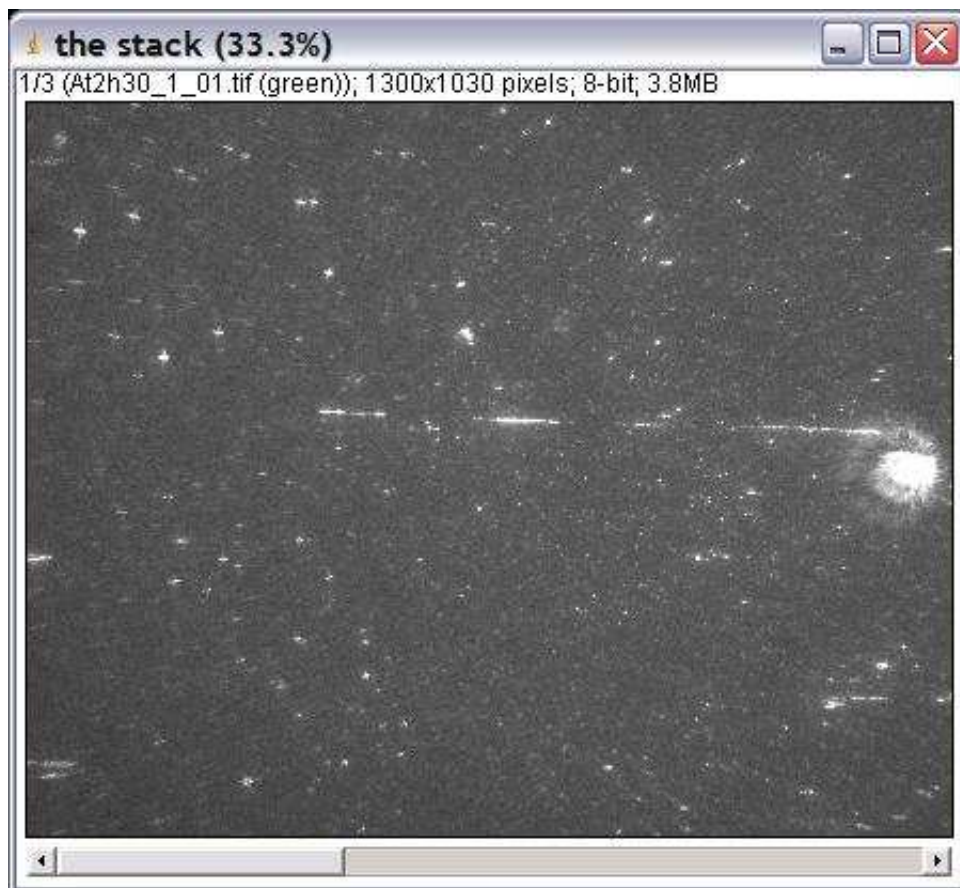
Operation: find ends

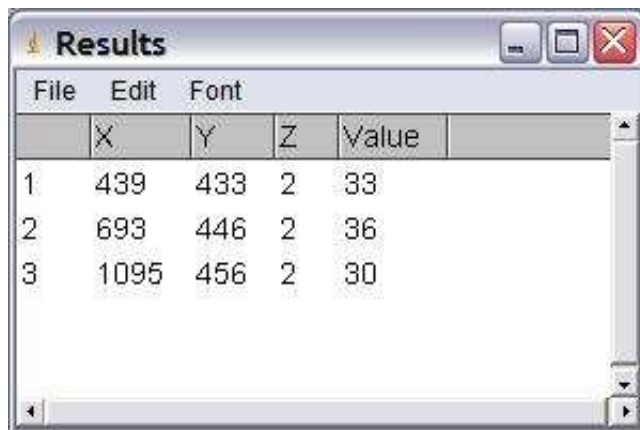


Author:

Volker Bäcker

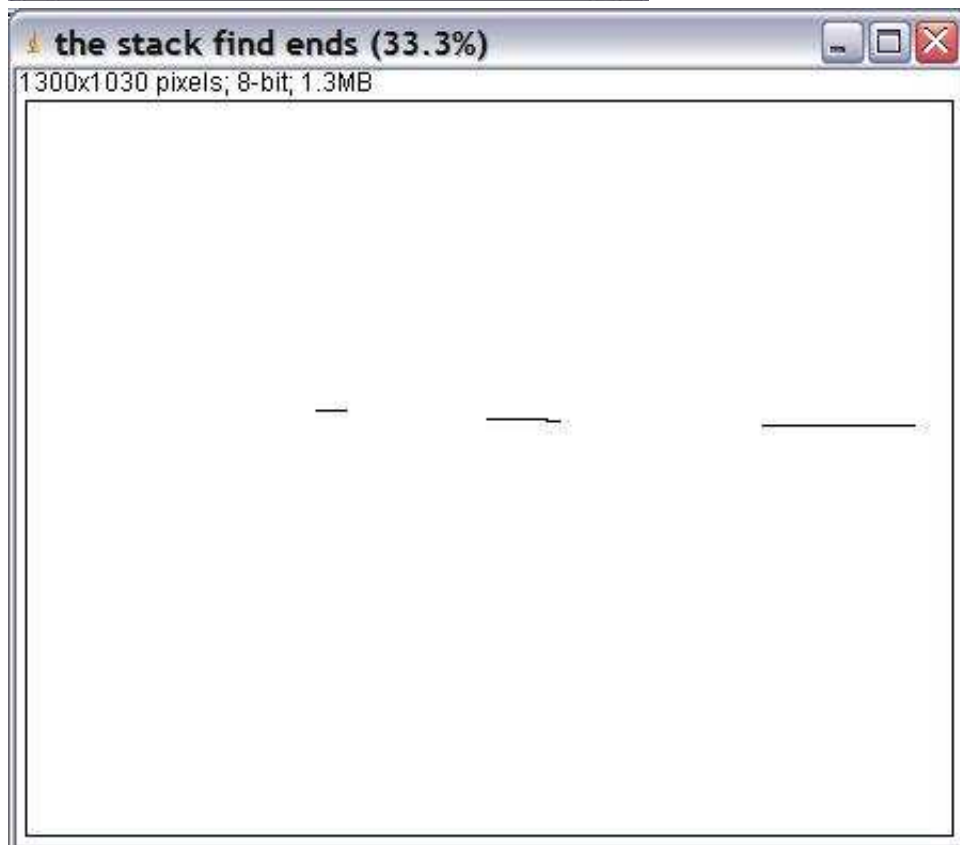
Example:





Results

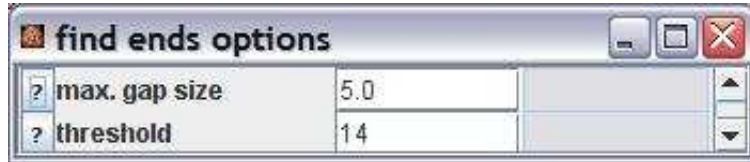
	X	Y	Z	Value
1	439	433	2	33
2	693	446	2	36
3	1095	456	2	30



Description

The operation finds the end points of more or less horizontal lines. The lines can have small gaps.

Options



max. gap size: The maximal allowed gap in which the intensity of pixels can be below or equal to the threshold

threshold: The intensity values of pixels on a line should be above the threshold.

Parameter

inputImage (ImagePlus): A stack with as many slices as there are points in the center points results table.

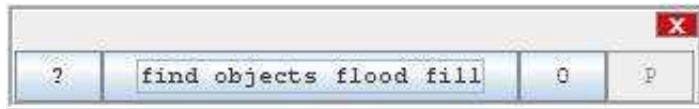
centerPoints (ResultsTable): The table must contain the X and Y columns with the coordinates of points on the lines to be traced.

Results

result (ImagePlus): An image showing the lines found.

resultLines (Vector): A list of the traced line segments.

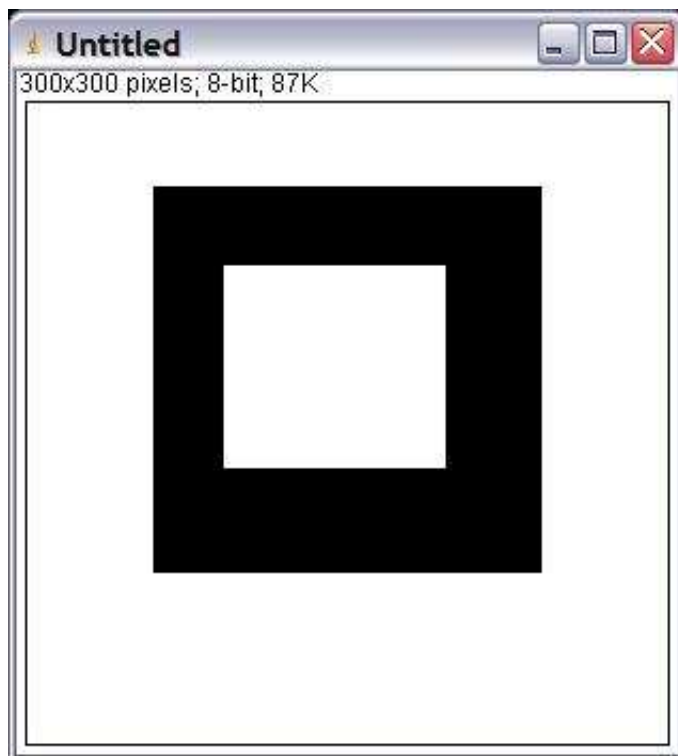
Operation: find objects flood fill



Author:

The operation calls the ParticleAnalyzer from ImageJ.

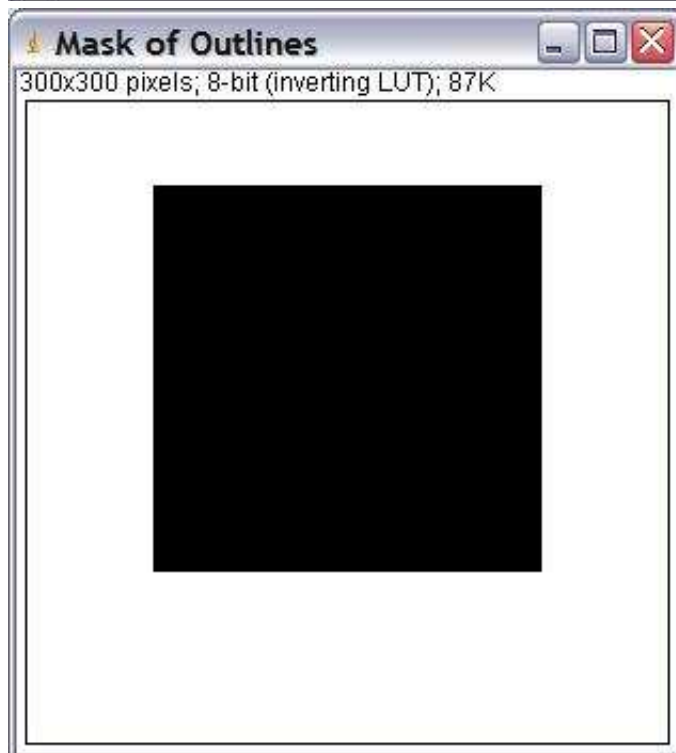
Example

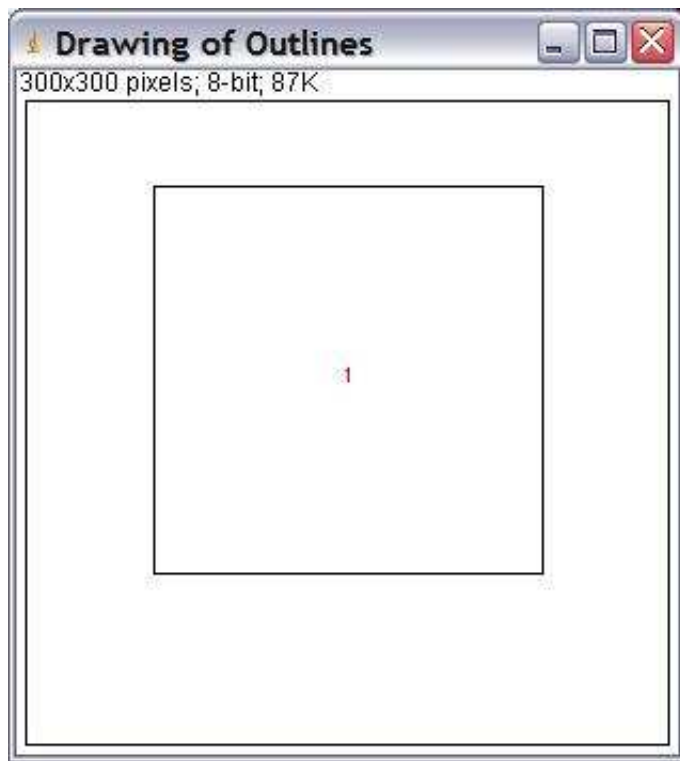


result from find objects

File Edit Font

	X	Y	Perim.	BX	BY	Width	Height	Circ.
1	150	129.500	723.657	59	39	182	181	0.790

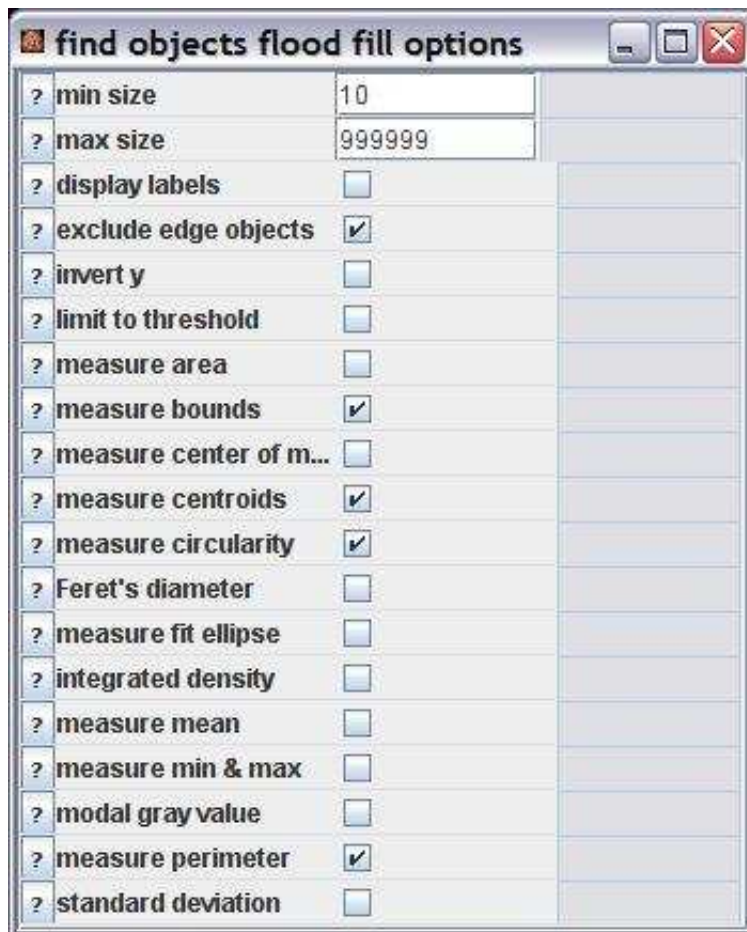




Description

This operation counts and measures objects in a binary image. This is a version of the find objects operation in which holes in the object are not taken into account. The operation will yield the same results if the object contains holes or not.

Options



min size: Objects smaller than min size are ignored.

max size: Objects bigger than max size are ignored.

display labels: If checked the title of the image is displayed in the results table.

exclude edge objects: If checked objects touching the edges of the image are ignored.

invert y: Inverts the y-coordinate so that the origin is in the lower left corner and not in the upper left corner anymore.

limit to threshold: If checked, only thresholded pixels are included in measurement calculations. Use *Image>Adjust>Threshold* to set the threshold limits.

measure area: Measure the areas of objects in square pixels. Areas are in calibrated units, such as square millimeters, if *Analyze>Set Scale* was used to spatially calibrate the image.

measure bounds: Measure the smallest rectangle enclosing each object. Uses the headings *BX*, *BY*, *Width* and *Height*, where *BX* and *BY* are the coordinates of the upper left corner of the rectangle.

measure center of mass: This is the brightness-weighted average of the x and y coordinates of all pixels in an object. Uses the *XM* and *YM* headings. These coordinates are the first order spatial moments.

measure centroids: The center point of an object. This is the average of the x and y coordinates of all of the pixels in the image or selection. Uses the *X* and *Y* Results table headings.

measure circularity: $4\pi(\text{area}/\text{perimeter}^2)$. A value of 1.0 indicates a perfect circle. As the value approaches 0.0, it indicates an increasingly elongated polygon. Values may not be valid for very small particles.

Feret's diameter: The longest distance between any two points along the selection boundary. Also known

as the caliper length.

measure fit ellipse: Fit an ellipse to the object. Uses the headings *Major*, *Minor* and *Angle*. *Major* and *Minor* are the primary and secondary axis of the best fitting ellipse. *Angle* is the angle between the primary axis and a line parallel to the x-axis of the image. Note that ImageJ cannot calculate the major and minor axis lengths if *Pixel Aspect Ratio* in the *Set Scale* dialog is not 1.0.

integrated density: The sum of the values of the pixels in the object.

measure mean: The average gray value within the object.

measure min & max: Minimum and maximum gray values within the object.

modal gray value: Most frequently occurring gray value within the object. Corresponds to the highest peak in the histogram.

measure perimeter: The length of the outside boundary of the object.

standard deviation: Standard deviation of the gray values used to generate the mean gray value.

Parameter

The only parameter is the input image. This must either be a two-dimensional, binary image (0 and 255) or a 2-dimensional image with a threshold set in the threshold adjuster.

Results

Measurements (ResultsTable): The results table containing the measurements of the found objects. Each line represents one object.

Mask (ImagePlus): A mask showing the objects that have been taken into account.

Outlines (ImagePlus): A drawing of the outer outlines of the objects that have been taken into account.

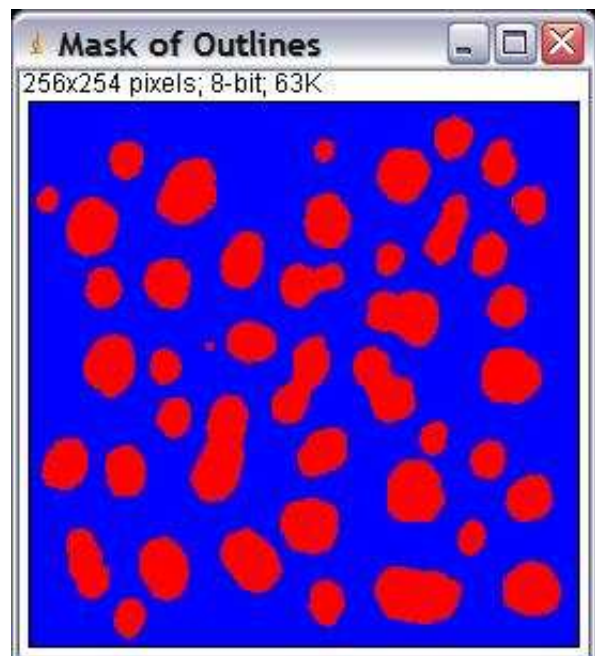
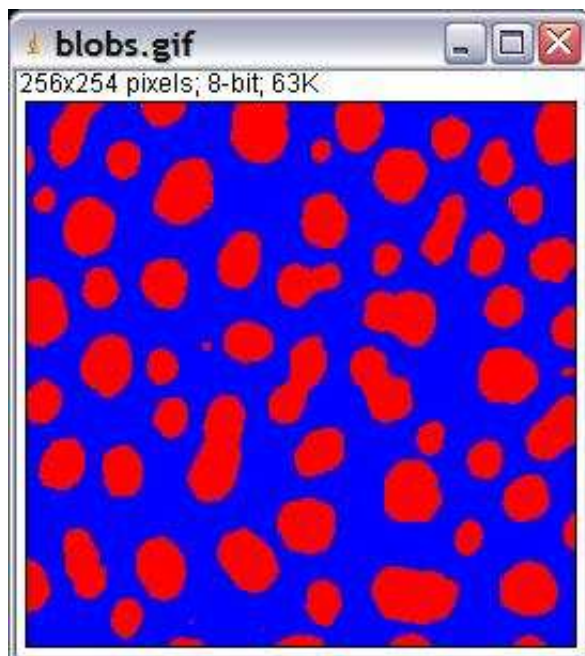
Operation: find objects

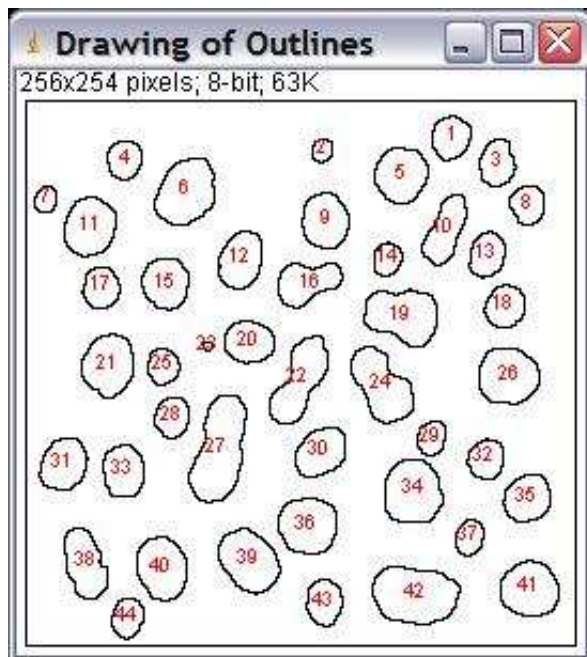


Author:

The operation calls the ParticleAnalyzer from ImageJ.

Example:





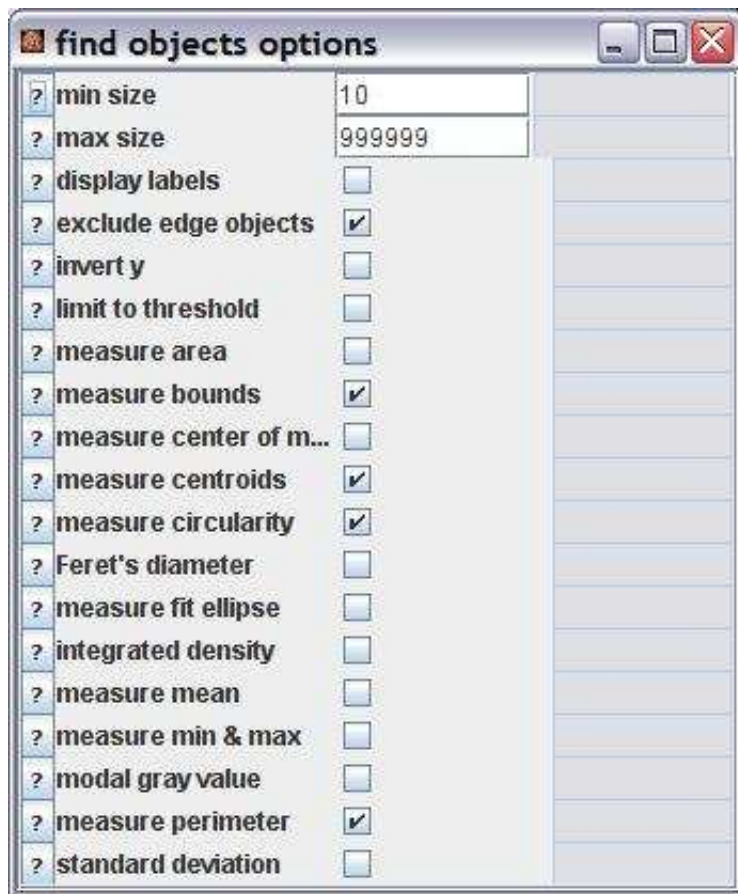
result from find objects

	X	Y	Perim.	BX	BY	Width	Height	Circ.
1	197.711	16.258	62.770	189	6	18	21	0.909
2	137.525	22.278	32.971	133	17	9	11	0.936
3	219.288	28.295	64.527	211	17	17	22	0.839
4	45.080	26.591	56.284	37	18	16	18	0.916
5	174.438	33.692	83.255	162	21	25	26	0.908
6	73.805	41.779	98.326	59	26	28	31	0.858
7	8.227	45.278	35.799	3	39	10	12	0.971

Description

Count and measure features of objects.

Options



min size: Objects smaller than min size are ignored.

max size: Objects bigger than max size are ignored.

display labels: If checked the title of the image is displayed in the results table.

exclude edge objects: If checked objects touching the edges of the image are ignored.

invert y: Inverts the y-coordinate so that the origin is in the lower left corner and not in the upper left corner anymore.

limit to threshold: If checked, only thresholded pixels are included in measurement calculations. Use *Image>Adjust>Threshold* to set the threshold limits.

measure area: Measure the areas of objects in square pixels. Areas are in calibrated units, such as square millimeters, if *Analyze>Set Scale* was used to spatially calibrate the image.

measure bounds: Measure the smallest rectangle enclosing each object. Uses the headings *BX*, *BY*, *Width* and *Height*, where *BX* and *BY* are the coordinates of the upper left corner of the rectangle.

measure center of mass: This is the brightness-weighted average of the x and y coordinates of all pixels in an object. Uses the *XM* and *YM* headings. These coordinates are the first order spatial moments.

measure centroids: The center point of an object. This is the average of the x and y coordinates of all of the pixels in the image or selection. Uses the *X* and *Y* Results table headings.

measure circularity: $4\pi(\text{area}/\text{perimeter}^2)$. A value of 1.0 indicates a perfect circle. As the value approaches 0.0, it indicates an increasingly elongated polygon. Values may not be valid for very small particles.

Feret's diameter: The longest distance between any two points along the selection boundary. Also known as the caliper length.

measure fit ellipse: Fit an ellipse to the object. Uses the headings *Major*, *Minor* and *Angle*. *Major* and *Minor* are the primary and secondary axis of the best fitting ellipse. *Angle* is the angle between the primary axis and a line parallel to the x-axis of the image. Note that ImageJ cannot calculate the major and minor axis lengths if *Pixel Aspect Ratio* in the *Set Scale* dialog is not 1.0.

integrated density: The sum of the values of the pixels in the object.

measure mean: The average gray value within the object.

measure min & max: Minimum and maximum gray values within the object.

modal gray value: Most frequently occurring gray value within the object. Corresponds to the highest peak in the histogram.

measure perimeter: The length of the outside boundary of the object.

standard deviation: Standard deviation of the gray values used to generate the mean gray value.

Parameter

The only parameter is the input image. This must either be a two-dimensional, binary image (0 and 255) or a 2-dimensional image with a threshold set in the threshold adjuster.

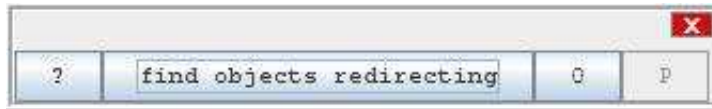
Results

Measurements (ResultsTable): The results table containing the measurements of the found objects. Each line represents one object.

Mask (ImagePlus): A mask showing the objects that have been taken into account.

Outlines (ImagePlus): A drawing of the outer outlines of the objects that have been taken into account.

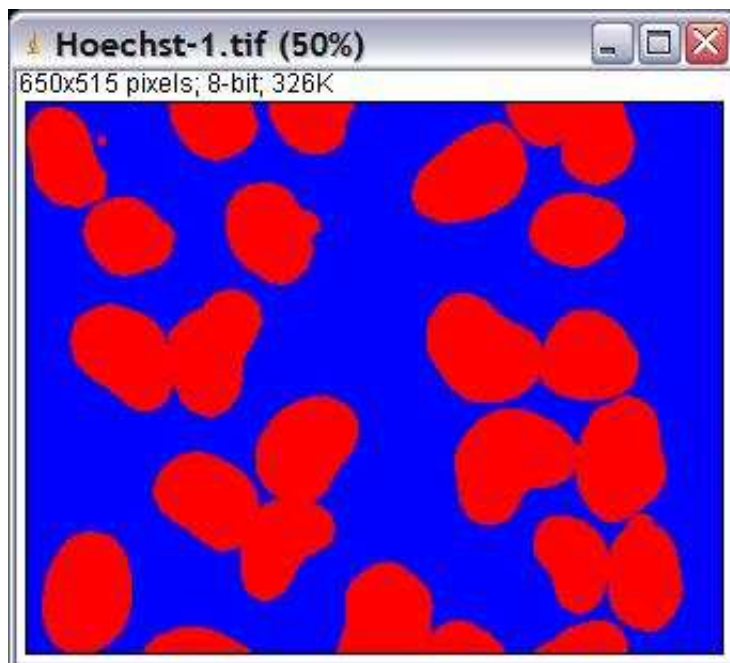
Operation: find objects redirecting

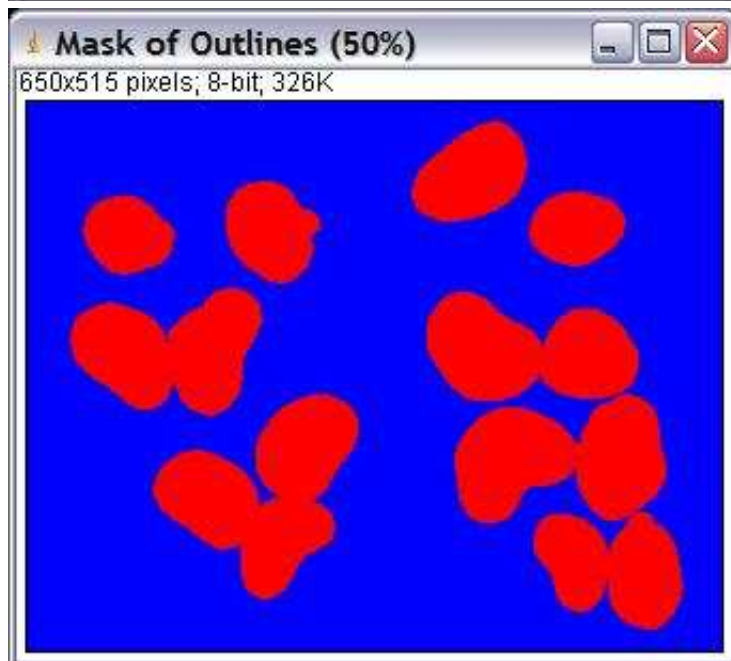
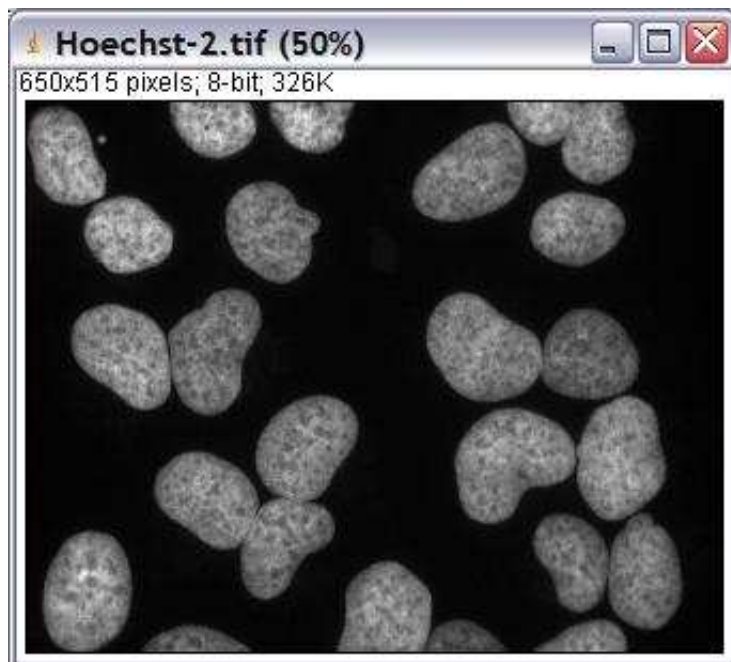


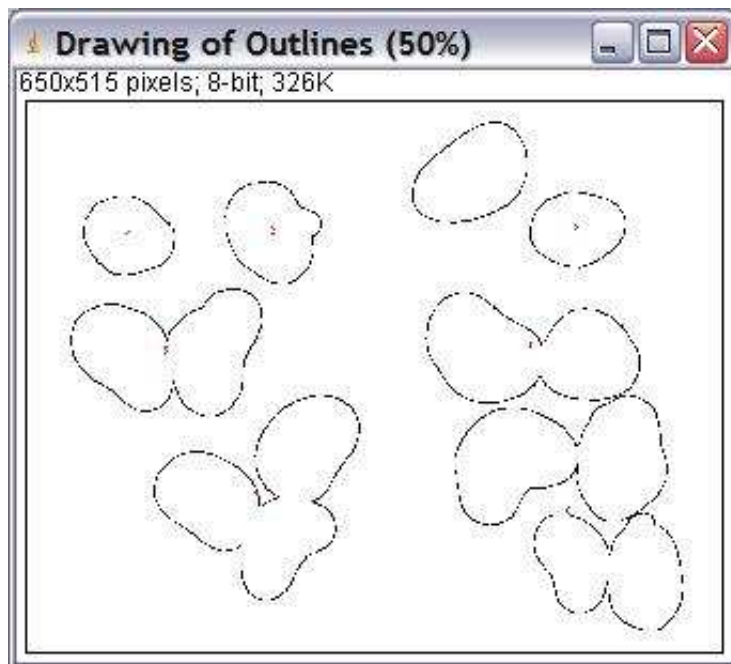
Author:

The operation calls the ParticleAnalyzer from ImageJ.

Example:





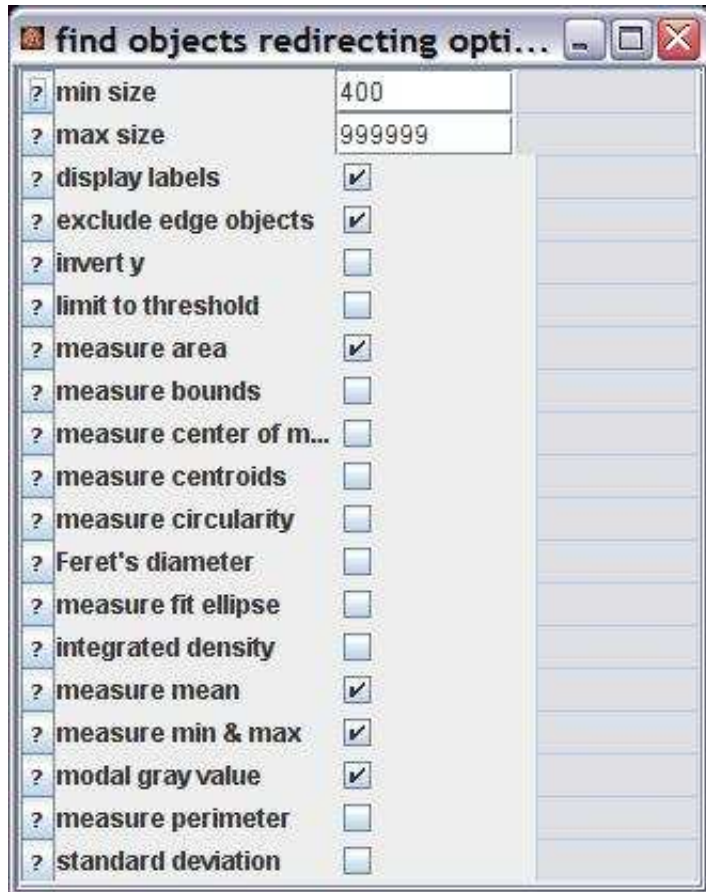


	Label	Area	Mean	Mode	Min	Max
1	Hoechst-2	7207	106.227	108	35	180
2	Hoechst-2	6193	105.265	108	35	176
3	Hoechst-2	4798	108.678	113	35	192
4	Hoechst-2	4740	140.673	157	35	217
5	Hoechst-2	14205	113.585	116	33	221
6	Hoechst-2	14312	101.029	107	35	192
7	Hoechst-2	19159	112.007	120	27	199
8	Hoechst-2	27059	110.405	121	35	194

Description

Count and measure features of objects. Features that measure pixel values are measured on the redirect image, not on the mask.

Options



min size: Objects smaller than min size are ignored.

max size: Objects bigger than max size are ignored.

display labels: If checked the title of the image is displayed in the results table.

exclude edge objects: If checked objects touching the edges of the image are ignored.

invert y: Inverts the y-coordinate so that the origin is in the lower left corner and not in the upper left corner anymore.

limit to threshold: If checked, only thresholded pixels are included in measurement calculations. Use *Image>Adjust>Threshold* to set the threshold limits.

measure area: Measure the areas of objects in square pixels. Areas are in calibrated units, such as square millimeters, if *Analyze>Set Scale* was used to spatially calibrate the image.

measure bounds: Measure the smallest rectangle enclosing each object. Uses the headings *BX*, *BY*, *Width* and *Height*, where *BX* and *BY* are the coordinates of the upper left corner of the rectangle.

measure center of mass: This is the brightness-weighted average of the x and y coordinates of all pixels in an object. Uses the *XM* and *YM* headings. These coordinates are the first order spatial moments.

measure centroids: The center point of an object. This is the average of the x and y coordinates of all of the pixels in the image or selection. Uses the *X* and *Y* Results table headings.

measure circularity: $4\pi(\text{area}/\text{perimeter}^2)$. A value of 1.0 indicates a perfect circle. As the value approaches 0.0, it indicates an increasingly elongated polygon. Values may not be valid for very small particles.

Feret's diameter: The longest distance between any two points along the selection boundary. Also known as the caliper length.

measure fit ellipse: Fit an ellipse to the object. Uses the headings *Major*, *Minor* and *Angle*. *Major* and *Minor* are the primary and secondary axis of the best fitting ellipse. *Angle* is the angle between the primary axis and a line parallel to the x-axis of the image. Note that ImageJ cannot calculate the major and minor axis lengths if *Pixel Aspect Ratio* in the *Set Scale* dialog is not 1.0.

integrated density: The sum of the values of the pixels in the object.

measure mean: The average gray value within the object.

measure min & max: Minimum and maximum gray values within the object.

modal gray value: Most frequently occurring gray value within the object. Corresponds to the highest peak in the histogram.

measure perimeter: The length of the outside boundary of the object.

standard deviation: Standard deviation of the gray values used to generate the mean gray value.

Parameter

The only parameter is the input image. This must either be a two-dimensional, binary image (0 and 255) or a 2-dimensional image with a threshold set in the threshold adjuster.

Results

Measurements (ResultsTable): The results table containing the measurements of the found objects. Each line represents one object.

Mask (ImagePlus): A mask showing the objects that have been taken into account.

Outlines (ImagePlus): A drawing of the outer outlines of the objects that have been taken into account.

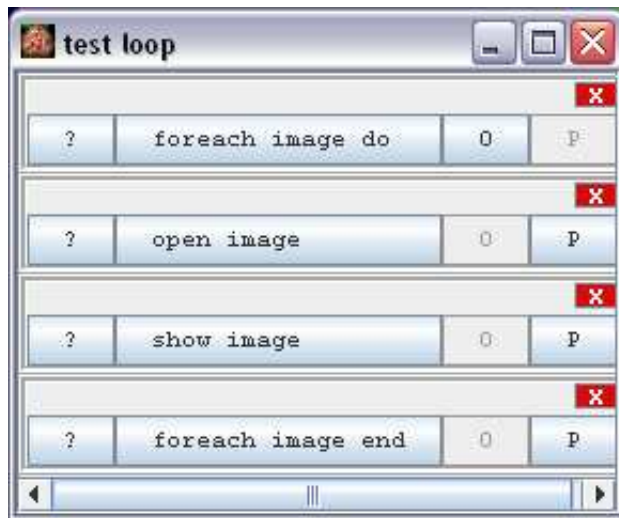
Operation: foreach image do



Author:

Volker Bäcker

Example



Open and show all images selected by the user.

Description

The user sets up a list of image filenames. "foreach image do" and "foreach image end" build a loop. In each cycle the result of "foreach image do" is the current filename. When "foreach image end" is reached the current filename is set to the next in the list and the execution continues after "foreach image do". When the last filename in the list is reached execution continues after "foreach image end".

Options





image list: Pressing "edit" opens the list editor that allows to add and remove images from the list. When a folder is added all images in the folder and in all subfolders are added.

use sequence opener: If checked the sequence opener is used instead of the file dialog to select the files. This is preferable when there are many files in a folder on a windows system.

Parameter

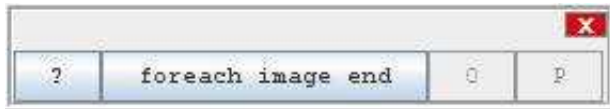
The operation has no parameters.

Results

The operation has three results:

1. current filename:
The current filename as a String.
2. at end:
A Boolean that tells whether the last image filename in the list has been reached.
3. do operation:
This is the operation itself. This result is needed by the "foreach image end" operation to determine where to continue execution.

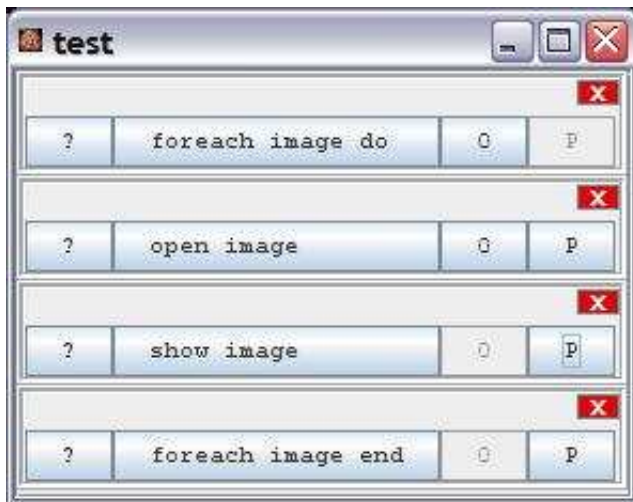
Operation: foreach image end



Author:

Volker Baecker

Example



Description

The operation marks the end of a loop. In each cycle the result of "foreach image do" is the current filename. When "foreach image end" is reached the current filename is set to the next in the list and the execution continues after "foreach image do". When the last filename in the list is reached execution continues after "foreach image end".

Options

The operation has no options.

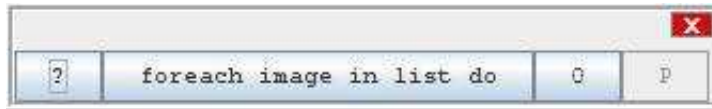
Parameter

doOperation (ForEachImageDoOperation) : The operation that marks the start of the loop.

Results

The operation has no result.

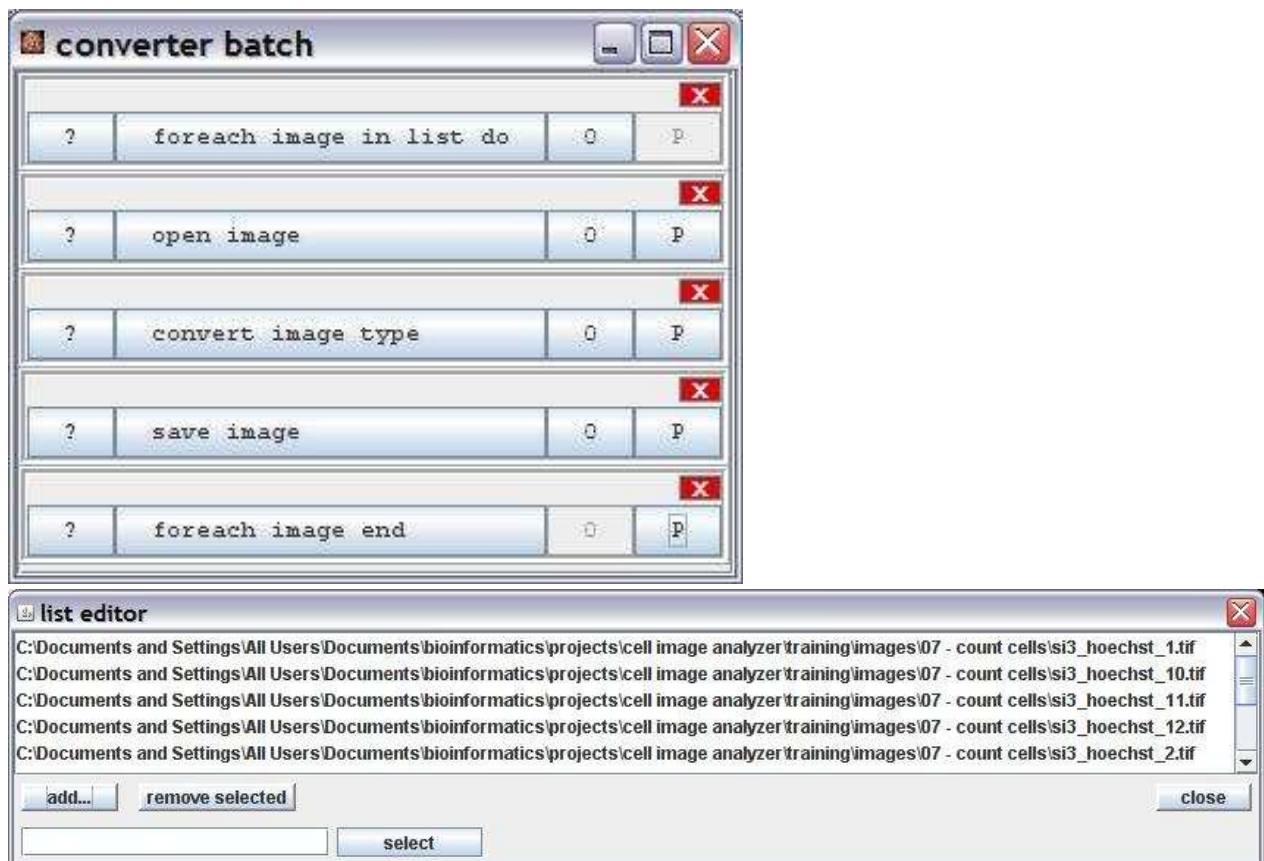
Operation: foreach image in list do



Author:

Volker Bäcker

Example



Description

The user sets up a list of image filenames. "foreach image in list do" and "foreach image end" build a loop. In each cycle the result of "foreach image do" is the current filename. When "foreach image end" is reached the current filename is set to the next in the list and the execution continues after "foreach image do". When the last filename in the list is reached execution continues after "foreach image end". The user sets up the file list using the list editor.

Options



image list: Pressing "edit" opens the list editor that allows to add and remove images from the list. When a folder is added all images in the folder and in all subfolders are added.

use sequence opener: If checked the sequence opener is used instead of the file dialog to select the files. This is preferable when there are many files in a folder on a windows system.

Parameter

The operation has no parameters.

Results

The operation has three results:

1. current filename:
The current filename as a String.
2. at end:
A Boolean that tells whether the last image filename in the list has been reached.
3. do operation:
This is the operation itself. This result is needed by the "foreach image end" operation to determine where to continue execution.

Operation: foreach object do



Author:

Volker Bäcker

Example

measure objects			
?	get current image	0	P
?	get image name	0	P
?	std dev around median threshold	0	P
?	invert image	0	P
?	find objects	0	P
?	foreach object do	0	P
?	select object	0	P
?	transfer selection	0	P
?	accept or skip or exit	0	P
?	measure	0	P
?	report measurements	0	P
?	foreach object end	0	P



image	Area	Mean	StdDev	Mode	Min	Max	X	Y	XM	YM	Perim.
A4 dapi 1.tif	6549	121,26	30,98	116	34	222	197,91	86,39	199,19	86,65	351,61
A4 dapi 1.tif	7445	144,28	37,65	157	34	239	428,95	113,63	428,18	113,49	334,63
A4 dapi 1.tif	5915	121,53	30,02	116	34	191	43,39	126,07	42,9	126,08	289,66
A4 dapi 1.tif	5856	121,14	31,73	135	34	235	199,74	160,98	200,44	161,1	297,32
A4 dapi 1.tif	7901	141,44	37,8	150	34	246	353,05	213,41	353,42	212,7	332,29
A4 dapi 1.tif	6007	130,5	30,38	145	34	198	64,78	237,61	65,29	237,17	306,49
A4 dapi 1.tif	6588	127,19	31,05	132	34	211	552,89	305,43	552,07	304,81	310,15
A4 dapi 1.tif	4962	130,88	37,53	130	34	224	268,03	300,31	267,51	300,03	302,49
A4 dapi 1.tif	7673	161,63	43,12	167	34	255	360,99	365,87	360,81	365,96	339,22
A4 dapi 1.tif	7385	118,1	31,05	126	34	210	497,41	459,25	499,43	459,55	339,89

Description

The operation marks the start of a loop through a results table. In each iteration the centroid of the current row is accessible as currentPoint. If the operation is started without being in an application the current results table is shown in a different table interface. The objects corresponding to the selected rows are selected in the mask image using the do wand command.

Options

The operation has no options.

Parameter

The only parameter is the results table containing at least the centroids of the objects.

Results

currentPoint (Point2D): The current point in each iteration.

atEnd (Boolean): True when the loop has reached the end.

doOperation (ForeachObjectDoOperation): The operation itself, needed as input for the ForeachObjectEndOperation.

index (Integer): The index of the current iteration (starting with 0).

Operation: foreach object end



Author:

Volker Bäcker

Example

measure objects			
?	get current image	0	P
?	get image name	0	P
?	std dev around median threshold	0	P
?	invert image	0	P
?	find objects	0	P
?	foreach object do	0	P
?	select object	0	P
?	transfer selection	0	P
?	accept or skip or exit	0	P
?	measure	0	P
?	report measurements	0	P
?	foreach object end	0	P



image	Area	Mean	StdDev	Mode	Min	Max	X	Y	XM	YM	Perim.
A4 dapi 1.tif	6549	121,26	30,98	116	34	222	197,91	86,39	199,19	86,65	351,61
A4 dapi 1.tif	7445	144,28	37,65	157	34	239	428,95	113,63	428,18	113,49	334,63
A4 dapi 1.tif	5915	121,53	30,02	116	34	191	43,39	126,07	42,9	126,08	289,66
A4 dapi 1.tif	5856	121,14	31,73	135	34	235	199,74	160,98	200,44	161,1	297,32
A4 dapi 1.tif	7901	141,44	37,8	150	34	246	353,05	213,41	353,42	212,7	332,29
A4 dapi 1.tif	6007	130,5	30,38	145	34	198	64,78	237,61	65,29	237,17	306,49
A4 dapi 1.tif	6588	127,19	31,05	132	34	211	552,89	305,43	552,07	304,81	310,15
A4 dapi 1.tif	4962	130,88	37,53	130	34	224	268,03	300,31	267,51	300,03	302,49
A4 dapi 1.tif	7673	161,63	43,12	167	34	255	360,99	365,87	360,81	365,96	339,22
A4 dapi 1.tif	7385	118,1	31,05	126	34	210	497,41	459,25	499,43	459,55	339,89

Description

The operation marks the end of a foreach object do loop.

Options

The operation has no options

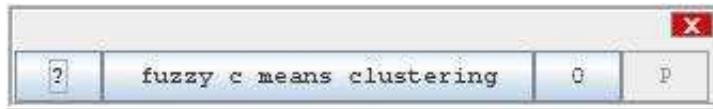
Parameter

doOperation (ForeachObjectDoOperation): The operation marking the start of the loop.

Results

The operation has no result.

Operation: fuzzy c means clustering

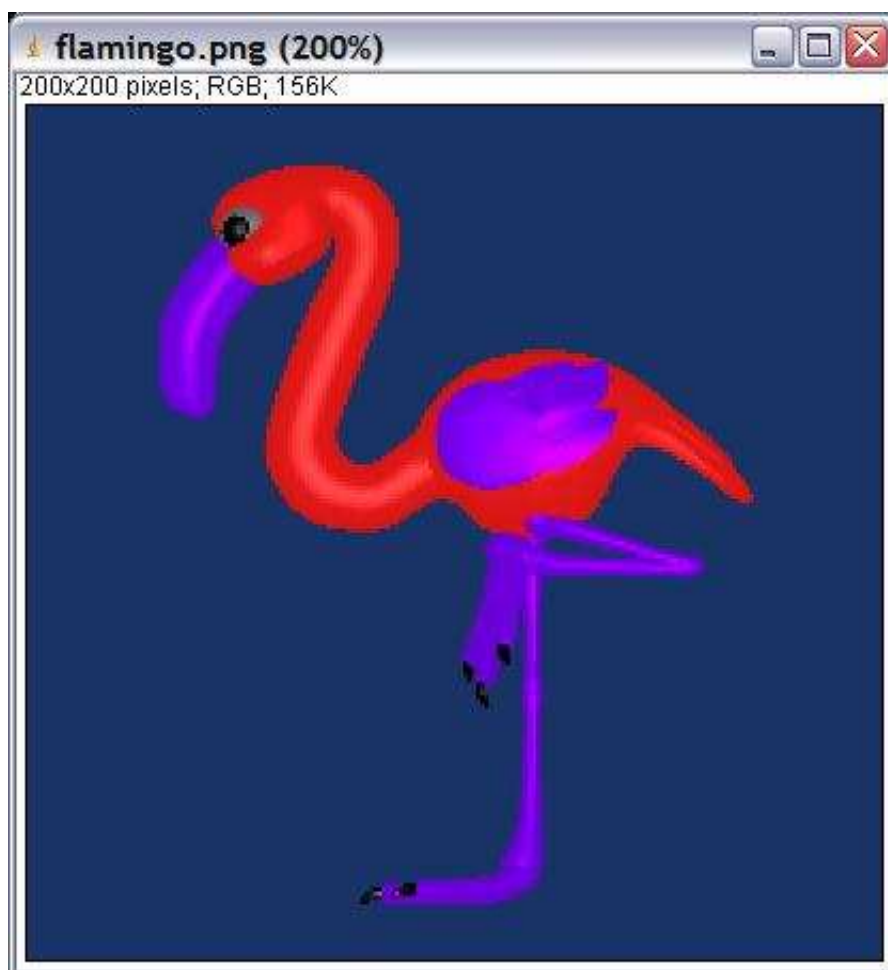


Author:

The algorithm as described in the paper below has been implemented by Volker Baecker.

Professional Paper, Comparison of Fuzzy C-means Algorithm and New Fuzzy Clustering and Fuzzy Merging Algorithm,
Liyan Zhang, Computer Science Department University of Nevada, Reno Reno, NV 89557,

Example

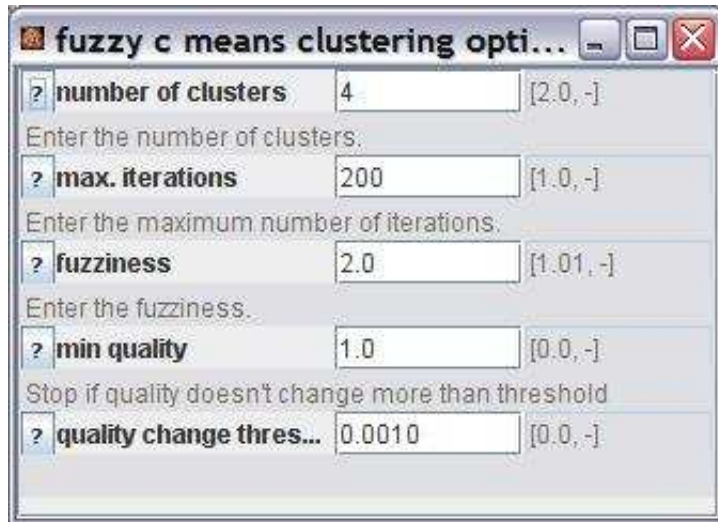




Description

The operation segments an image into n classes using the fuzzy c means clustering algorithm.

Options



number of clusters: The number of clusters is the number of different segments in the result image.

max. iterations.: The maximum number of iterations that the optimisation runs.

fuzziness: The higher this value, the faster the algorithm converges.

min quality: The minimal quality that must be reached before the algorithm may be stopped because of the quality change threshold.

quality change threshold: The algorithm stops when the change of the quality is less than the threshold value and the min quality has already been reached.

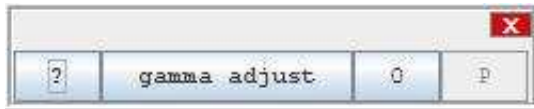
Parameter

The only parameter is the input image.

Results

The only result is the result image. Each cluster is represented by a different number (0, 1, 2, ...) starting from 0.

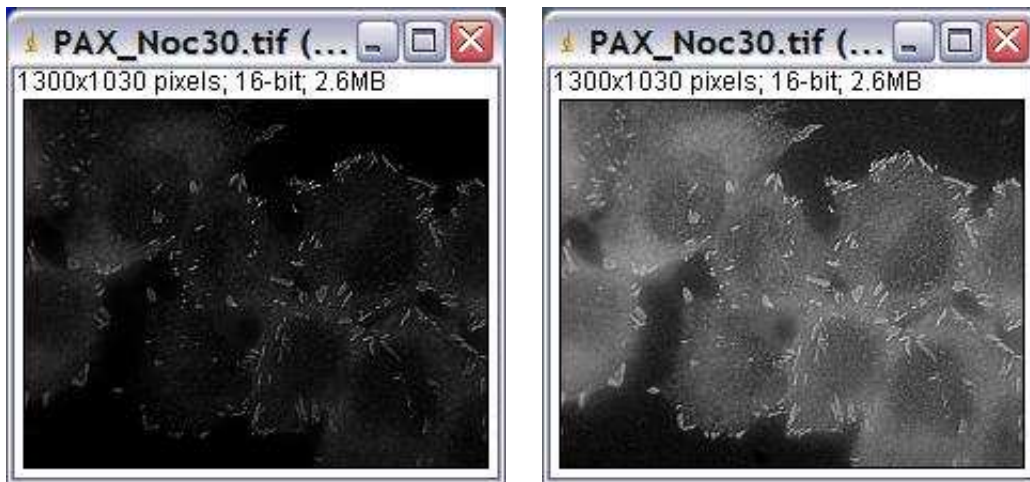
Operation: gamma adjust



Author:

Uses the method gamma of the class ImageProcessor from ImageJ.

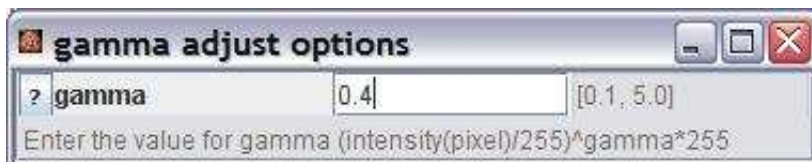
Example



Description

The operation applies a gamma correction to the intensity values of the input image according to the formula $i' = (\exp(\log(i/255)*\text{gamma}))*255$.

Options



gamma: The new pixel intensity i' will be calculated from the old pixel intensity i according to $i' = (\exp(\log(i/255)*\text{gamma}))*255$

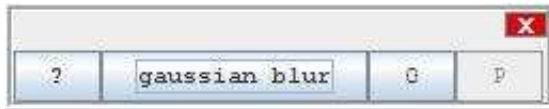
Parameter

The only parameter is the input image.

Results

The only result is the image with the gamma adjusted intensity values.

Operation: gaussian blur



Author:

Uses the class GaussianBlur from ImageJ.

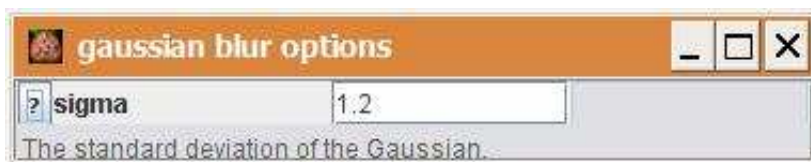
Example



Description

Applies a Gaussian blur filter to the input image.

Options



sigma: The radius of decay to $\exp(-0.5) \sim 61\%$, i.e. the standard deviation sigma of the Gaussian.

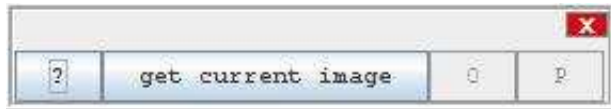
Parameter

The only parameter is the input image.

Results

The only result is the result image.

Operation: get current image



Author:

Volker Bäcker

Example



Description

Get a reference to the active image

Options

The operation has no options.

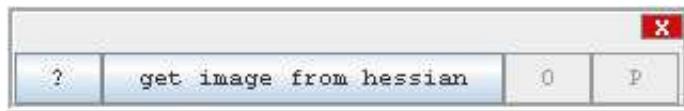
Parameter

The operation has no parameters.

Results

result (ImagePlus): The active image.

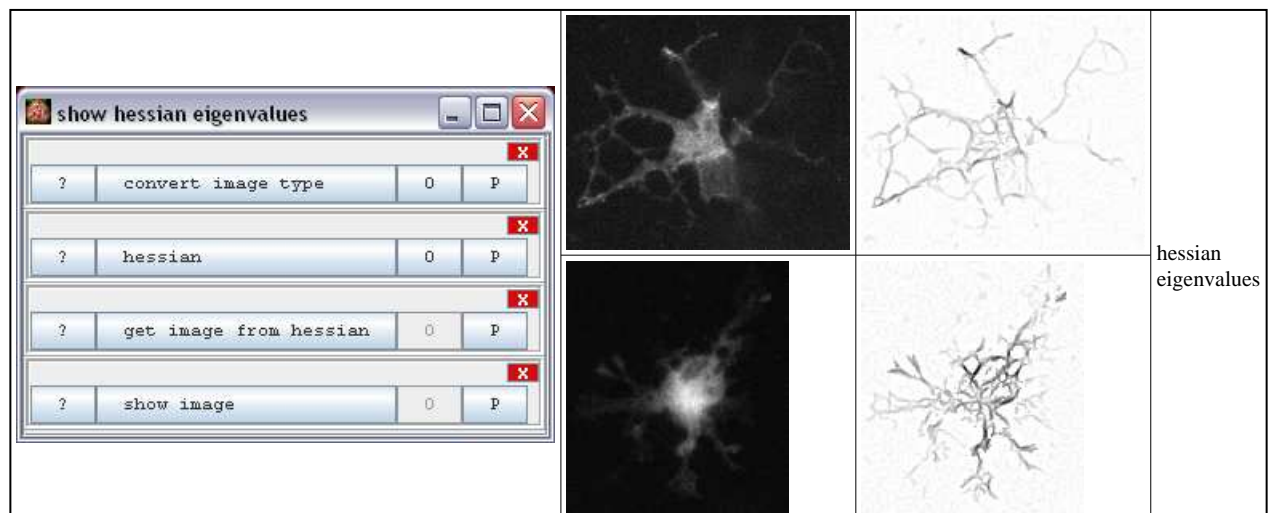
Operation: get image from hessian



Author:

Volker Bäcker.

Example



Description

The operation takes the eigenvalues from the operation hessian and computes an image from them.

Options

The operation has no options.

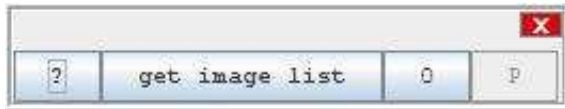
Parameter

The only parameter is the hessian image.

Results

The only result is the result image.

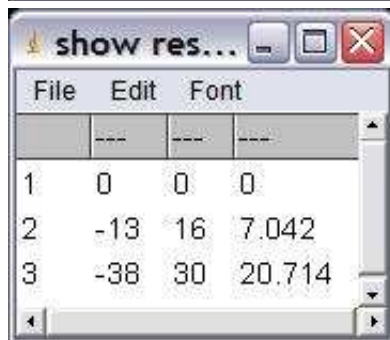
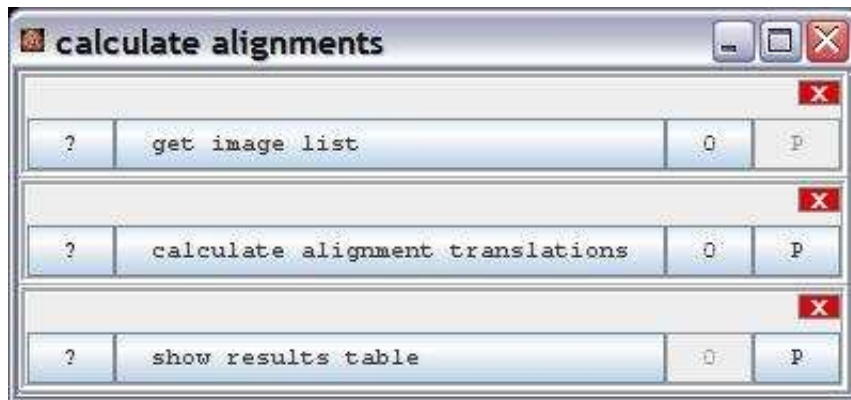
Operation: get image list



Author:

Volker Bäcker

Example



Description

Ask the user to provide a list of images. The list will be available to other operations as a result of this operation. The operation can be used to provide the input for the auto align slices and calculate alignment translations operations.

Options



image list: Pressing "edit" opens the list editor that allows to add and remove images from the list. When a folder is added all images in the folder and in all subfolders are added.

use sequence opener: If checked the sequence opener is used instead of the file dialog to select the files. This is preferable when there are many files in a folder on a windows system.

Parameter

The operation has no parameters.

Results

imageList (Vector): The list of images.

filename (String): The filename of the first image in the list.

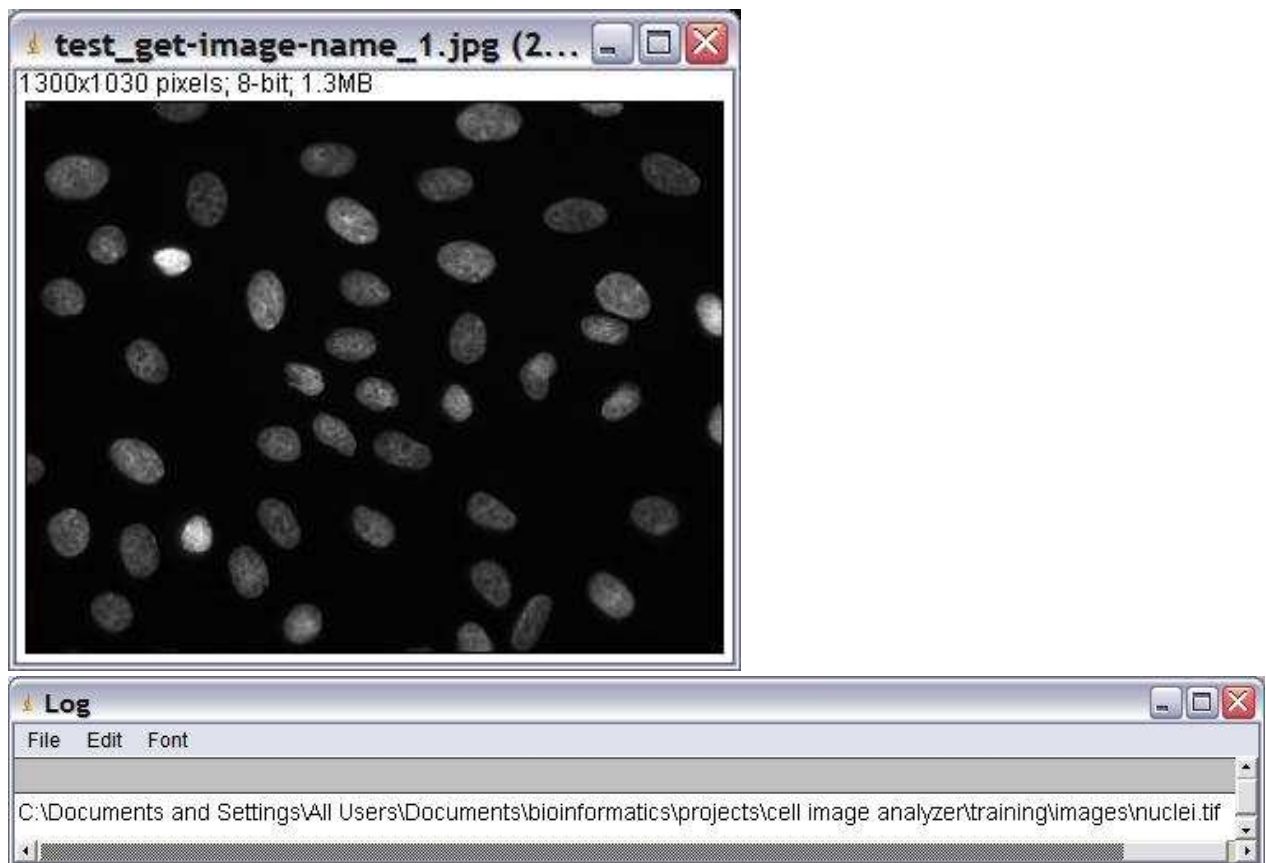
Operation: get image name



Author:

Volker Bäcker

Example



Description

Get the absolute filename of an image.

Options

The operation has no options.

Parameter

The only parameter is the input image.

Results

imageName (String): The absolute filename of the image.

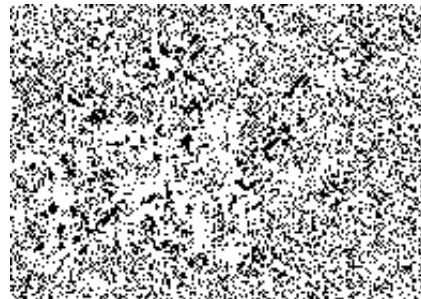
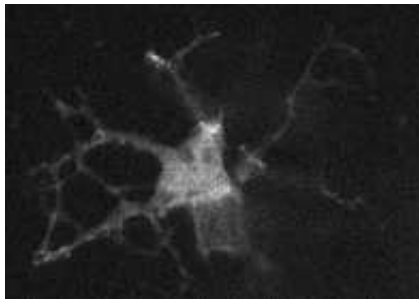
Operation: get vector image from hessian



Author:

Volker Bäcker.

Example



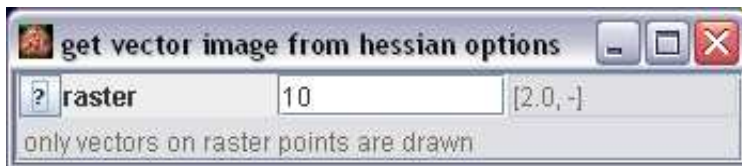
hessian displayed with raster 2

the eigenvectors of the

Description

Draws the eigenvectors from the hessian operations.

Options



raster: Only points on the raster are drawn with the maximum length of raster / 2.

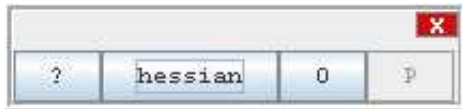
Parameter

The only parameter is the hessian image.

Results

The only result is the result image.

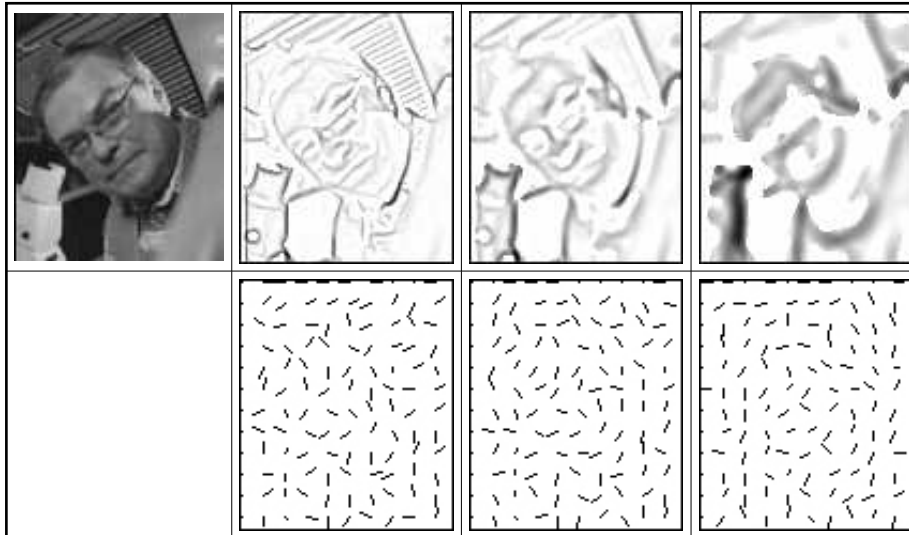
Operation: hessian



Author:

The operation uses the cost computation from NeuronJ by Erik Meijering. There exists a separated plugin to compute the Hessian in FeatureJ.

Example



eigenvalues and normalized eigenvectors of the hessian for smoothing scale = 1, 2, 4

Description

Creates a cost image and a vector field computed from the eigenvalues and eigenvectors of the Hessian of the input image. The Hessian derivative can be used to discriminate locally between plate-like, line-like, and blob-like image structures. The Hessian matrix of a scalar function of an n -dimensional vector is the symmetric $n \times n$ matrix of second partial derivatives.

Options



scale: the smoothing scale is equal to the standard deviation of the Gaussian derivative kernel used in computing the second-order derivatives

Parameter

The only parameter is the input image. The input image must be a 2D, 8bit, greyscale image.

Results

The result is a Hessian image that contains both the eigenvalue array and the eigenvector array. To convert them to a normal, displayable image use the operations "get image from hessian" and "get vector image from hessian".

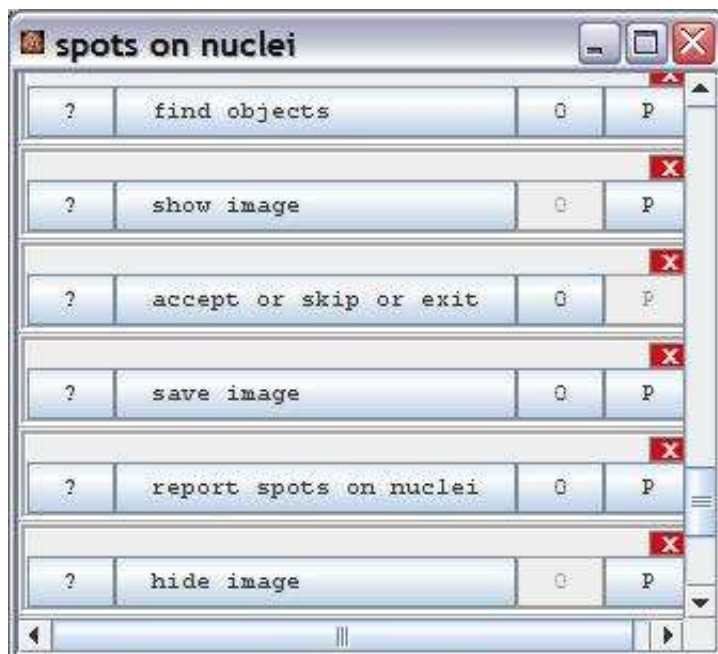
Operation: hide image



Author:

Volker Bäcker

Example



Description

Hides the input image. Closes the window of the image if there is one and removes the image from ImageJ's image list.

Options

The operation has no options.

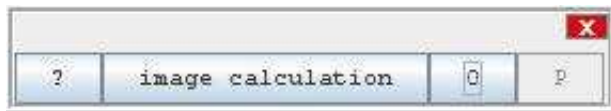
Parameter

The only parameter is the input image.

Results

The operation has no results.

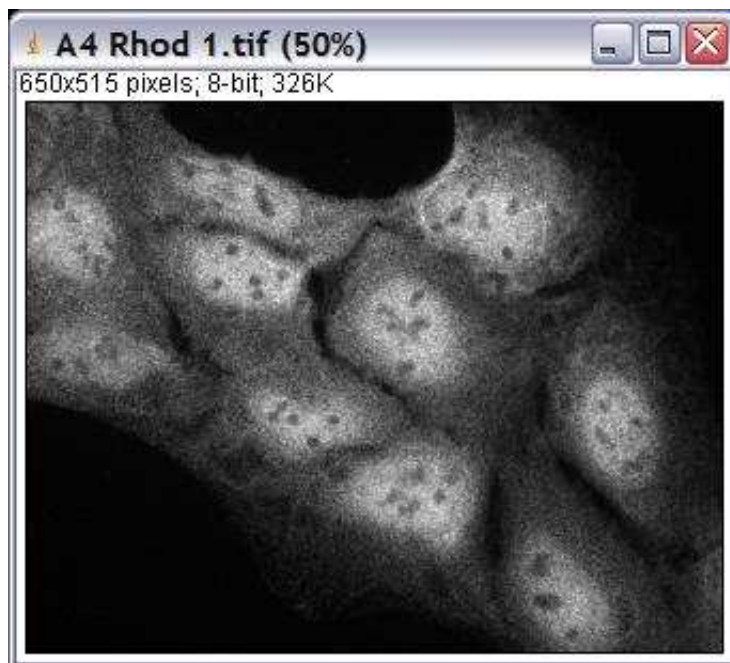
Operation: image calculation

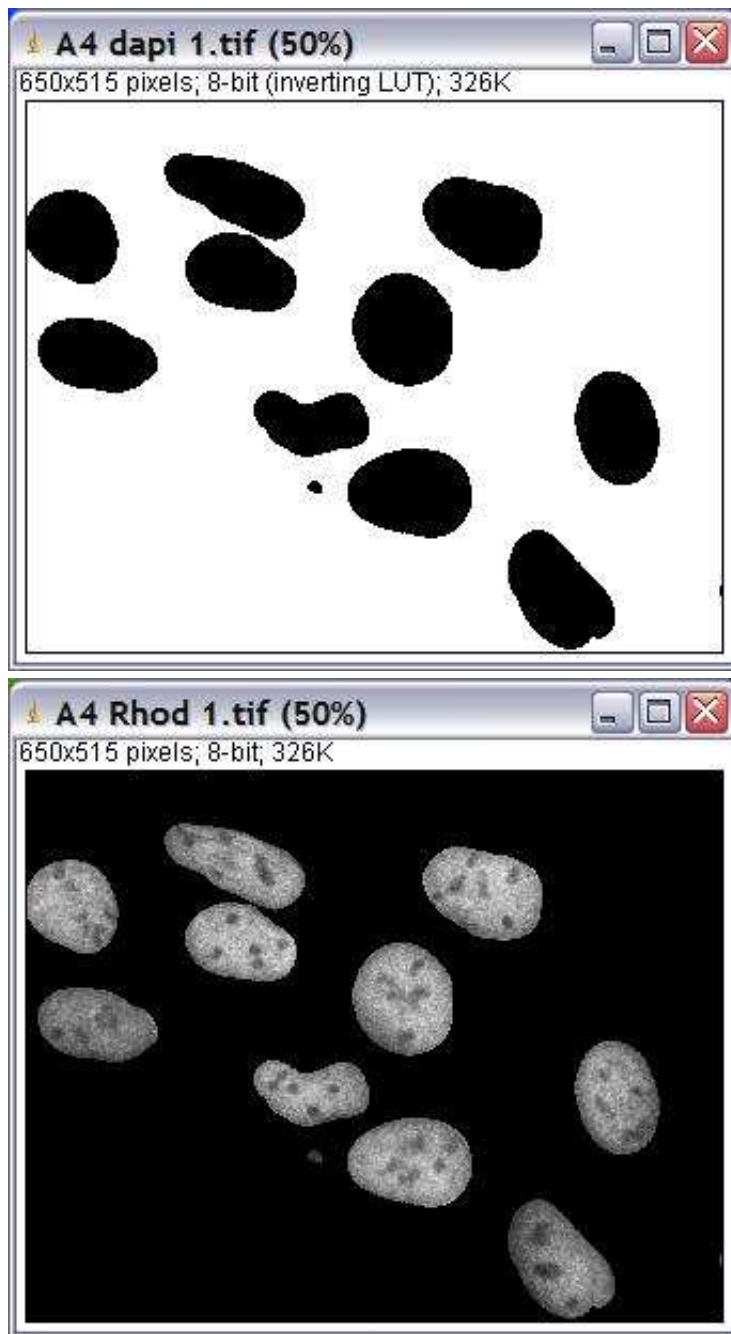


Author:

The operation uses the class ImageCalculator from ImageJ.

Example

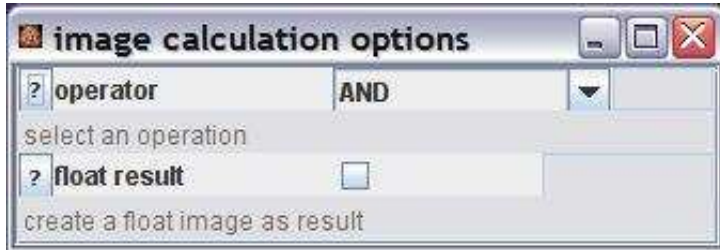




Description

The operation applies one of the operations "Add", "Subtract", "Multiply", "Divide", "AND", "OR", "XOR", "Min", "Max", "Average", "Difference" or "Copy" to the two input images, pixel by pixel.

Options



operator: Select the operator you want to apply to the two images.

float result: If float result is selected, the result image will be of type 32-bit and will therefore be able to represent floating point values..

Parameter

inputImage (ImagePlus): The first operand for the operation.

secondInputImage (ImagePlus): The second operand for the operation.

Results

The only result is the result image.

Operation: invert image



Author:

The operation uses the method `invert` of the class `ImageProcessor` from `ImageJ`.

Example



Description

The operation inverts the intensity values in the image.

Options

The operation has no options.

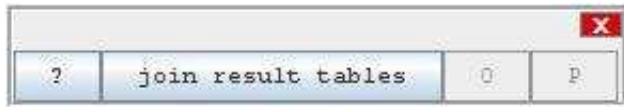
Parameter

The only parameter is the input image.

Results

The only result is the result image.

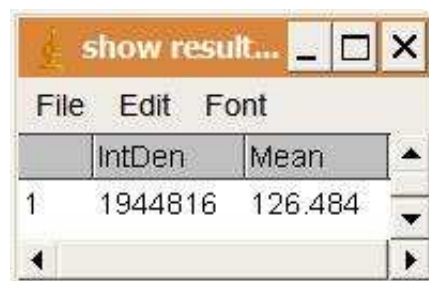
Operation: join result tables



Author:

Volker Bäcker

Example



Description

The operation joins the columns of two results tables.

Options

The operation has no options.

Parameter

firstResultsTable (ResultsTable): The first results table.

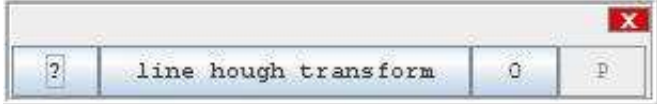
secondResultsTable (ResultsTable): The second results table.

The column names of the two input tables must be distinct.

Results

The result is a result table containing the data of the two input tables.

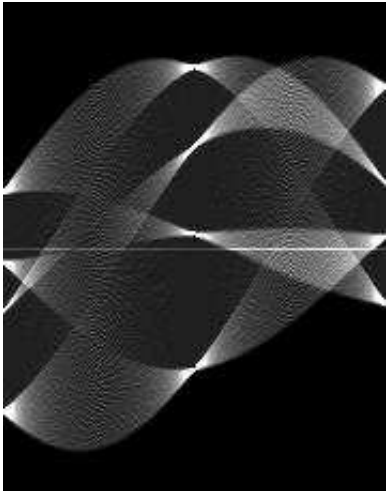
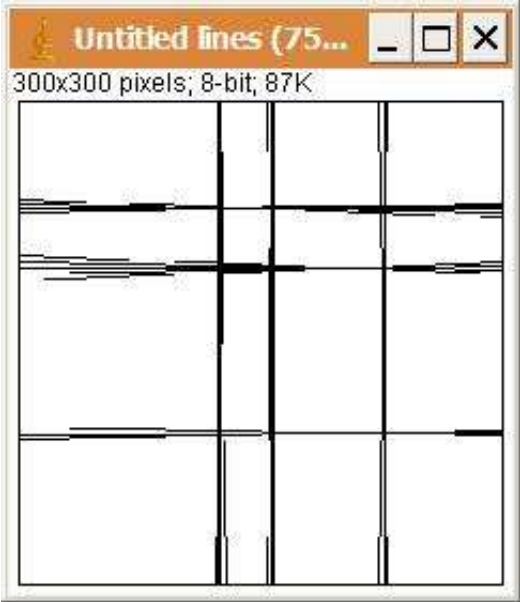
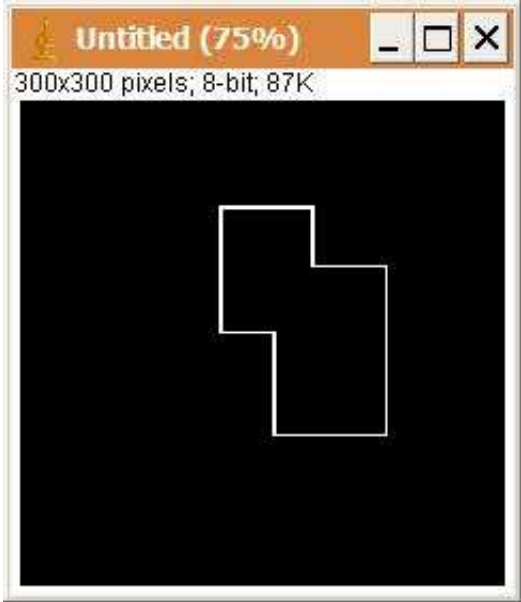
Operation: line hough transform



Author:

Based on the code of LineHoughTransformOp.java from the Generation5 jdk.

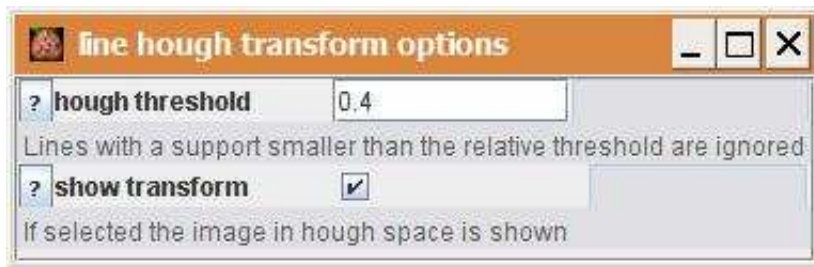
Example



Description

The operation calculates the line hough transform and answers lines with a support above a relative threshold value. The input image is supposed to be a greyscale line image.

Options



hough threshold: A threshold relative to the line with the biggest support. Only lines with a support above hough_threshold will be taken into account.

show_transform: If checked the hough transform of the image, i.e. the image in hough space is shown.

Parameter

The only parameter is the input image. It has to be a greyscale line image, i.e. an edge detecting filter has to be applied before using the line hough transform operation.

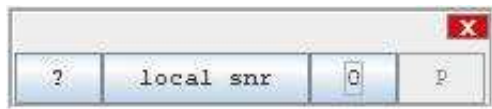
Results

result (ImagePlus): An image of the detected lines.

resultLines (ArrayList<Line2D>): A list of the detected lines.

houghTransform (ImagePlus): An image of the hough transform of the input image.

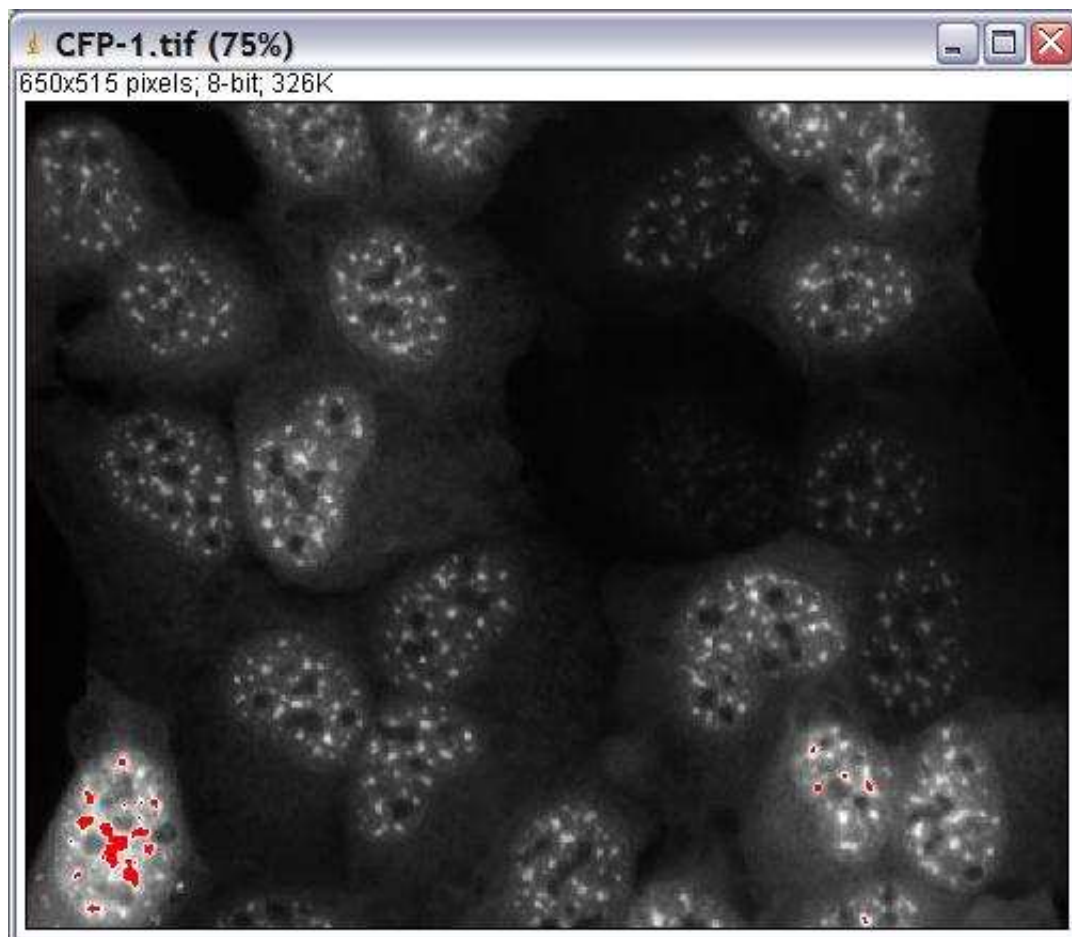
Operation: local snr

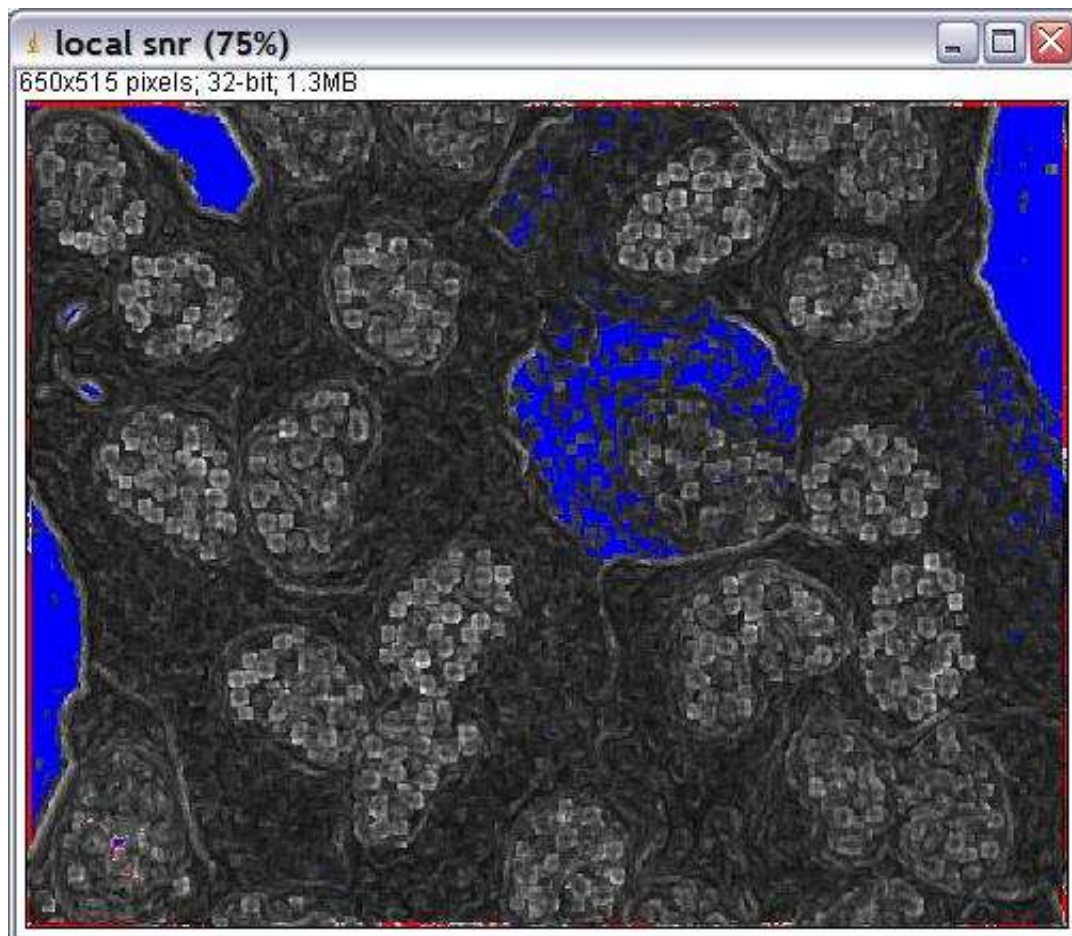


Author:

Volker Bäcker

Example:



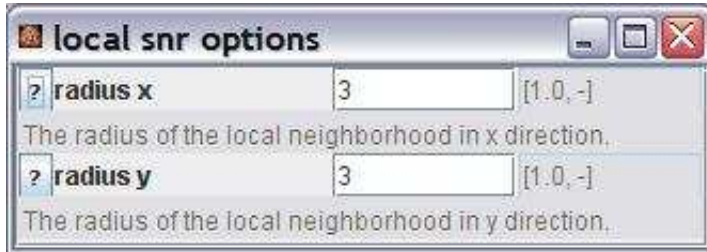


1	3.089

Description

Estimate the signal to noise ratio, defined as $snr = (\text{foreground mean} - \text{background mean}) / \text{background stdDev}$, for each pixel in the image, based on a region of a given size by applying an otsu-threshold to the region and counting pixels with an intensity above the threshold as foreground and other pixels as background.

Options



radius x: radius of the neighborhood in x-direction.
radius y: radius of the neighborhood in y-direction.

Parameter

The only parameter is the input image. The input image must be a 8-bit or 16-bit greyscale image.

Results

Result (ImagePlus) : A 32 bit image in which each pixel value is the snr at the corresponding position in the input image. If the background stdDev is 0 in a region the snr will be "not a number".
Median (ResultsTable): A results table containing the median snr of the image.

Operation: log



Author:

The operations uses the method `log` of the class `ImageProcessor` from `ImageJ`.

Example



Description

The operation replace each intensity value in the image by the natural logarithm of the value.

Options

The operation has no options.

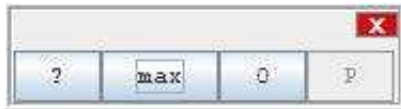
Parameter

The only parameter is the input image.

Results

The only result is the result image.

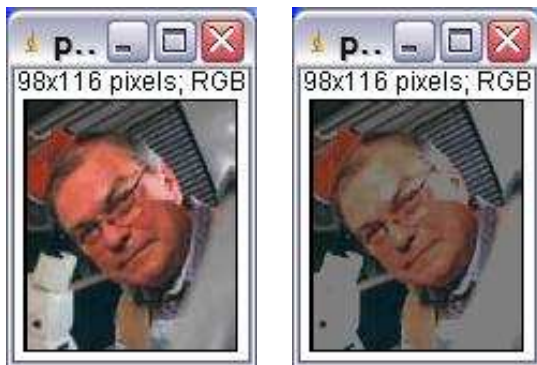
Operation: max



Author:

Uses the method max of the class ImageProcessor from ImageJ.

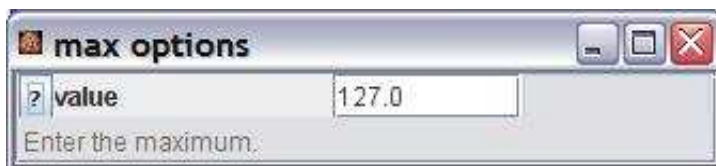
Example



Description

Intensity values greater than a constant value c will be set to c .

Options



value: The maximum value c . Values above c will be set to c .

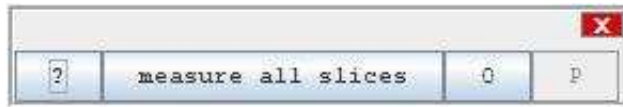
Parameter

The only parameter is the input image.

Results

The only result is the result image.

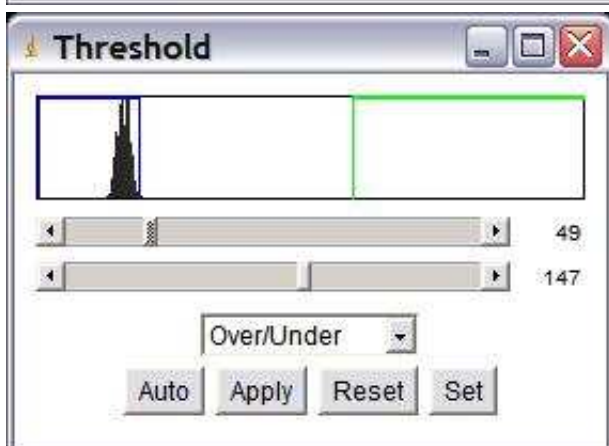
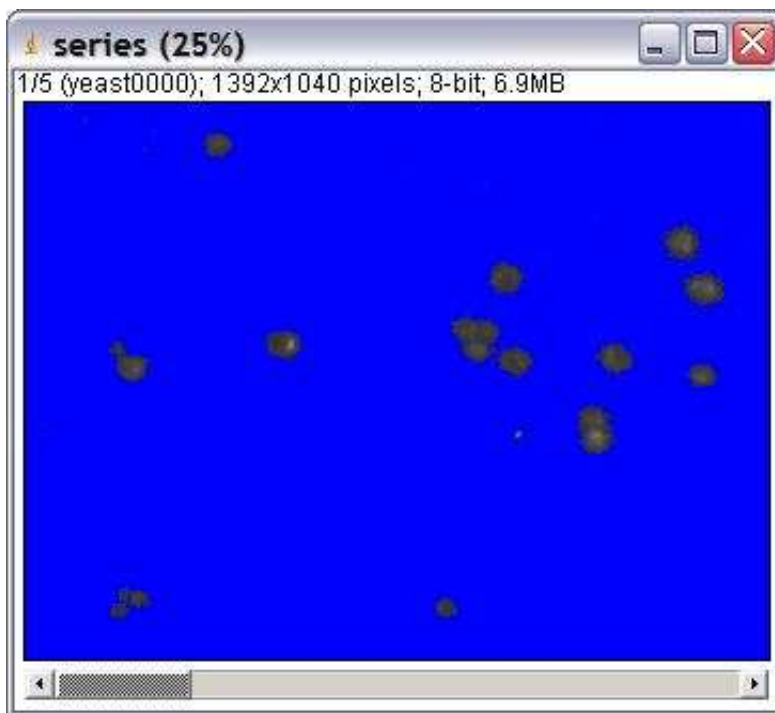
Operation: measure all slices



Author:

The operation uses the class Analyzer from ImageJ.

Example:



	Label	Area	Mean	StdDev	Mode	Min	Max	IntDen
1	series:yeast0000	37598	58.893	10.578	49	49	147	2214244
2	series:yeast0001	26734	59.296	10.549	49	49	147	1585225
3	series:yeast0002	25778	59.359	10.343	49	49	130	1530148
4	series:yeast0003	22918	55.982	6.955	49	49	91	1282988
5	series:yeast0004	16948	52.317	3.183	49	49	65	886671

Description

The operation measures all slices of the stack. It measures either each slice as a whole, the content of the selection on each slice or the pixels between min and max threshold on each slice, in case the limit threshold option is checked and a threshold is set using the threshold adjuster.

Options



measure area: Measure the area of the selection in square pixels. Areas are in calibrated units, such as square millimeters, if *Analyze>Set Scale* was used to spatially calibrate the image.

measure bounds: Measure the smallest rectangle enclosing the selection. Uses the headings *BX*, *BY*, *Width* and *Height*, where *BX* and *BY* are the coordinates of the upper left corner of the rectangle.

measure center of mass: This is the brightness-weighted average of the x and y coordinates of all pixels in the selection. Uses the *XM* and *YM* headings. These coordinates are the first order spatial moments.

measure centroids: The center point of the selection. This is the average of the x and y coordinates of all of the pixels in the image or selection. Uses the *X* and *Y* Results table headings.

measure circularity: $4\pi(\text{area}/\text{perimeter}^2)$. A value of 1.0 indicates a perfect circle. As the value approaches 0.0, it indicates an increasingly elongated polygon. Values may not be valid for very small particles.

Feret's diameter: The longest distance between any two points along the selection boundary. Also known as the caliper length.

measure fit ellipse: Fit an ellipse to the selection. Uses the headings *Major*, *Minor* and *Angle*. *Major* and *Minor* are the primary and secondary axis of the best fitting ellipse. *Angle* is the angle between the primary axis and a line parallel to the x-axis of the image. Note that ImageJ cannot calculate the major and minor axis lengths if *Pixel Aspect Ratio* in the *Set Scale* dialog is not 1.0.

integrated density: The sum of the values of the pixels in the selection.

measure mean: The average gray value within the selection.

measure min & max: Minimum and maximum gray values within the selection.

modal gray value: Most frequently occurring gray value within the selection. Corresponds to the highest peak in the histogram.

measure perimeter: The length of the outside boundary of the selection.

standard deviation: Standard deviation of the gray values used to generate the mean gray value.

display labels: If checked the title of the image is displayed in the results table.

invert y: Inverts the y-coordinate so that the origin is in the lower left corner and not in the upper left corner anymore.

limit to threshold: If checked, only thresholded pixels are included in measurement calculations. Use *Image>Adjust>Threshold* to set the threshold limits.

Parameter

The only parameter is the input image.

Results

Measurements (ResultsTable): The results table containing the measurements of the selection on each slice of the stack.

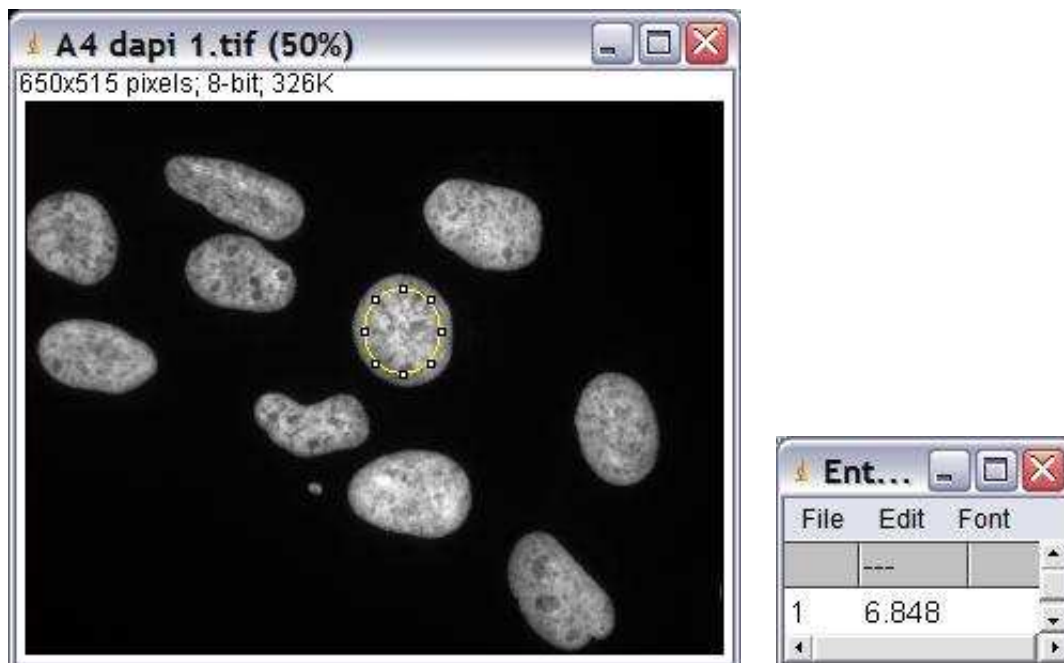
Operation: measure entropy



Author:

Dimitri Vibert

Example



Description

Calculates the image-entropy within the image or the selection. The image must be 8-bit.

Options

The operation has no options.

Parameter

The only parameter is the input image. It must be an 8-bit image.

Results

measurements (ResultsTable): The results table containing the measured image-entropy.

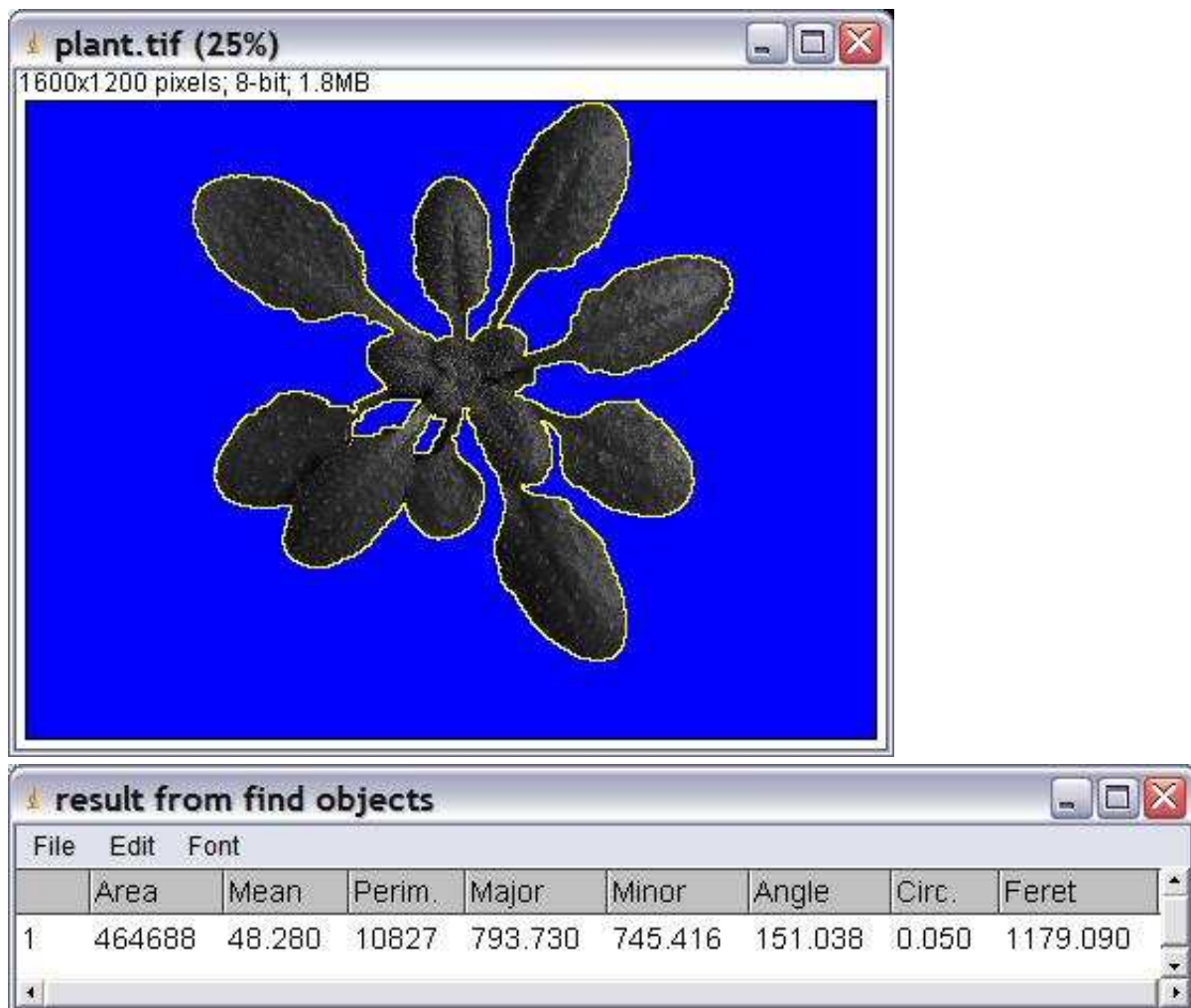
Operation: measure



Author:

This operation calls the Analyzer from ImageJ.

Example



The example shows two windows from ImageJ. The top window, titled 'plant.tif (25%)', displays a grayscale image of a plant with a yellow outline on a blue background. The window title also indicates '1600x1200 pixels; 8-bit; 1.8MB'. The bottom window, titled 'result from find objects', shows a table of measurement results for the plant object.

	Area	Mean	Perim.	Major	Minor	Angle	Circ.	Feret
1	464688	48.280	10827	793.730	745.416	151.038	0.050	1179.090

Description

The operation measures the features of an image or a selection.

Options



measure area: Measure the area of the selection in square pixels. Areas are in calibrated units, such as square millimeters, if *Analyze>Set Scale* was used to spatially calibrate the image.

measure bounds: Measure the smallest rectangle enclosing the selection. Uses the headings *BX*, *BY*, *Width* and *Height*, where *BX* and *BY* are the coordinates of the upper left corner of the rectangle.

measure center of mass: This is the brightness-weighted average of the x and y coordinates of all pixels in the selection. Uses the *XM* and *YM* headings. These coordinates are the first order spatial moments.

measure centroids: The center point of the selection. This is the average of the x and y coordinates of all of the pixels in the image or selection. Uses the *X* and *Y* Results table headings.

measure circularity: $4\pi(\text{area}/\text{perimeter}^2)$. A value of 1.0 indicates a perfect circle. As the value approaches 0.0, it indicates an increasingly elongated polygon. Values may not be valid for very small particles.

Feret's diameter: The longest distance between any two points along the selection boundary. Also known as the caliper length.

measure fit ellipse: Fit an ellipse to the selection. Uses the headings *Major*, *Minor* and *Angle*. *Major* and *Minor* are the primary and secondary axis of the best fitting ellipse. *Angle* is the angle between the primary axis and a line parallel to the x-axis of the image. Note that ImageJ cannot calculate the major and minor axis lengths if *Pixel Aspect Ratio* in the *Set Scale* dialog is not 1.0.

integrated density: The sum of the values of the pixels in the selection.
measure mean: The average gray value within the selection.
measure min & max: Minimum and maximum gray values within the selection.
modal gray value: Most frequently occurring gray value within the selection. Corres The only parameter is the input image. ponds to the highest peak in the histogram.
measure perimeter: The length of the outside boundary of the selection.
standard deviation: Standard deviation of the gray values used to generate the mean gray value.
display labels: If checked the title of the image is displayed in the results table.
invert y: Inverts the y-coordinate so that the origin is in the lower left corner and not in the upper left corner anymore.
limit to threshold: If checked, only thresholded pixels are included in measurement calculations. Use *Image>Adjust>Threshold* to set the threshold limits.

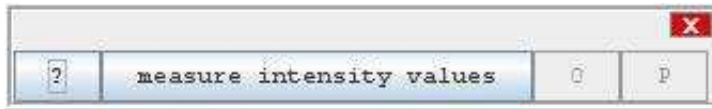
Parameter

The only parameter is the input image. If there is a selection on the input image, only the selection is measured.

Results

Measurements (ResultsTable): The results table containing the measurements of the image or selection.

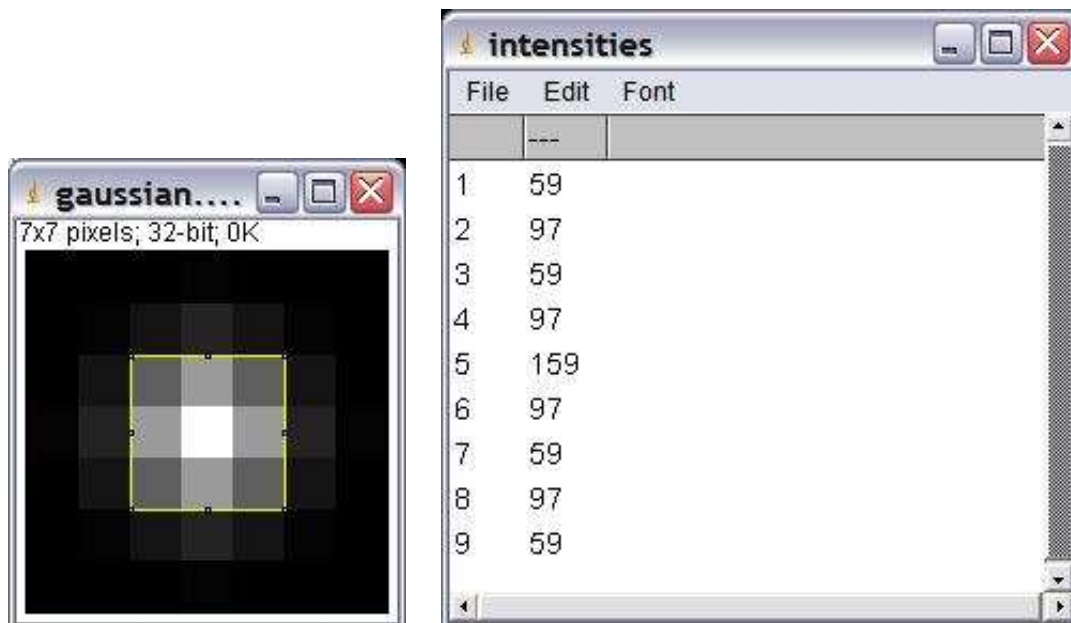
Operation: measure intensity values



Author:

Volker Bäcker

Example



Description

The operation writes a list of all intensity values in the image or in the selection into a results table. The values are written from left to right, column by column.

Options

The operation has no options.

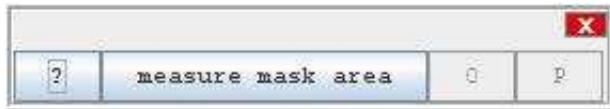
Parameter

The only parameter is the input image. The input image must be a 32-bit image (float image).

Results

The only result is the results table, containing the intensity values of the pixels in the image or selection.

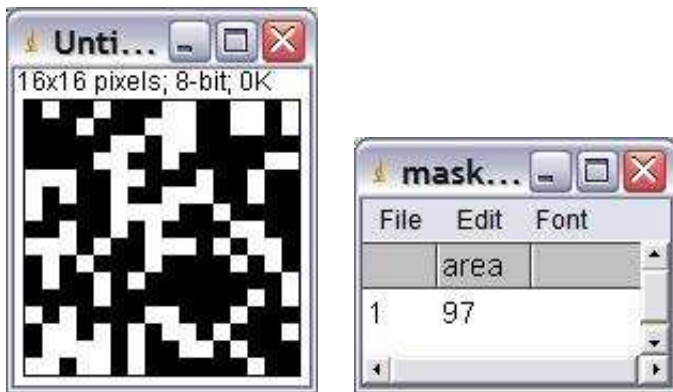
Operation: measure mask area



Author:

Volker Bäcker

Example



Description

Counts the pixel with value 255 in a mask (binary image (0 and 255)).

Options

The operation has no options.

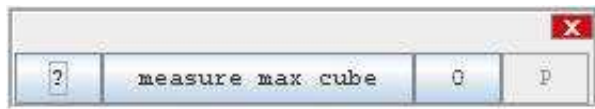
Parameter

The only parameter is the input image. It must be a mask (binary image) with the values 0 and 255 exclusively.

Results

The only result is the results table with the area of the mask.

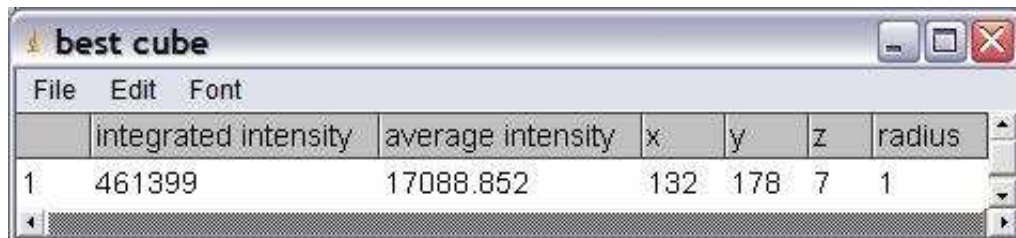
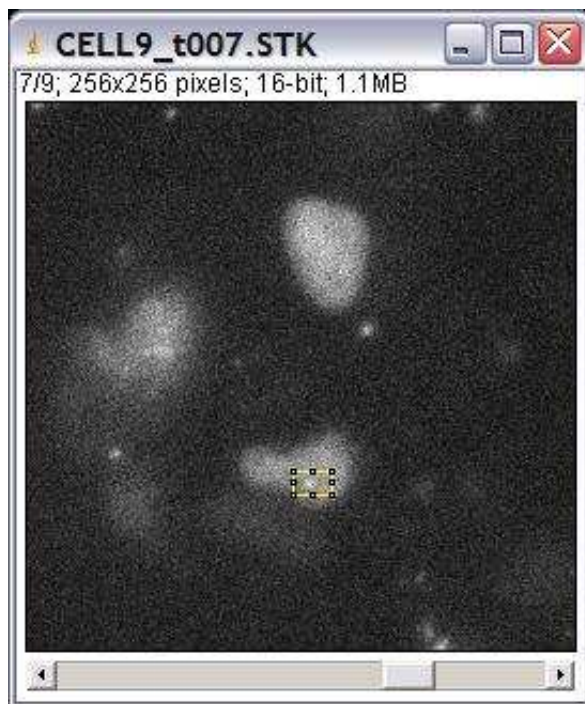
Operation: measure max cube



Author:

Volker Bäcker

Example

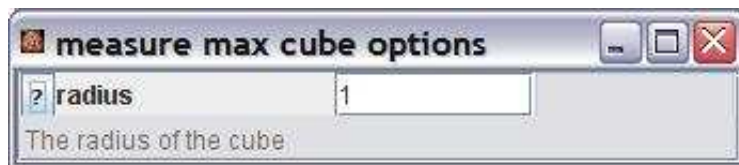


	integrated intensity	average intensity	x	y	z	radius
1	461399	17088.852	132	178	7	1

Description

The operation finds the brightest cube of radius r within a rectangular selection and measures its integrated intensity and its average intensity.

Options



radius: The radius of the cube. A radius 1 defines a cube of 27 voxels.

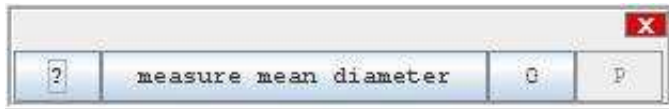
Parameter

The only parameter is the input image. The image must be an 8-bit or 16-bit greyscale image.

Results

The only result is a results table containing the integrated density, the average intensity, the x,y and z coordinates of the start point and the radius of the cube.

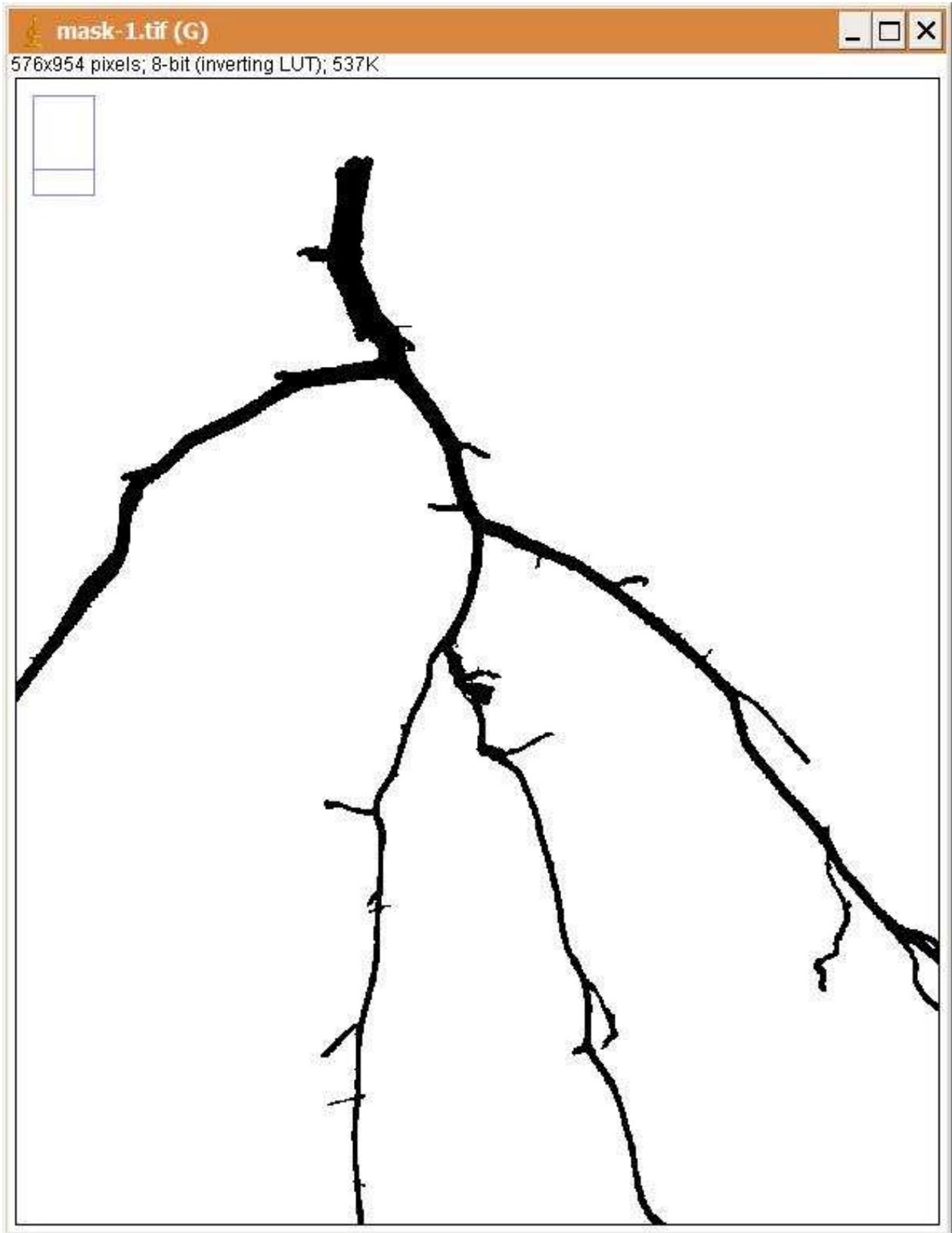
Operation: measure mean diameter



Author:

Volker Bäcker

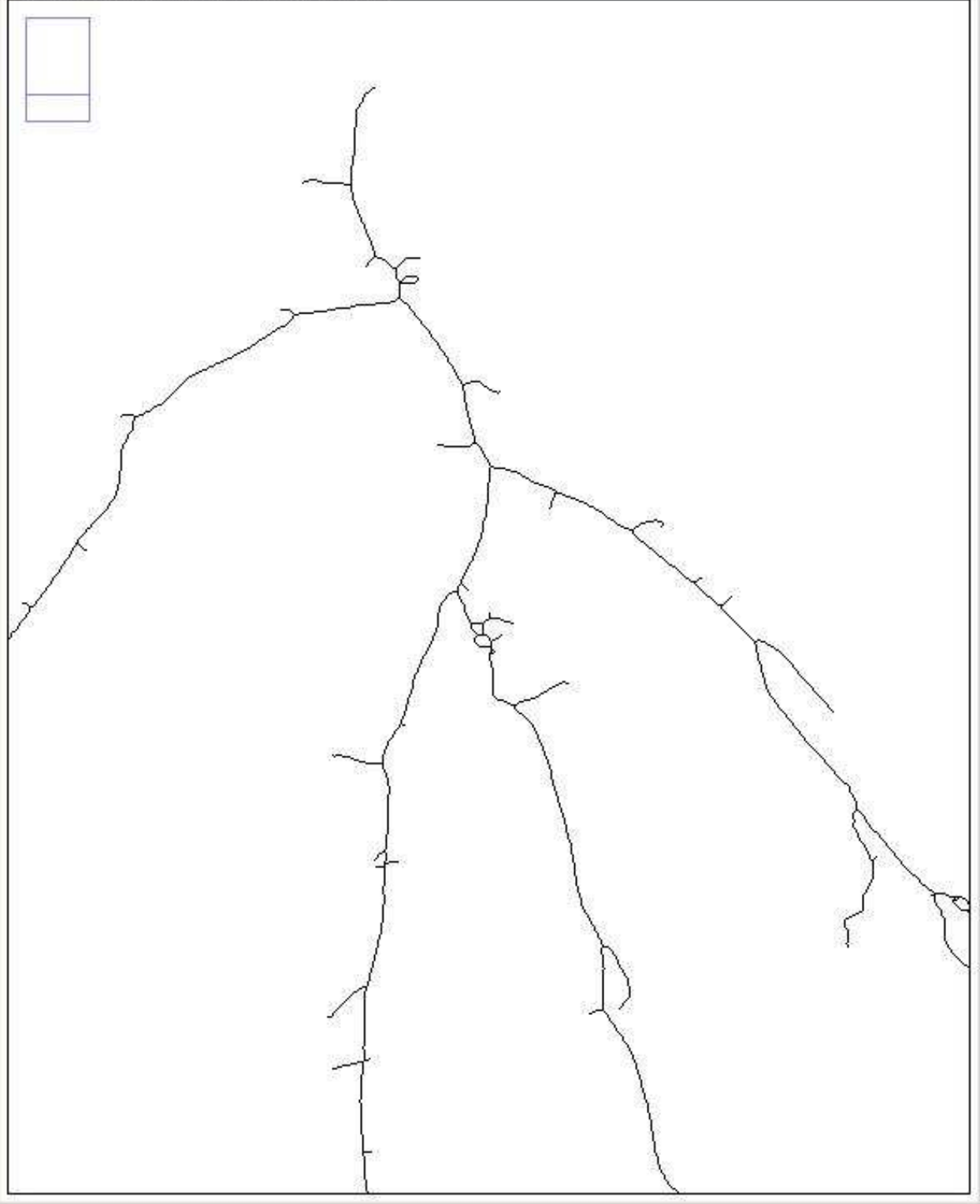
Example



skeleton-1.tif (G)

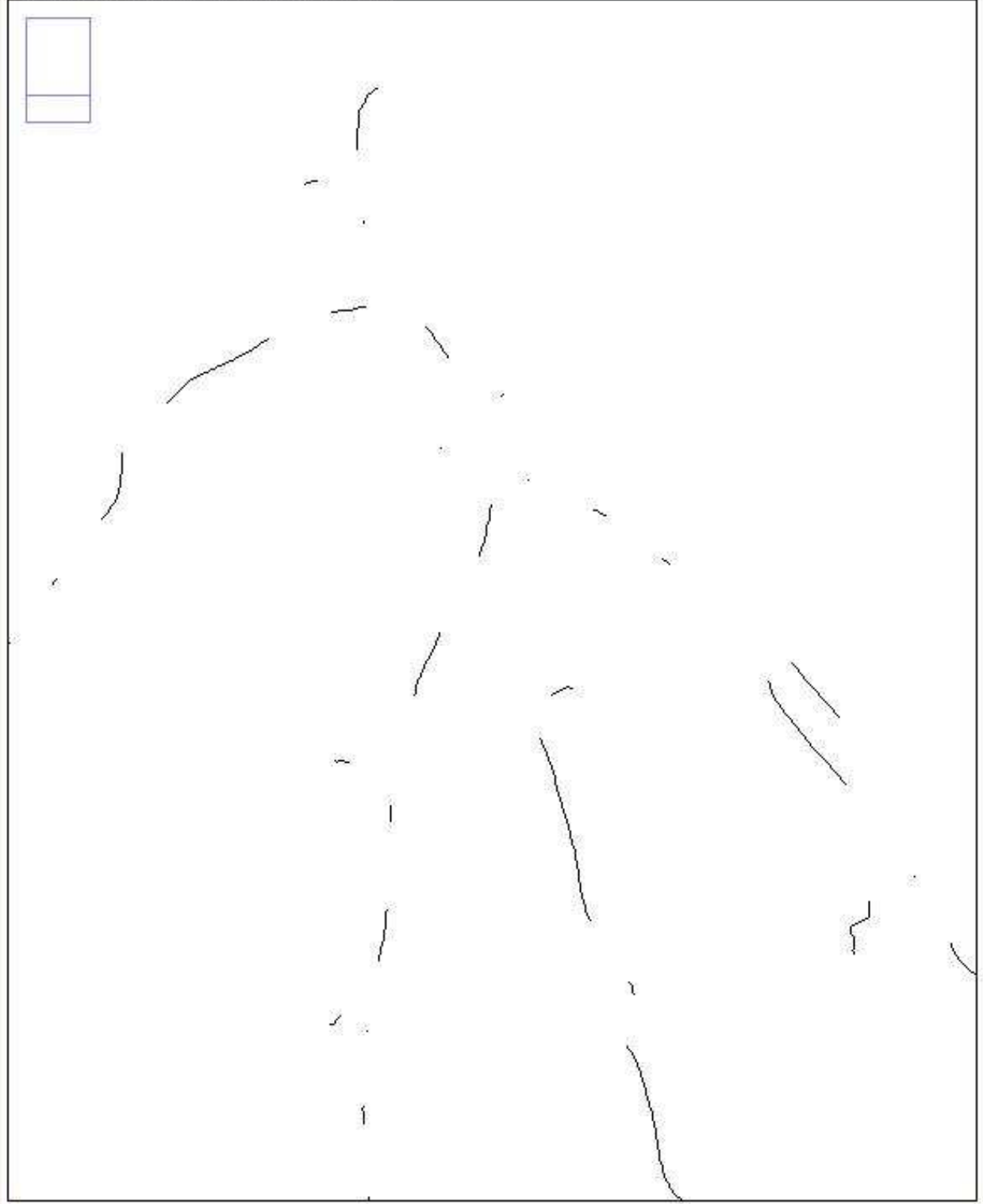


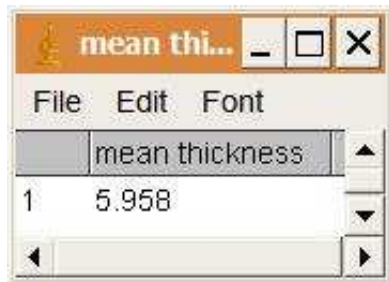
576x954 pixels; 8-bit (inverting LUT); 537K



disconnected skeleton-1.tif (G) - □ ×

576x954 pixels; 8-bit (inverting LUT); 537K

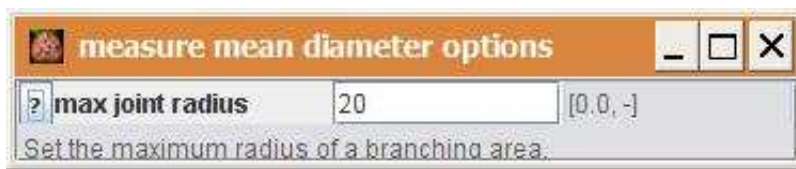




Description

The operation takes a mask and a skeleton and estimates the mean diameter of the structure in the mask image. Areas around branching points are not taken into account.

Options



max joint radius: The radius of the circle that will be deleted around each branching point.

Parameter

inputImage (ImagePlus): The mask of the structure to be measured.

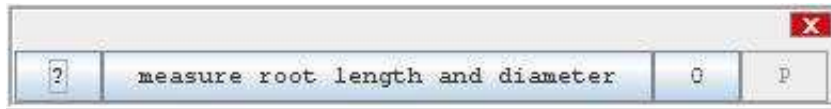
skeleton (ImagePlus): A skeleton of the input mask.

Results

result (ImagePlus): An image that indicates the parts of the mask that have been taken into account for the calculation.

meanThickness (ResultsTable): A results table containing the estimated mean thickness of the structure in the mask image.

Operation: measure root length and diameter



Author:

Volker Bäcker

Example

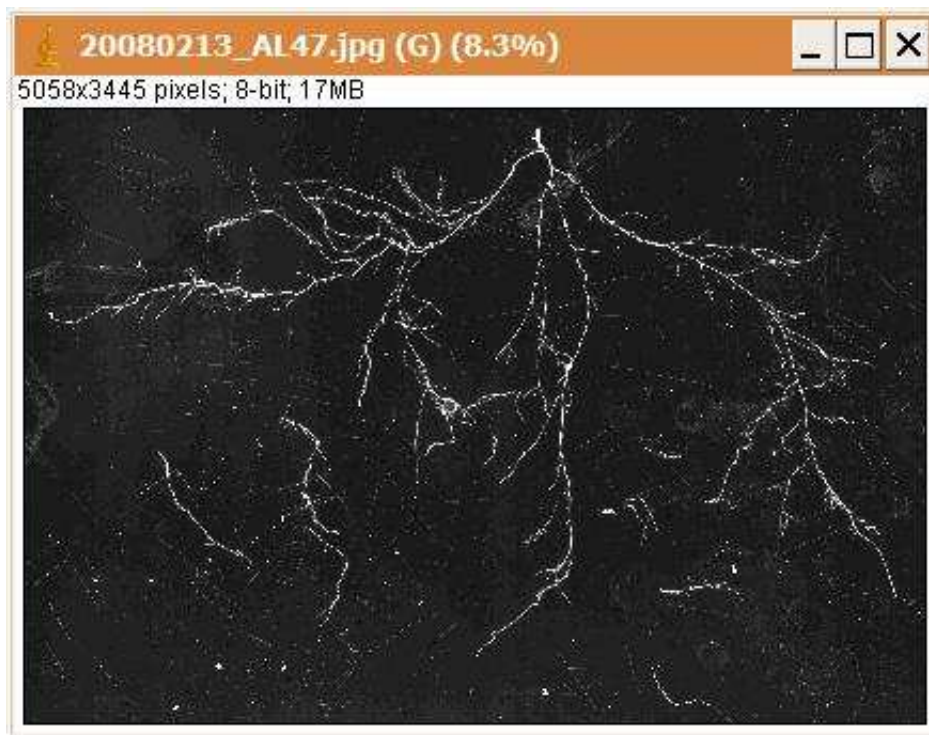


image	number of objects	length	mean thickness	folder
20080213_AL47.jpg	1	63281,01	5,05	Z:\baecker\muller\marie\

Description

The application measures the total length and the average thickness of filament like structures.

When the application is started the list-editor is opened. Press the add button and select the files you want to measure. Press close on the list editor to start. You will be asked where to save the result spreadsheet file. The image is opened and displayed. You might eventually need to inverse the contrast, so that high pixel values are bright. You can do this by activating the image and pressing SHIFT+i. Open the contrast-adjuster (SHIFT-t) and adjust the contrast. Press apply on the threshold-adjuster. You can now

clean up the image by making selections and using the fill or clear command from the menu Edit. When you finished cleaning up, press the continue button.

Options

To change options either press the O-button of the application or open the application (right click on the top of the application tile and select open from the context menu) and press the O-button of one operation of the application.

Option	Value / State
image list	edit
use sequence opener	<input type="checkbox"/>
replace string	
replace with	null
min size	500
max size	999999999
display labels	<input type="checkbox"/>
exclude edge objects	<input type="checkbox"/>
invert y	<input type="checkbox"/>
limit to threshold	<input type="checkbox"/>
measure area	<input type="checkbox"/>
measure bounds	<input type="checkbox"/>
measure center of mass	<input type="checkbox"/>
measure centroids	<input type="checkbox"/>
measure circularity	<input type="checkbox"/>
Feret's diameter	<input type="checkbox"/>
measure fit ellipse	<input type="checkbox"/>
integrated density	<input type="checkbox"/>
measure mean	<input type="checkbox"/>
measure min & max	<input type="checkbox"/>
modal gray value	<input type="checkbox"/>
measure perimeter	<input type="checkbox"/>
standard deviation	<input type="checkbox"/>
max joint radius	20 [0.0, -]
output path	ickerBureau\repc browse
output folder	control browse
create in source folder	<input checked="" type="checkbox"/>
name addition	
add loop index to name	<input type="checkbox"/>

Important options are the min size and max size of the find objects operation. The min size should be smaller than the smallest object you want to keep. The max size must be bigger than the size of the biggest object you want to keep. Another important option is the max joint radius of the measure mean diameter operation. Set it to a value, so that branching areas are completely for the calculation of the thickness.

Results

A control image showing the skeleton is saved in the subfolder control. Length and thickness will be written to a spreadsheet file. The length is calculated using the measure skeleton length operation.

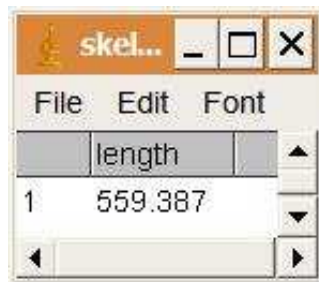
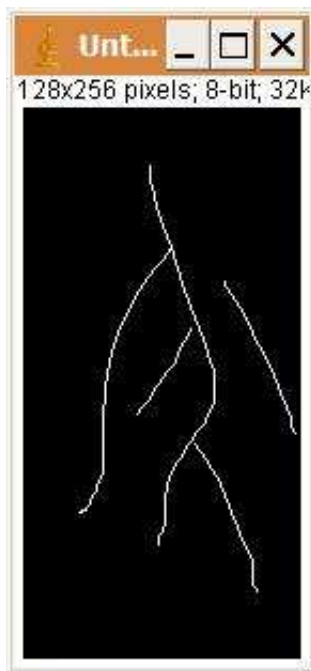
Operation: measure skeleton length



Author:

Volker Bäcker

Example



square-pixel.

Remark that the area in the example is 488

Description

Measure the length of a skeleton using the algorithm from "Robust Quantification of In Vitro Angiogenesis Through Image Analysis", published in IEEE Transactions on Medical Imaging Vol24, No.4, April 2005, as described in the ImageJ mailing list by Michael Miller. The aspect ratio of a pixel is considered to be one. Lengths are calculated from the middle of one pixel to the middle of another pixel. Diagonal steps are by a factor of $\sqrt{2}$ longer than horizontal or vertical steps.

Options

The operation has no options.

Parameter

The only parameter is the input image. The input image must be a skeleton. The background value must be zero and the skeleton value 255.

Results

The result is a results table with the measured length. The result is either in pixel or in the defined unit, if a scale is set.

Operation: median filter



Author:

The operation uses the class RankFilter from ImageJ.

Example



The source image has been despeckled with the radius 1.

Description

Applies a median filter with the given radius. The median is the value that divides the pixel intensities in the radius into two groups of the same size. The value of the central pixel is set to the median value.

Options

<screenshot options>

<anchor><option1>: <short description>

<anchor><option2>: <short description>

<anchor><option3>: <short description>

...

Parameter

<describe the parameters>

Results

<describe the results>

Operation: merge channels



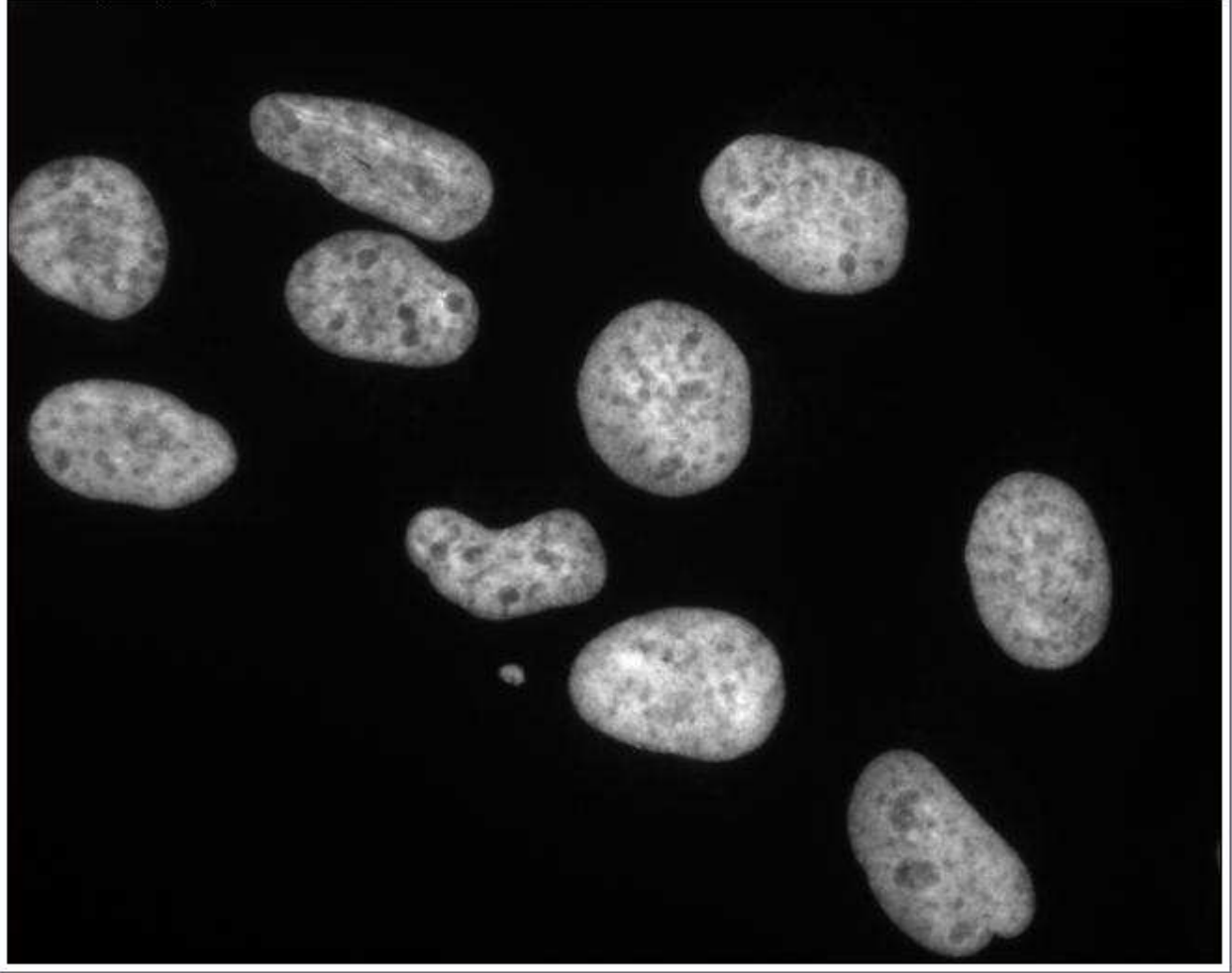
Author:

The operation uses the class `RGBStackMerge` from ImageJ.

Example

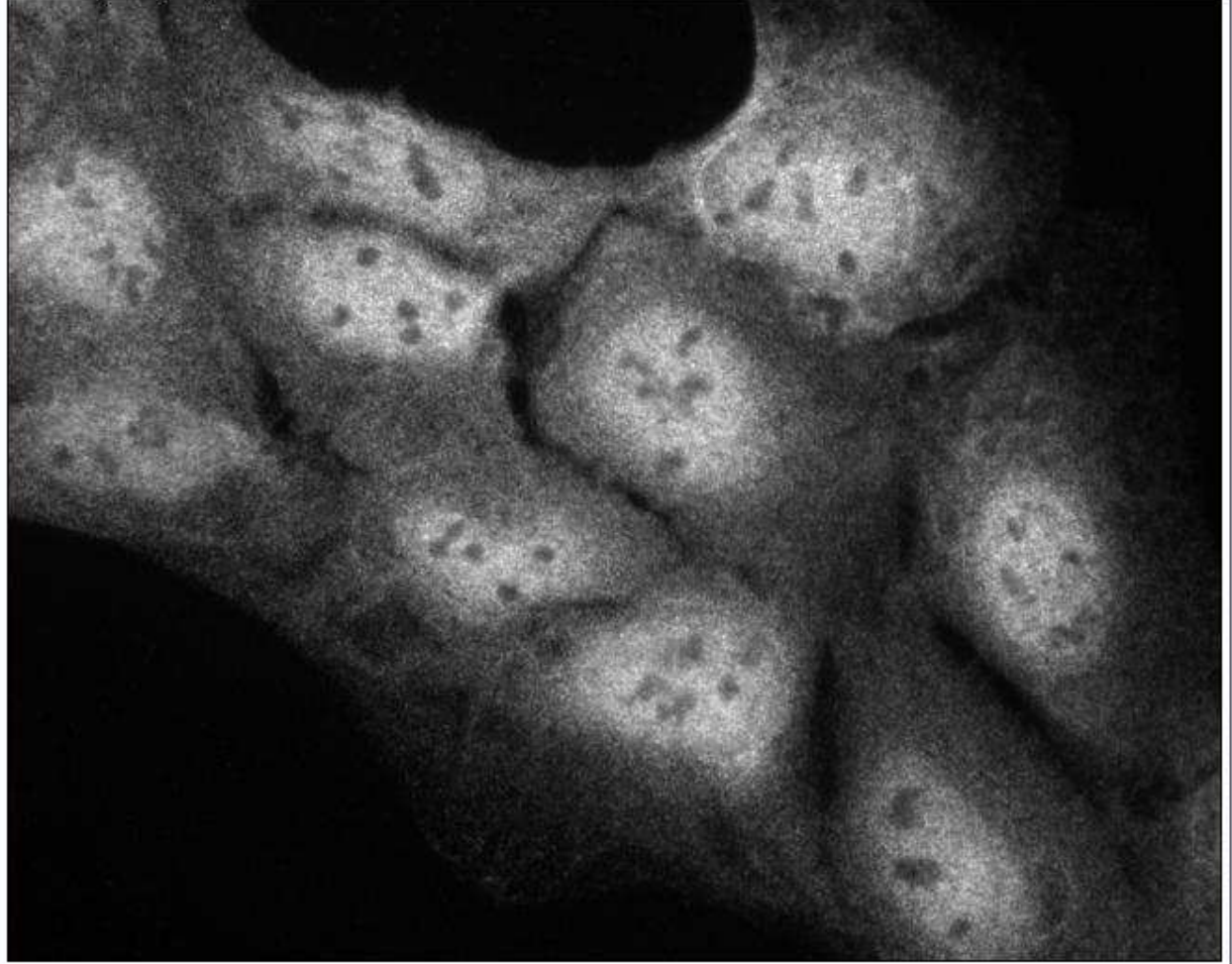
A4 dapi 1.tif

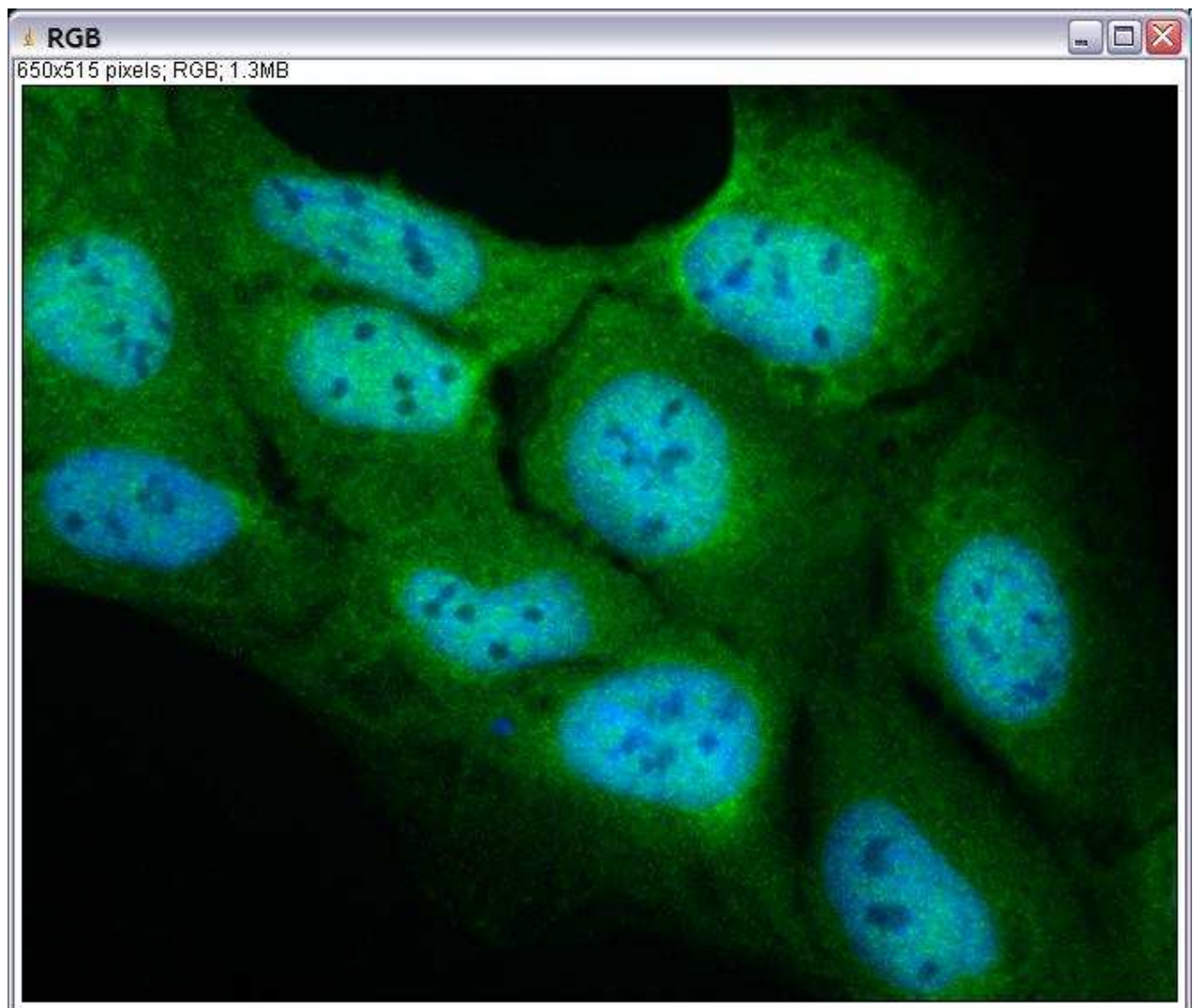
650x515 pixels; 8-bit; 326K



A4 Rhod 1.tif

650x515 pixels; 8-bit; 326K





Description

The operation merges two or three greyscale images into an RGB image.

Options



red channel: Choose the image for the red channel.

green channel: Choose the image for the green channel.

blue channel: Choose the image for the blue channel.

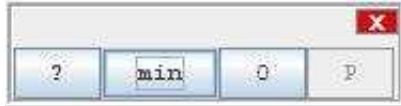
Parameter

Two or three greyscale input images.

Results

The merged RGB image.

Operation: min



Author:

The operation uses the method `min` of the class `ImageProcessor` from `ImageJ`.

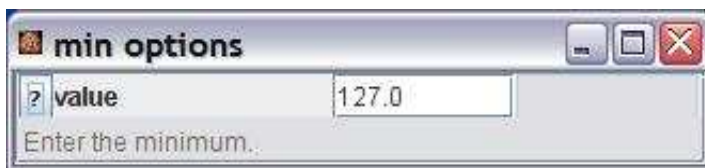
Example



Description

The operation replaces each intensity value below `min` with `min`.

Options



value: The minimum value. Intensities below value will be replaced by value.

Parameter

The only parameter is the input image.

Results

The only result is the result image.

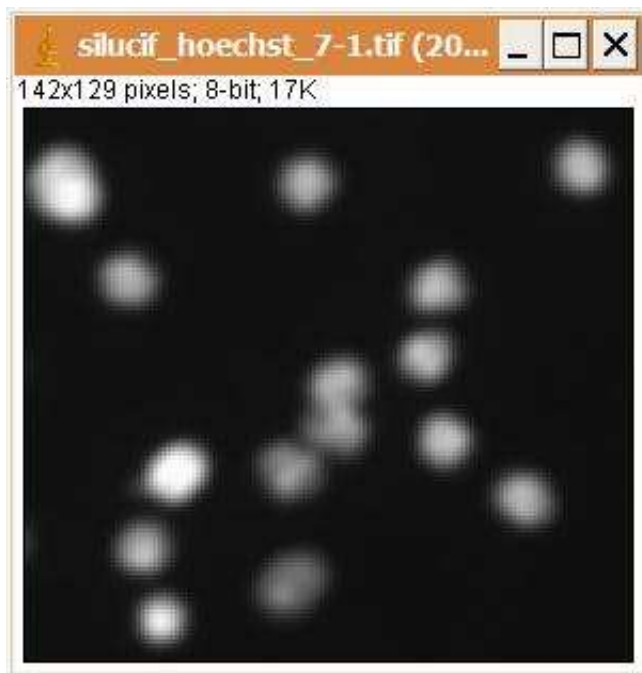
Operation: morpho contrast enhance

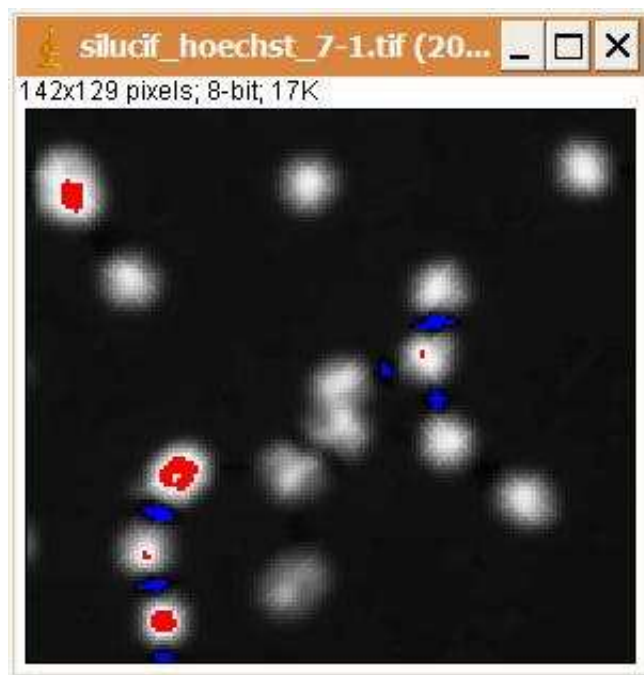


Author:

The operation calls the Grayscale Morphology plugin by Dimitar Prodanov.

Example

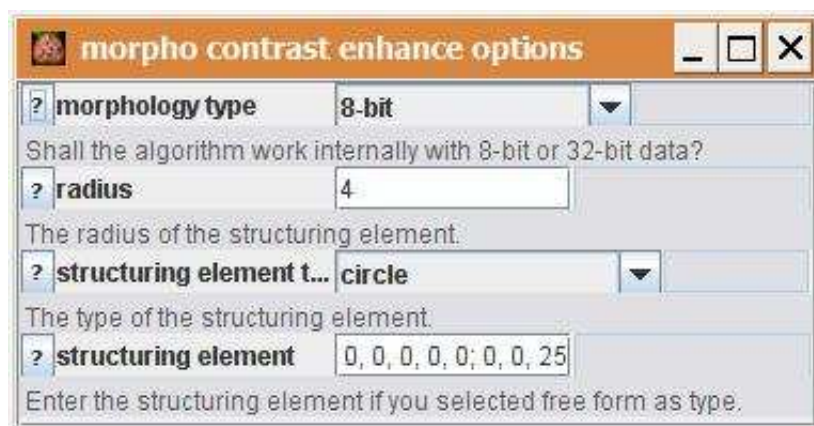




Description

The operation enhances the contrast of an image by adding the result of the top-hat filter and subtracting the result of the bottom hat filter.

Options



morphology type: Choose whether the 8-bit or the 32-bit morphology plugins are used.

radius: The radius of the structuring element.

structuring element type: Select the form of the structuring element. If free form is selected the structuring element can be entered manually.

structuring element: Enter the structuring element directly. This is used when free form is selected in the field structuring element type.

Parameter

The only parameter is the input image.

Results

The only result is the result image. The result image is a 32-bit or an 8-bit image, depending on the morphology type used.

Operation: multiply



Author:

The operation uses the method multiply of the class ImageProcessor from ImageJ.

Example



Description

The operation multiplies each intensity value in the image with a constant factor.

Options



value: The constant factor with which each intensity value in the image will be multiplied.

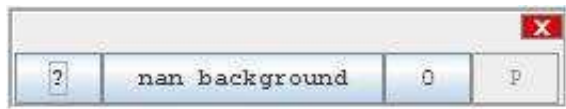
Parameter

The only parameter is the input image

Results

The only result is the result image.

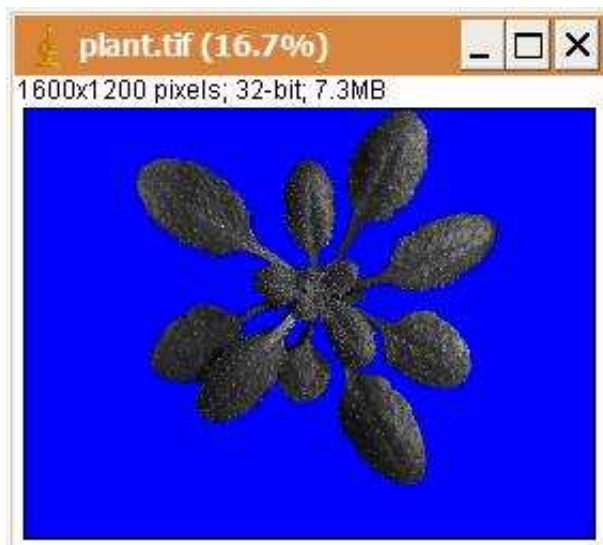
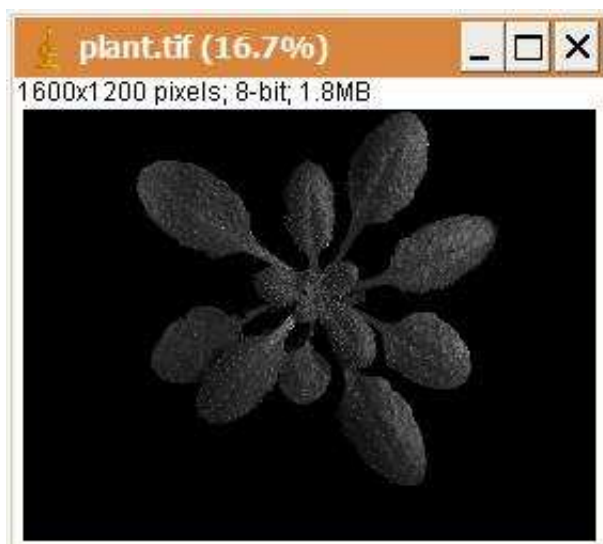
Operation: nan background



Author:

Uses the command NaN Background from the menu Process>Math in ImageJ.

Example



Description

Sets all intensities below min and all intensities above max to the special floating point value NaN (not a number). If the image is not a 32-bit image it is automatically converted to 32-bit.

Options



min: The minimum value. All intensities below will be set to NaN.

max: The maximum value. All intensities above will be set to NaN.

Parameter

The only parameter is the input image.

Results

The result image is a 32-bit image.<describe the results>

Operation: n times repeat end

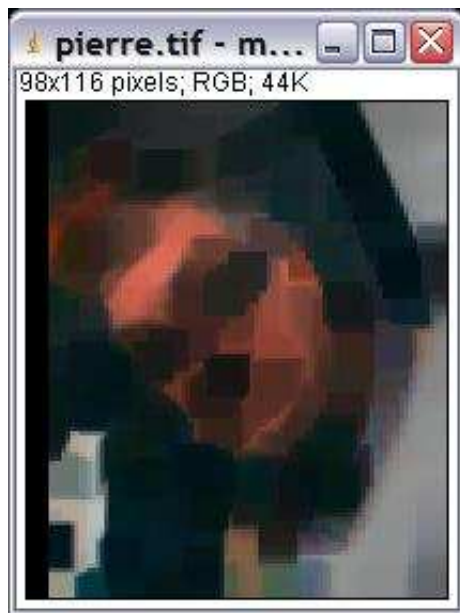


Author:

Volker Bäcker

Example





Description

The operation marks the end of a n-times repeat loop.

Options

The operation has no options.

Parameter

nTimesRepeatOperation (NTimesRepeatOperation): The operation that marks the start of the n-times-repeat loop.

Results

The operation has no result.

Operation: n times repeat

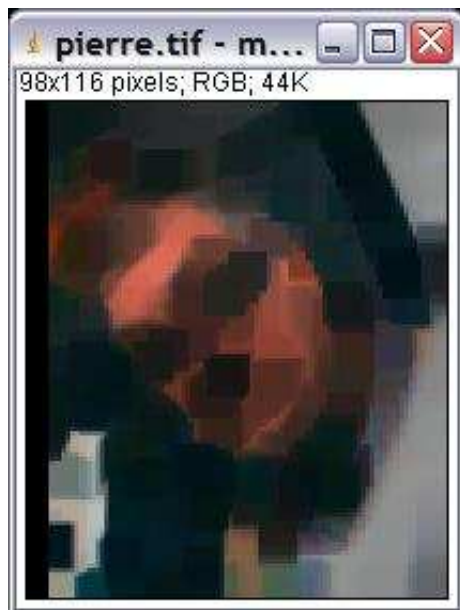


Author:

Volker Bäcker

Example





Description

The operation marks the start of a n-times repeat loop. The operations between the start and the end of the loop are repeated n times.

Options



number of repetitions: The number of times the operations within the loop are repeated.

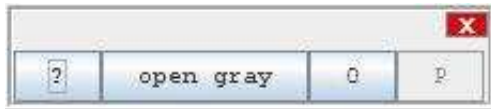
Parameter

The operation has no parameters.

Results

nTimesRepeatOperation (NTimesRepeatOperation): The operation itself. The operation marking the end of the loop needs this result as input.

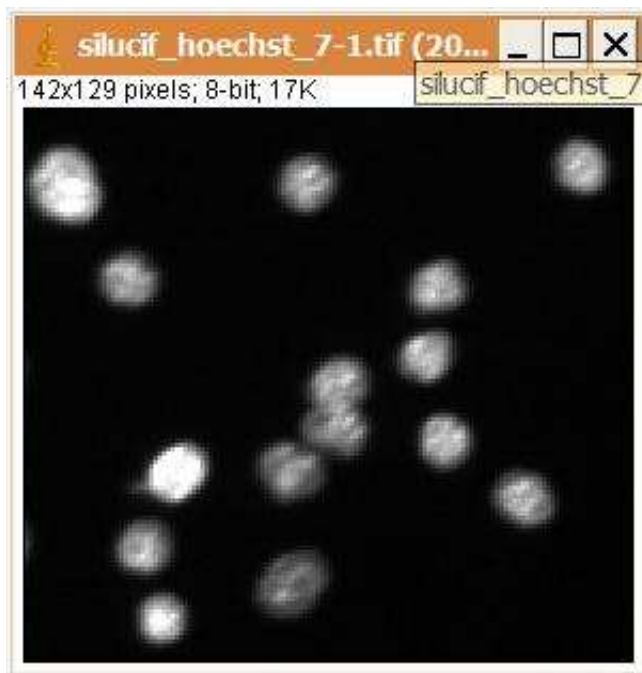
Operation: open gray

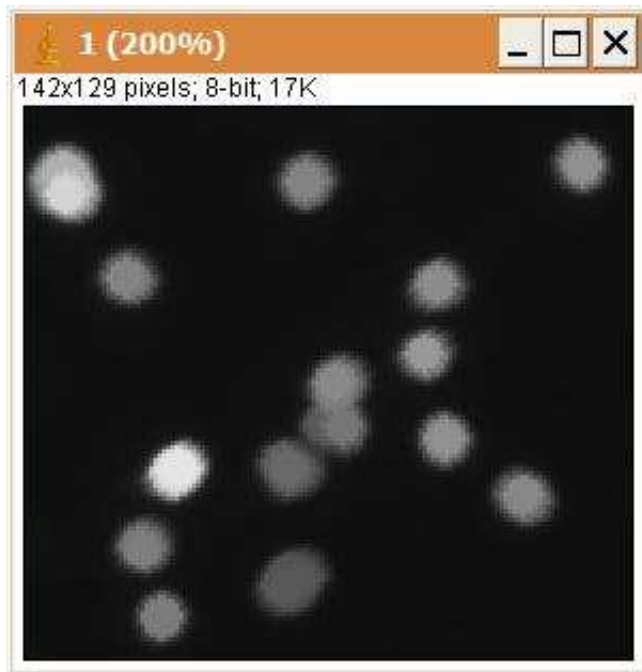


Author:

The operation calls the Grayscale Morphology plugin by Dimitar Prodanov.

Example

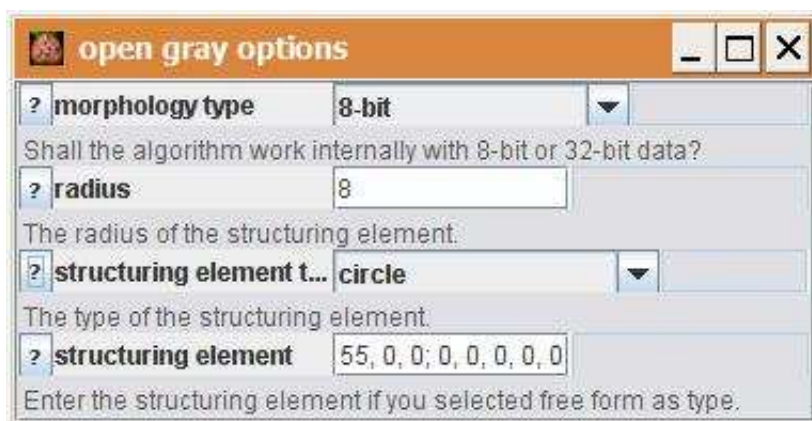




Description

The operation opens the grayscale image with a structuring element. It shrinks the foreground while keeping regions that have a similar shape as the structuring element or that can completely contain the structuring element. An open is an erosion followed by a dilation.

Options



morphology type: Choose whether the 8-bit or the 32-bit morphology plugins are used.

radius: The radius of the structuring element.

structuring element type: Select the form of the structuring element. If free form is selected the structuring element can be entered manually.

structuring element: Enter the structuring element directly. This is used when free form is selected in the

field structuring element type.

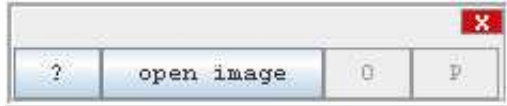
Parameter

The only parameter is the input image.

Results

The only result is the result image. The result image is a 32-bit or an 8-bit image, depending on the morphology type used.

Operation: open image



Author:

The operation uses ImagePlus to load an image and the OpenFileDialog from ImageJ to browse for the filename.

Description

The operation has a filename as parameter and loads the image into memory. File formats understood are:

- TIFF (uncompressed)
- GIF
- JPEG
- DICOM
- BMP
- PGM
- FITS
- STK

If path is not set the operation opens a file dialog and loads the image selected by the user. Otherwise the image in path is loaded. With help of the options part of the filename can be replaced. That allows to automatically open images belonging together in an application. If path is null the operation opens a file dialog and loads the image selected by the

* user. Otherwise the image in path is loaded. With help of the options part of the filename
* can be replaced. That allows to automatically open images belonging together in an application.

Options



replace string: The part of the input filename to be replaced.

replace with: The string the part of the input filename is replaced with.

Parameter

The only parameter is the path of the image to be opened.

Results

result (ImagePlus): The loaded image.

absoluteFilename (String): The absolute filename of the loaded image.

Operation: open series as stack

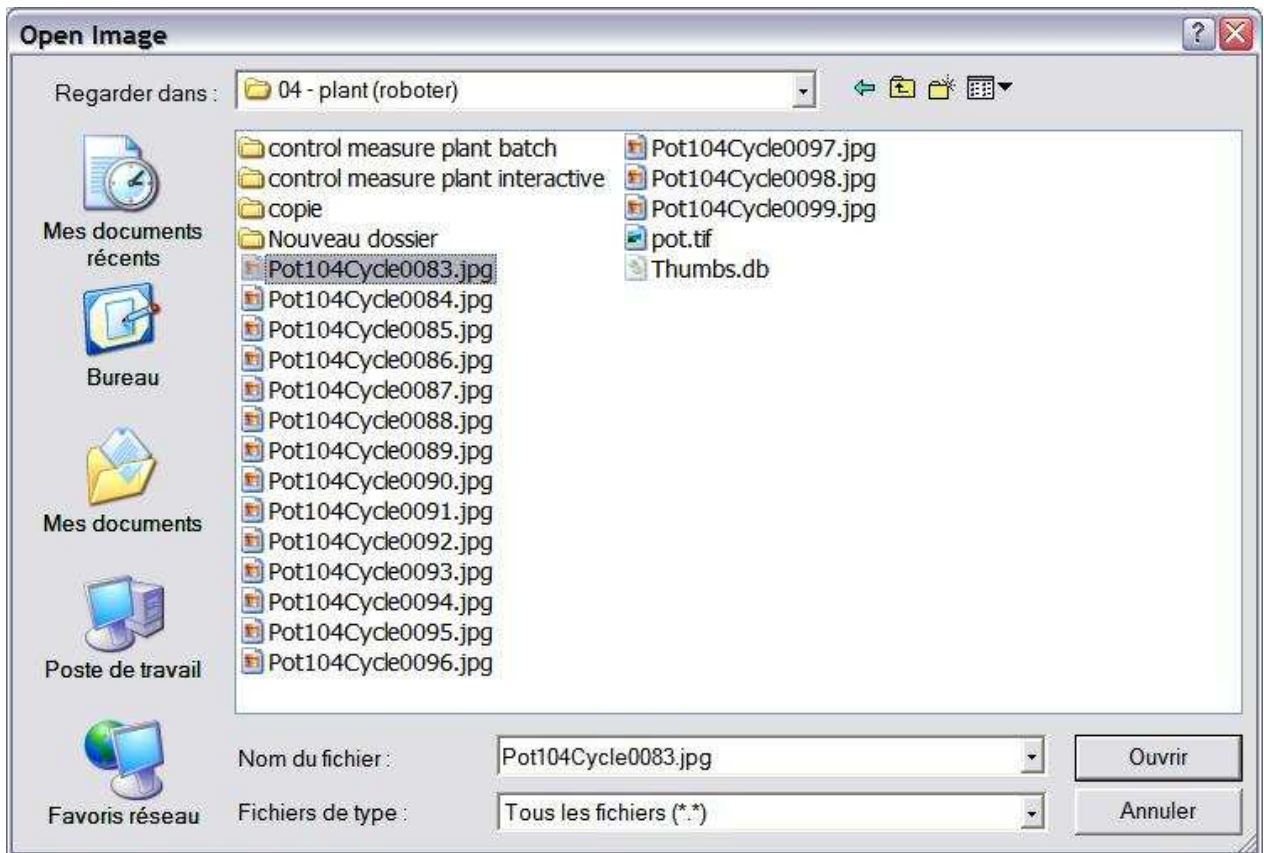


- * If path is null the operation opens a file dialog and loads the image selected by the
- * user. Otherwise the image in path is loaded. With help of the options part of the filename
- * can be replaced. That allows to automatically open images belonging together in an application.
- *
- * @author Volker Baecker
- */

Author:

A modification of the FolderOpener code from ImageJ. Modified by Volker Baecker.

Example

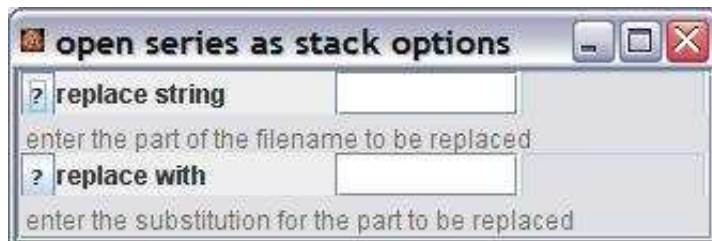




Description

The operation opens a series of images as a stack. If the path is not set the series is defined by the file selected by the user. All images with the same basename are loaded into the stack. The order is determined by the trailing numbers of the filenames. The numerical order is respected (e.g. 1<9<10<99<100, etc.).

Options



replace string: The part of the input filename to be replaced.

replace with: The string the part of the input filename is replaced with.

Parameter

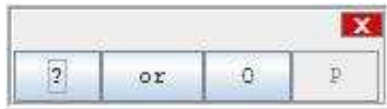
The only parameter is the path of the image to be opened.

Results

result (ImagePlus): The loaded image stack.

absoluteFilename (String): The absolute filename of the image that has been selected to define the series.

Operation: or



Author:

Uses the method or of the class ImageProcessor from ImageJ.

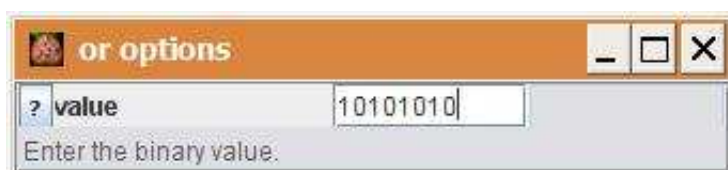
Example



Description

The operation replaces the value of each pixel in the image with the result of a binary or of the value and the input value.

Options



value: The value in binary form.

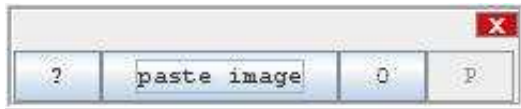
Parameter

The only parameter is the input image.

Results

The only result is the result image.

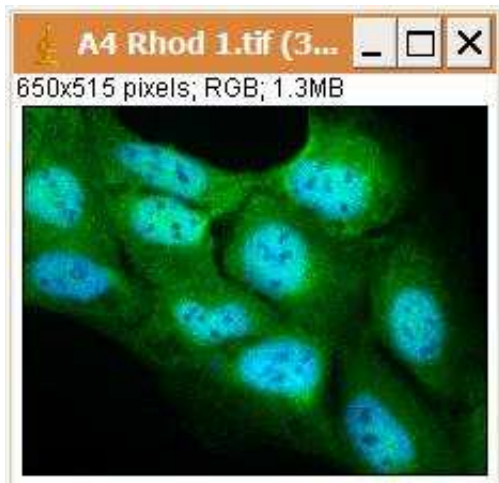
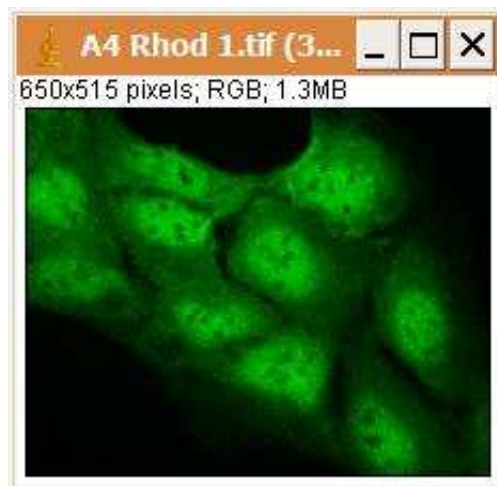
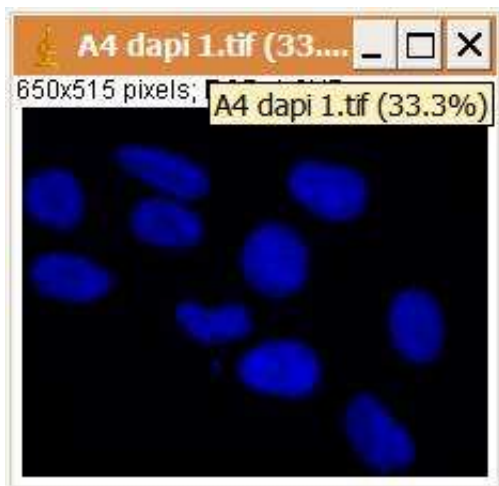
Operation: paste image



Author:

Uses the method copyBits of the class ImageProcessor from ImageJ.

Example



Description

The operation pastes one image into another. Different paste-modes can be used.

Options



mode: The paste mode determines the way the source image is pasted into the destination image.

Parameter

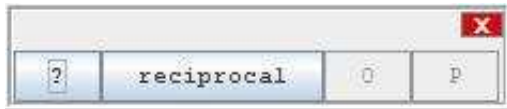
inputImage (ImagePlus): The image that will be pasted into the other image.

secondInputImage (ImagePlus): The image into which the first image will be pasted.

Results

ImagePlus: An image of the first input image pasted into the second one.

Operation: reciprocal



Author:

Volker Baecker.

Example



Description

The operation creates a float image in which each pixel has the reciprocal value $1/f$ of the pixel value f in the original image.

Options

The operation has no options.

Parameter

The only parameter is the input image.

Results

The only result is the result image. The result image is a 32-bit (float) image.

Operation: replace null with empty image

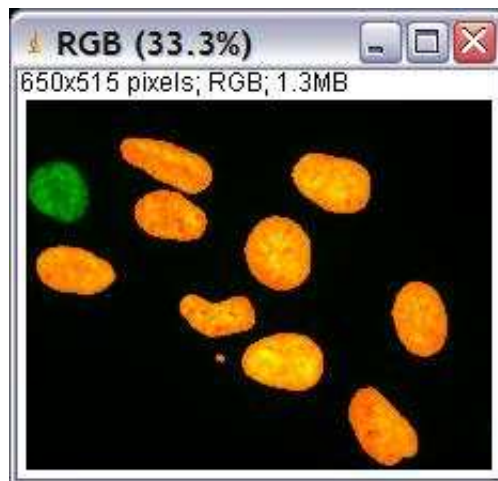
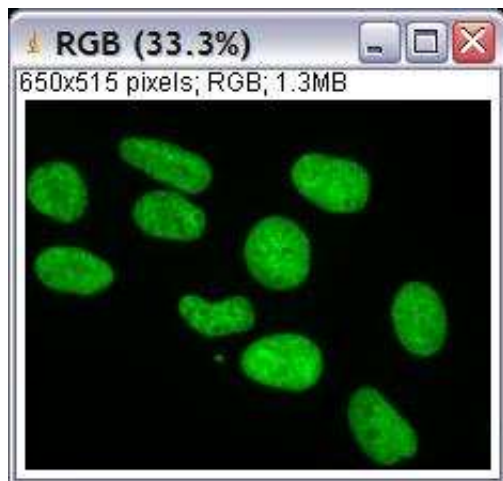


Author:

Volker Bäcker

Example





The first image

is the result when no objects are found.

Description

If the input image is null an empty image with the same size and type as the reference image is created, otherwise the result is a reference to the input image.

Options

The operation has no options.

Parameter

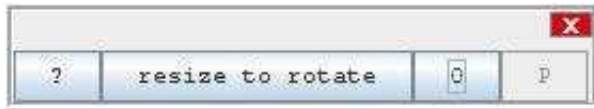
inputImage (ImagePlus): The input image. This image will be the result if it is not null.

referenceImage (ImagePlus): The reference image is used to determine the size and type of the new image that is created when the result image is null.

Results

The result is either a reference to the input image or a new image of the same size and type as the reference image.

Operation: resize to rotate



Author:

Volker Bäcker

Example



Description

Enlarges the image canvas in a way that the image can be rotated at all pixels will still be within the canvas.

Options



fill black: If checked the background will be black, otherwise it will be white.

Parameter

The only parameter is the input image.

Results

The only result is the result image.

Operation: save as tiff sequence



Author:

Uses the FileSaver from ImageJ.

Description

Save each slice of a stack as a single tif-image.

Options



output folder: The folder into which the image sequence will be written. This is either a fixed folder selected by the user or a subfolder with the given name of each source folder.

create in source folder: If selected the output folder will be created within the source folder, otherwise the value of output folder is interpreted as an absolute path.

name addition: A text that is appended to the filename.

add loop index to name: In each iteration the current index of the loop is added to the filename. This only works when the operation is used within a loop.

Parameter

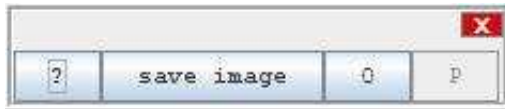
inputImage (ImagePlus): The image stack that will be saved as a series.

path (String): The path of the input image.

Results

success (Boolean): True if the image series has been saved without problems.

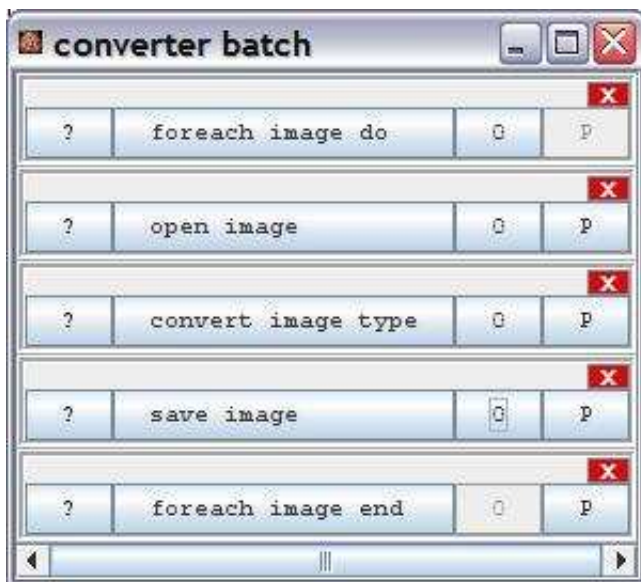
Operation: save image



Author:

Uses the FileSaver from ImageJ.

Example



Description

Save the input image in a configurable location.

Options



output folder: The folder into which the image sequence will be written. This is either a fixed folder

selected by the user or a subfolder with the given name of each source folder.

create in source folder: If selected the output folder will be created within the source folder, otherwise the value of output folder is interpreted as an absolute path.

name addition: A text that is appended to the filename.

add loop index to name: In each iteration the current index of the loop is added to the filename. This only works when the operation is used within a loop.

Parameter

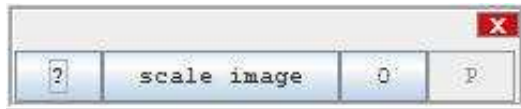
inputImage (ImagePlus): The image that will be saved as a series.

path (String): The path of the input image.

Results

success (Boolean): True if the image series has been saved without problems.

Operation: scale image



Author:

Runs the ImageJ scale command.

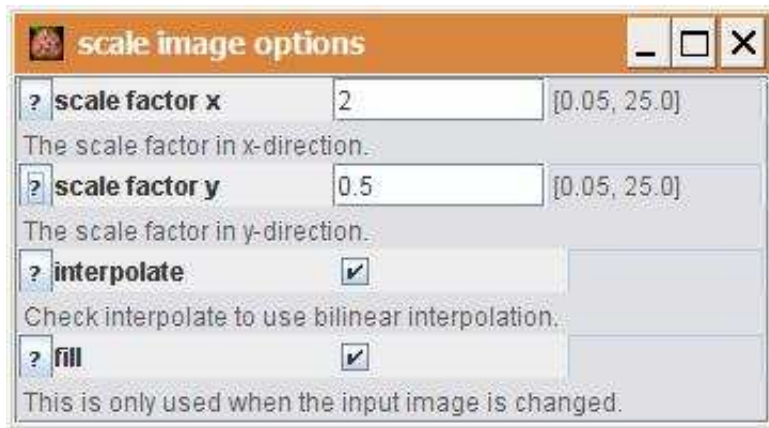
Example



Description

The operation scales an image in x and y-direction by a given factor.

Options



scale factor x: The scale factor for the x-direction.

scale factor y: The scale factor for the y-direction.

interpolate: If checked interpolation is used.

fill: If fill is checked and "keep source" is not checked in the operation, the empty space after scaling down the image will be filled with the current background color.

Parameter

The only parameter is the input image.

Results

The only result is the result image.

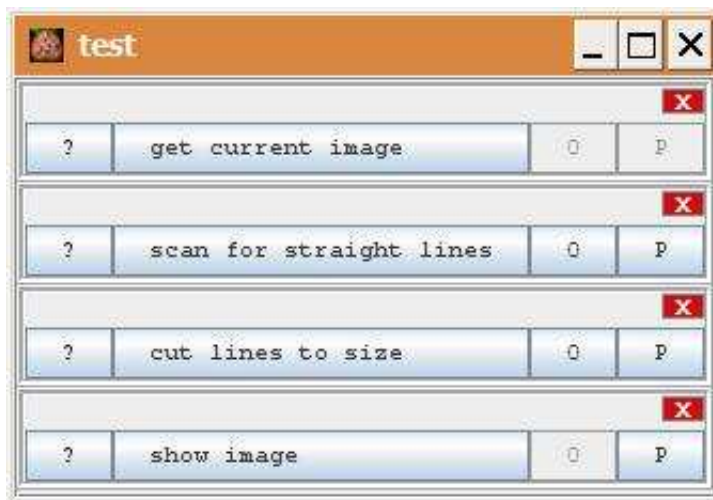
Operation: scan for straight lines

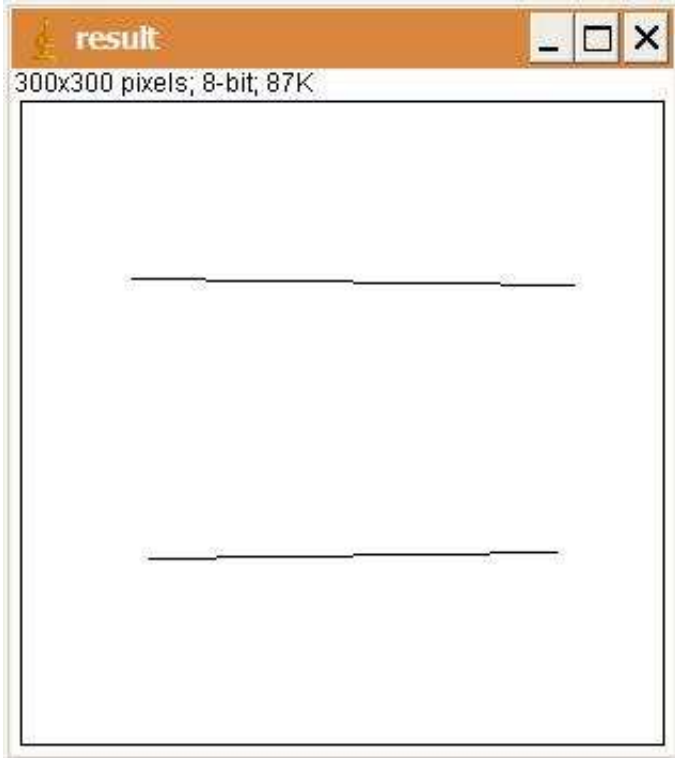
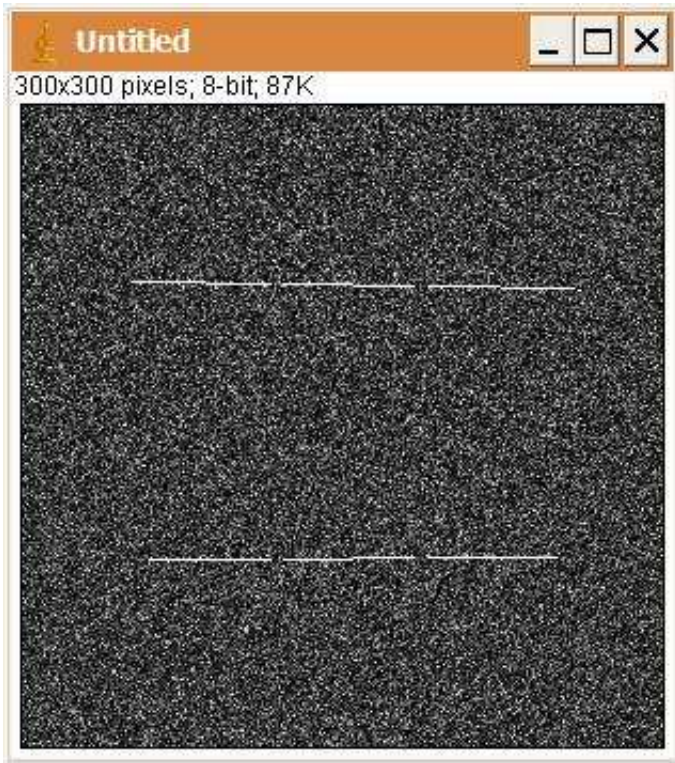


Author:

Volker Bäcker

Example





Description

The operation sweeps the image with lines of inclinations between -angle to angle and counts the pixel above threshold. If the count is above count threshold a line is detected.

Options



threshold: Only pixels with an intensity above threshold are counted in the line-detection process.

max angle: Lines with an angle between -max angle and max angle are detected.

count threshold: A line is detected if a part of at least count threshold pixels is above threshold.

Parameter

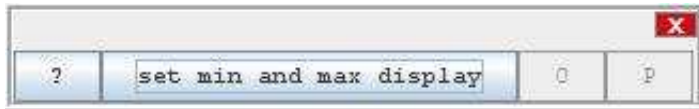
The only parameter is the input image.

Results

result (ImagePlus): An image of the detected lines.

resultLines (ArrayList<Line2D>): A list of the detected lines.

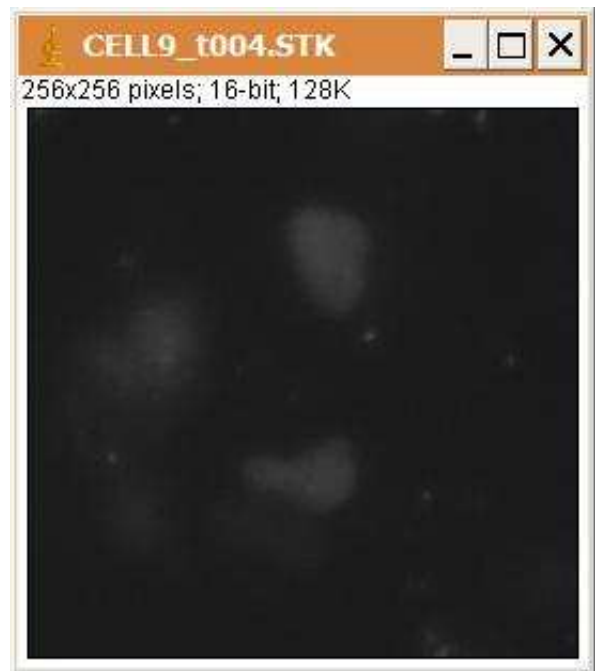
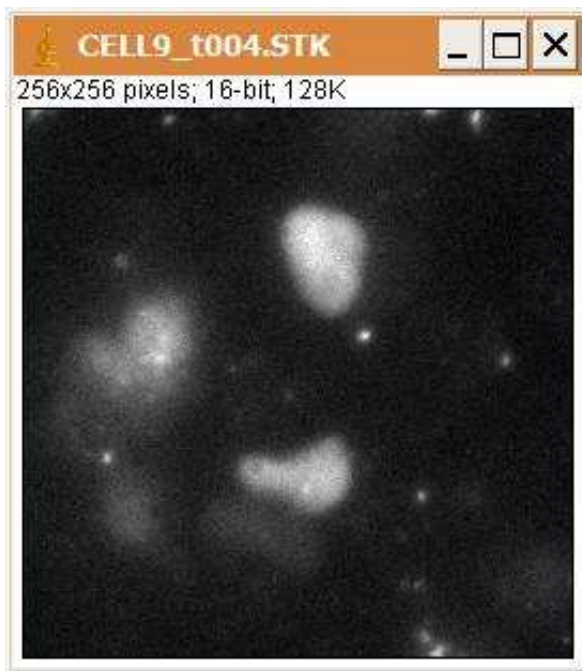
Operation: set min and max display



Author:

Volker Bäcker

Example



Description

Sets the min and max display values to 0 and 255 for 8-bit and color images and to 0 and 65535 for 16 and 32 bit images.

This allows to visually compare intensities between multiple images.

Options

The operation has no options.

Parameter

The only parameter is the input image.

Results

The result is the input image. The min and max display parameters are changed.

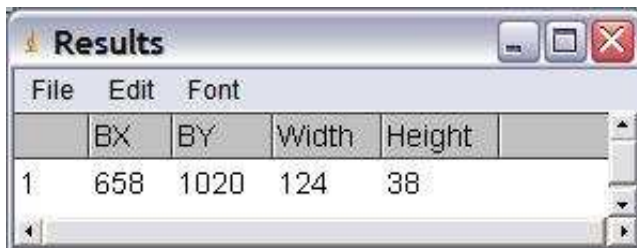
Operation: set scale from measurement



Author:

Volker Bäcker

Example



A screenshot of a "Results" window. It features a menu bar with "File", "Edit", and "Font". Below the menu bar is a table with the following data:

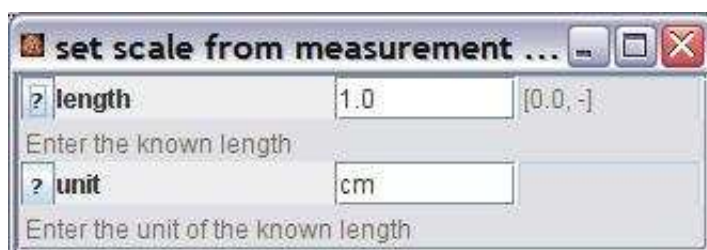
	BX	BY	Width	Height
1	658	1020	124	38



Description

The operation sets the spacial calibration of the image. The known distance and the unit are options of the operation. The measured distance is taken from the results table that is passed in as a parameter. It must contain the distance in the first row of the column width.

Options



length: The known length. To calibrate the spacial dimensions of the image it will be assumed that this is the width of the measured object.

unit: The unit of the length. This is an information for the user. It is not used internally.

Parameter

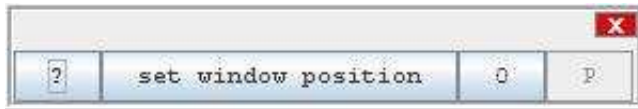
inputImage (ImagePlus): Any image type is possible.

measuredLength (ResultsTable): A results table containing the measured length of the reference object in the first row of the column width.

Results

The only result is the result image with its spacial calibration.

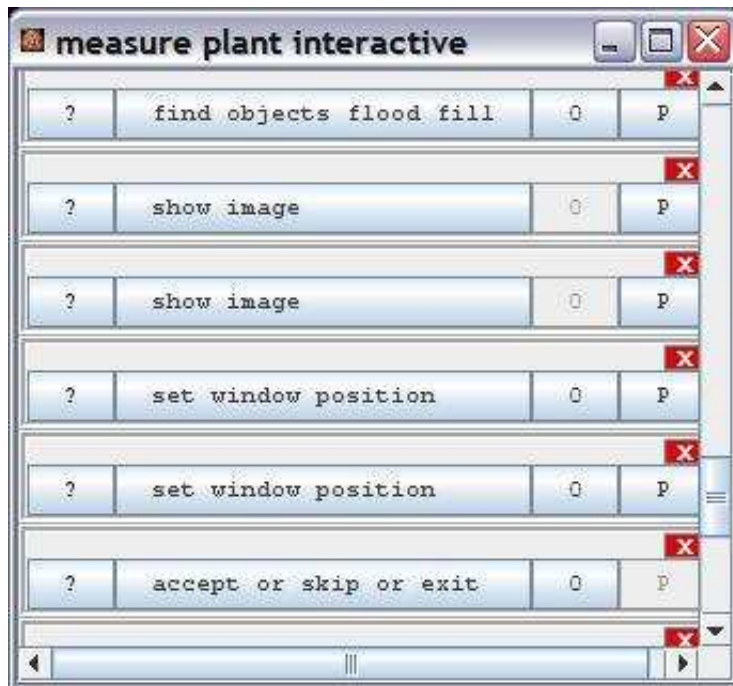
Operation: set window position



Author:

Volker Bäcker

Example



Description

The operation moves the upper left corner of the window of the input image to the position specified in the options of the operation.

Options



x: The x-coordinate of the new position on the screen.

y: The y-coordinate of the new position on the screen.

Parameter

The only parameter is the input image. The input image must have a window.

Results

The operation has no results.

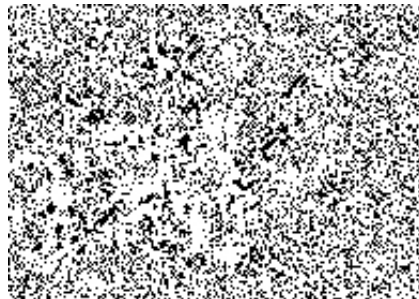
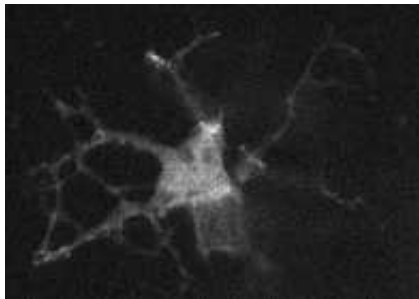
Operation: get vector image from hessian



Author:

Volker Bäcker.

Example



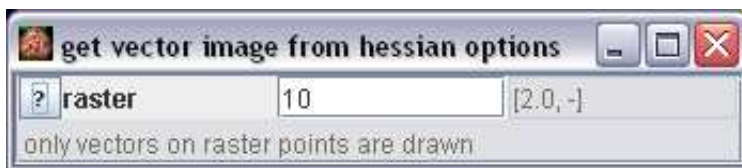
hessian displayed with raster 2

the eigenvectors of the

Description

Draws the eigenvectors from the hessian operations.

Options



raster: Only points on the raster are drawn with the maximum length of raster / 2.

Parameter

The only parameter is the hessian image.

Results

The only result is the result image.

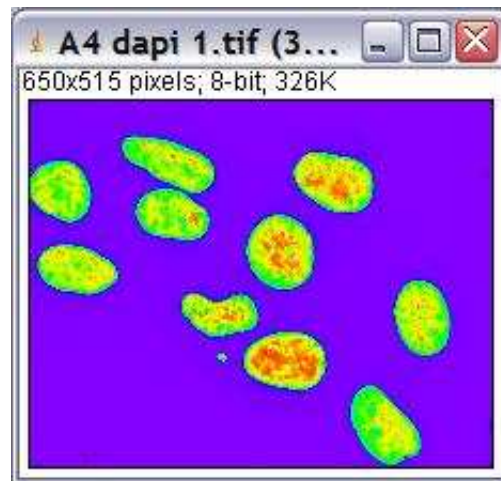
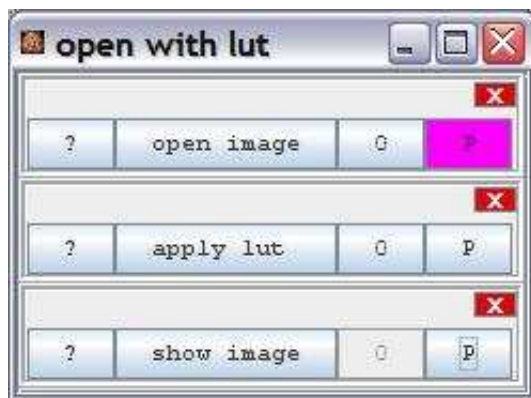
Operation: show image



Author:

Volker Bäcker

Example



Description

The operation opens a window and display the input image.

Options

The operation has no options.

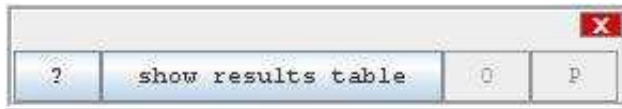
Parameter

The only parameter is the input image.

Results

The operation has no results.

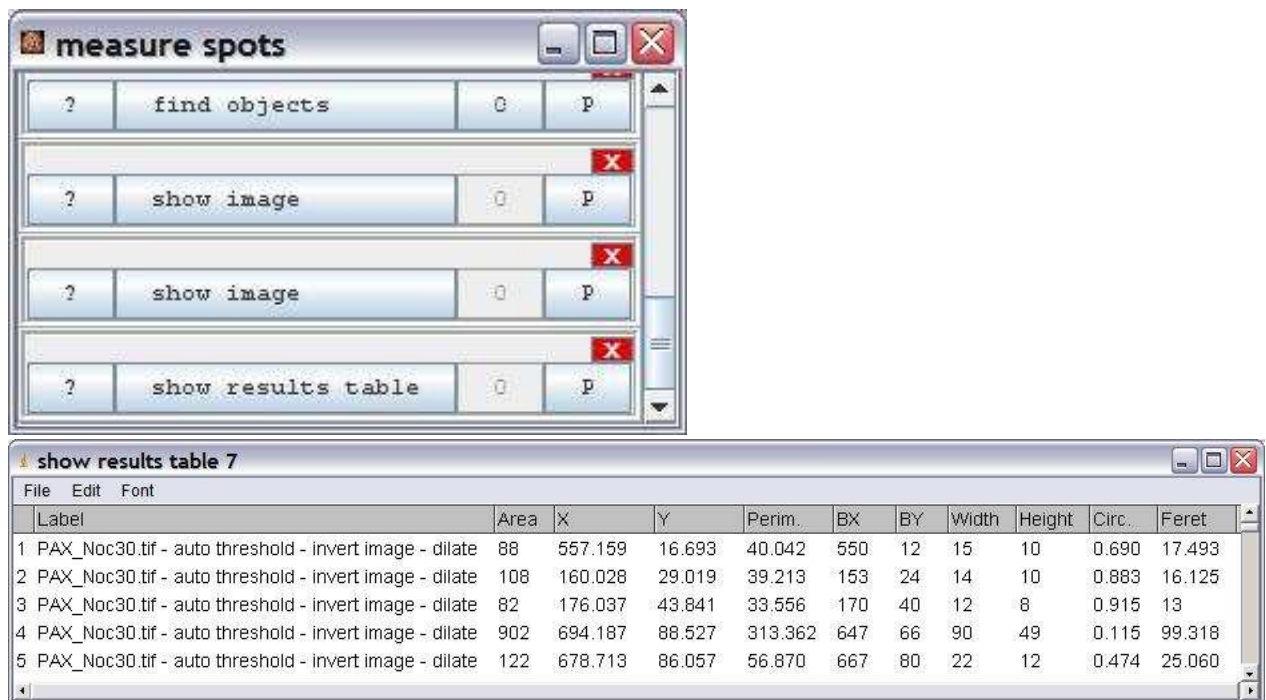
Operation: show results table



Author:

Volker Bäcker

Example



Description

The operation opens a window and displays the input results table.

Options

The operation has no options.

Parameter

The only parameter is the input results table.

Results

The operation has no results.

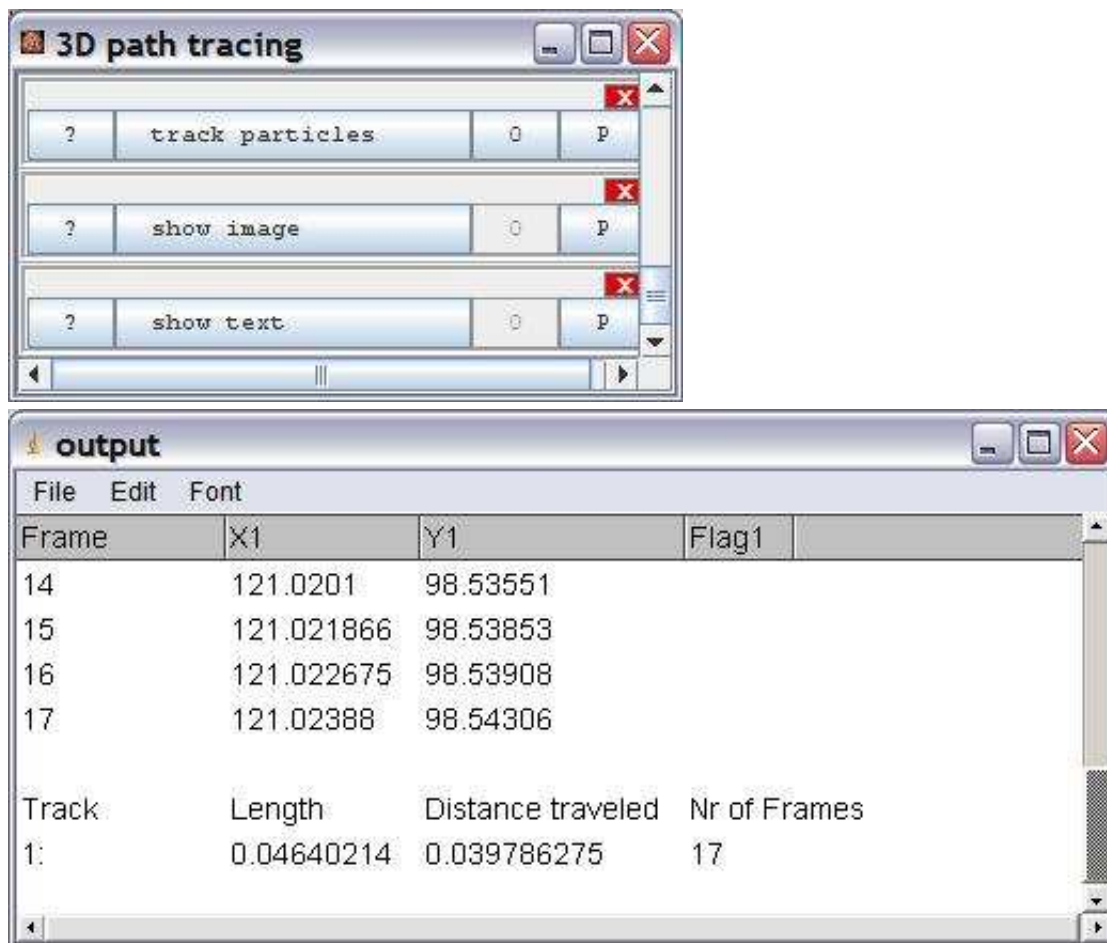
Operation: show text



Author:

Volker Bäcker

Example



The screenshot shows two windows from a 3D path tracing application. The top window, titled "3D path tracing", contains a list of operations:

- track particles
- show image
- show text

The bottom window, titled "output", displays a table of frame data:

Frame	X1	Y1	Flag1
14	121.0201	98.53551	
15	121.021866	98.53853	
16	121.022675	98.53908	
17	121.02388	98.54306	

Track	Length	Distance traveled	Nr of Frames
1:	0.04640214	0.039786275	17

Description

Opens a window and displays the input text panel.

Options

The operation has no options.

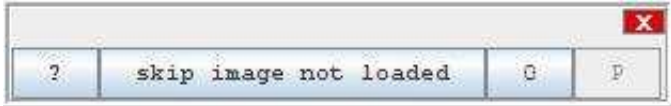
Parameter

inputTextPanel (TextPanel): The input text panel.

Results

The operation has no results.

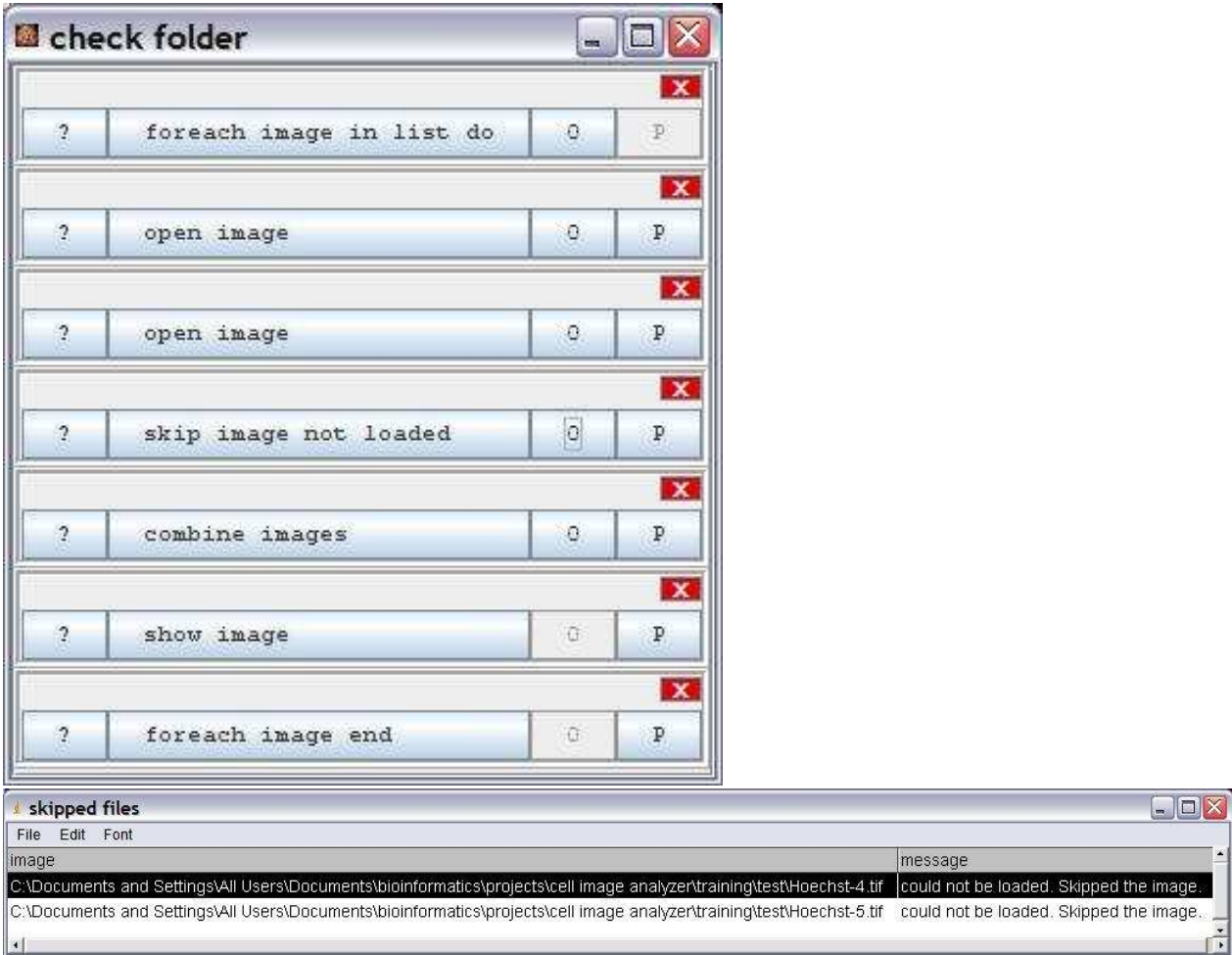
Operation: skip image not loaded



Author:

Volker Bäcker

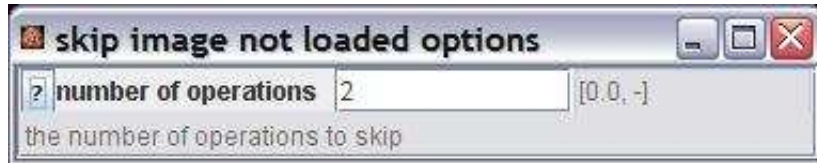
Example



Description

The operation allows to skip a configurable number of operations if the input image is null.

Options



number of operations: The number of operations that will be skipped if the input image is null.

Parameter

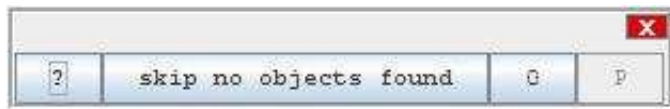
inputImage (ImagePlus): The input image. It might be null.

filename (String): The filename to be reported when an image is skipped.

Results

The operation has no results.

Operation: skip no objects found



Author:

Volker Bäcker

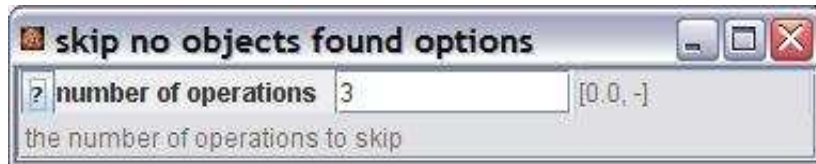
Example



Description

If the results table doesn't contain at least one object a configurable number of operations is skipped.

Options



number of operations: The number of operations that will be skipped if the input image is null.

Parameter

objects (ResultsTable): The results table to be checked.

Results

The operation has no results.

Operation: skip saturated



Author:

Volker Bäcker

Example

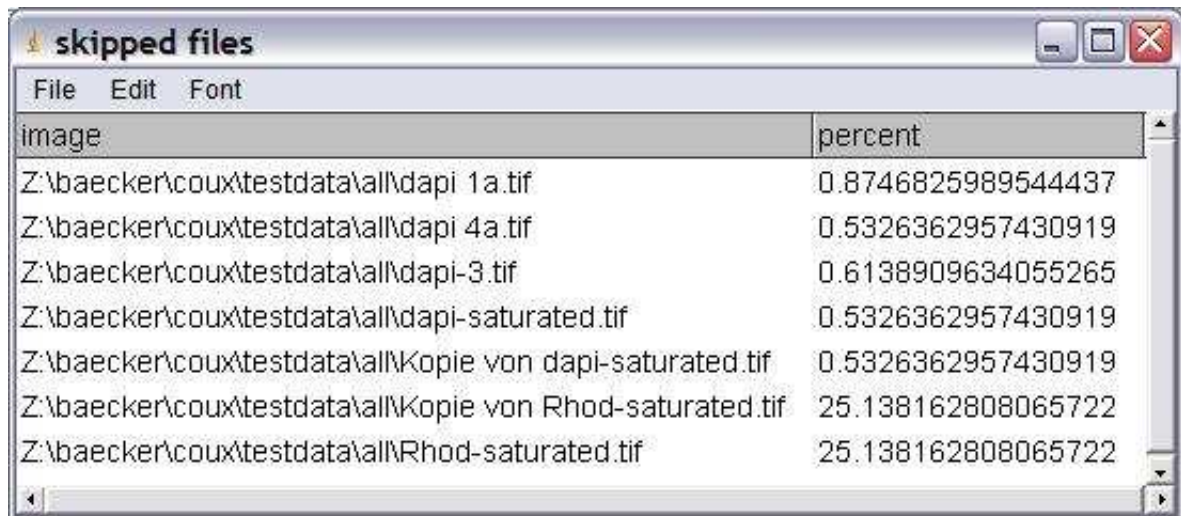
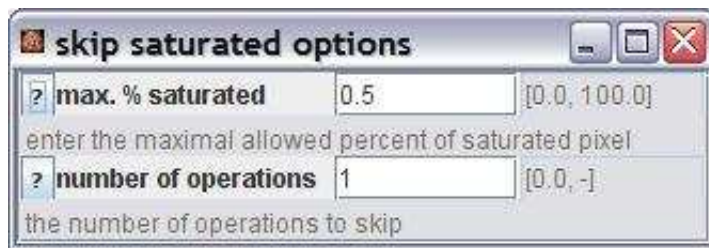


image	percent
Z:\baecker\coux\testdata\all\dapi 1a.tif	0.8746825989544437
Z:\baecker\coux\testdata\all\dapi 4a.tif	0.5326362957430919
Z:\baecker\coux\testdata\all\dapi-3.tif	0.6138909634055265
Z:\baecker\coux\testdata\all\dapi-saturated.tif	0.5326362957430919
Z:\baecker\coux\testdata\all\Kopie von dapi-saturated.tif	0.5326362957430919
Z:\baecker\coux\testdata\all\Kopie von Rhod-saturated.tif	25.138162808065722
Z:\baecker\coux\testdata\all\Rhod-saturated.tif	25.138162808065722

Description

If more than a configurable percentage of pixels in the input image is saturated (has the maximal intensity), a configurable number of operations is skipped.

Options



max % saturated: The maximum allowed percentage of saturated pixels. If more pixels are saturated a number of operations will be skipped.

number of operations: The number of operations that will be skipped.

Parameter

inputImage (ImagePlus): The image for which the percentage of saturated pixels will be checked.

filename (String): The filename to be reported for the input image.

Results

result (ImagePlus): A reference to the input image.

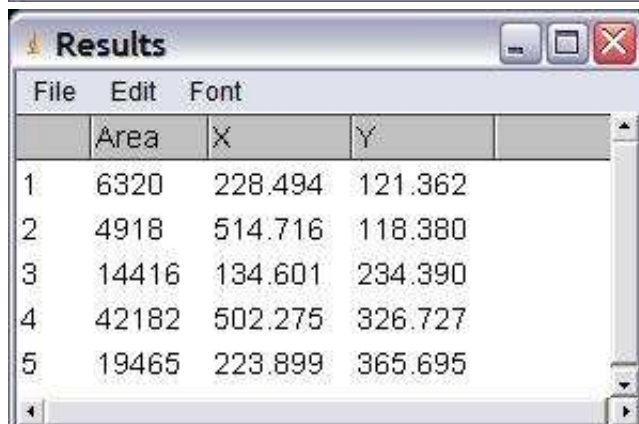
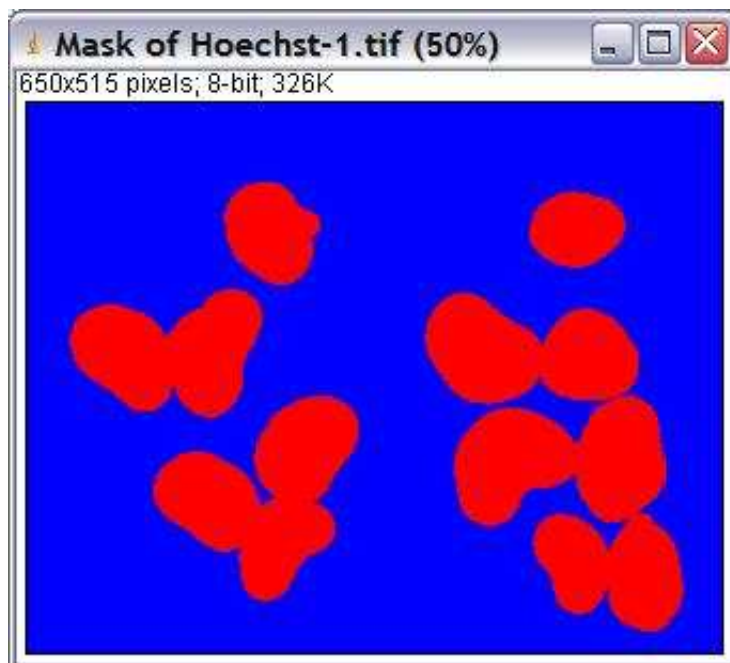
Operation: split and count



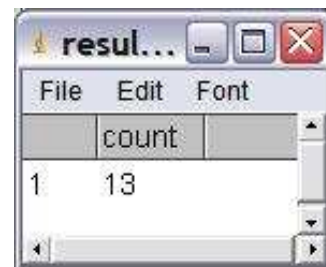
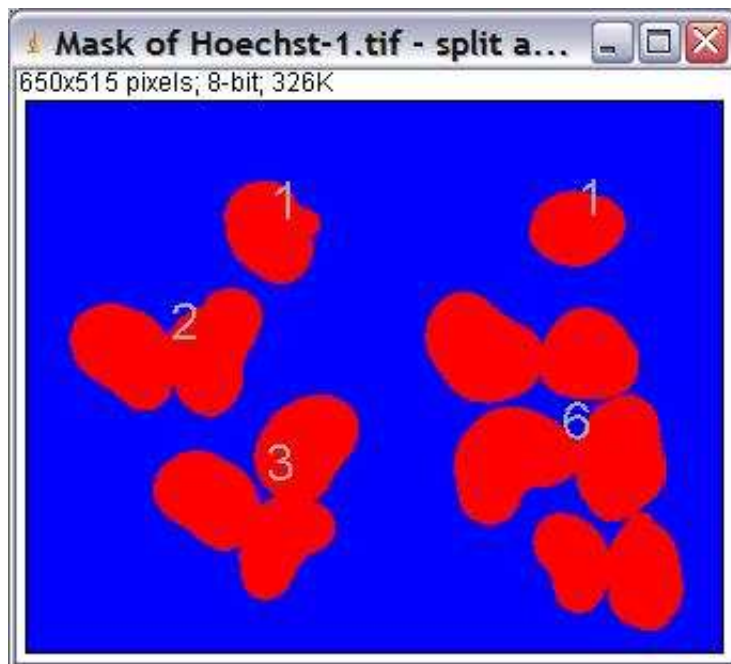
Author:

Volker Bäcker

Examples



	Area	X	Y
1	6320	228.494	121.362
2	4918	514.716	118.380
3	14416	134.601	234.390
4	42182	502.275	326.727
5	19465	223.899	365.695



Description

The operation takes a mask and a results table containing the areas and centroids of the objects in the mask. Corrects the count of the objects by using the average size of this kinds of objects, that must be provided as an option.

Options



average size: The average size of the kinds of objects to be counted.

Parameter

inputImage (ImagePlus): Any kind of image. A copy of the image is used to indicate how many objects have been counted for each input object.

measurements (ResultsTable): A results table with the centroids and areas of the objects in the image.

Results

result (ImagePlus): An image indicating how many objects have been counted for each input object.

count (ResultsTable): The results table with the corrected count.

Operation: split channels



Author:

The operation uses the class RGBStackSplitter from ImageJ.

Example



Description

The operation separates the three channels of an rgb image or stack.

Options

The operation has no options.

Parameter

The only parameter is the input image. This must be an rgb image or rgb stack.

Results

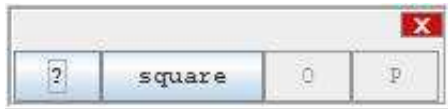
There are three results, one for each channel of the rgb image.

result red: the red component of the rgb image

result green: the green component of the rgb image

result blue: the blue component of the rgb image

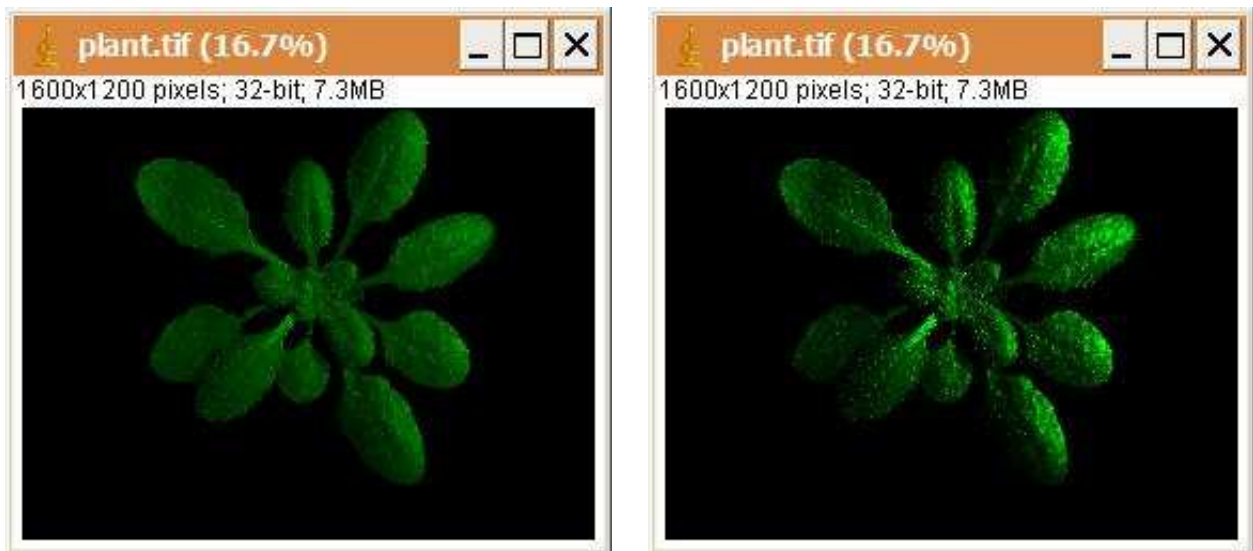
Operation: square



Author:

Uses the method `sqr` of the class `ImageProcessor` from `ImageJ`.

Example



Description

The operation replaces each intensity value by the square of the value.

Options

The operation has no options.

Parameter

The only parameter is the input image.

Results

The only result is the result image.

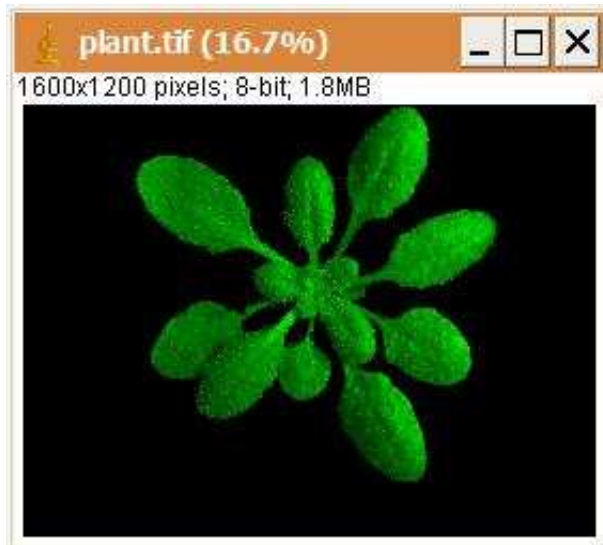
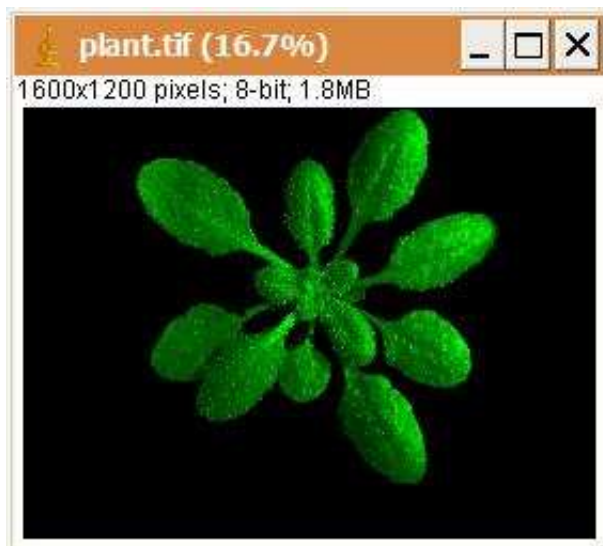
Operation: square root



Author:

Uses the method `sqrt` of the class `ImageProcessor` from `ImageJ`.

Example



Description

The operation replaces each intensity value in the image with the square root of the intensity value.

Options

The operation has no options.

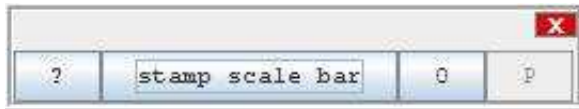
Parameter

The only parameter is the input image.

Results

The only result is the result image.

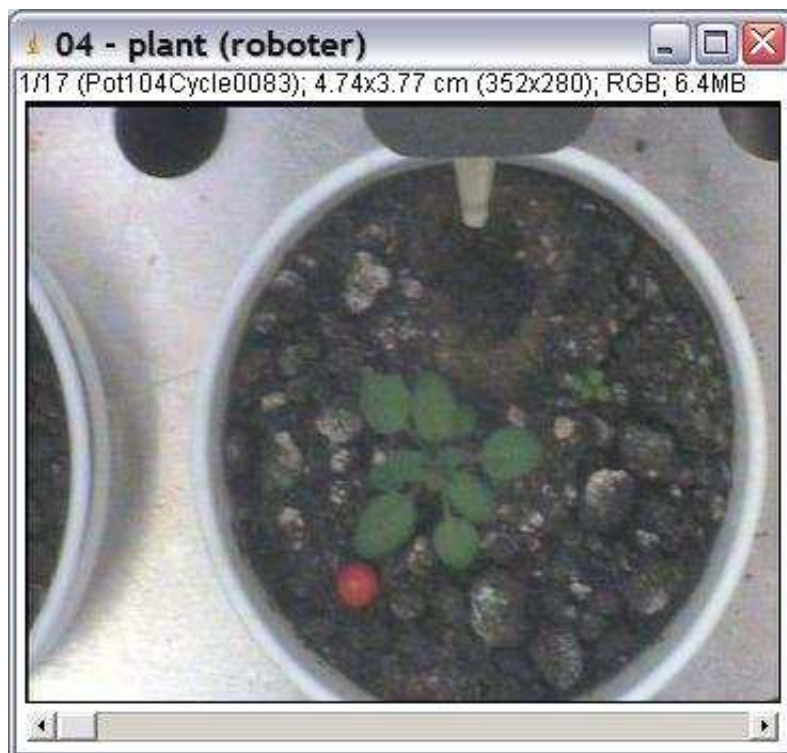
Operation: stamp scale bar

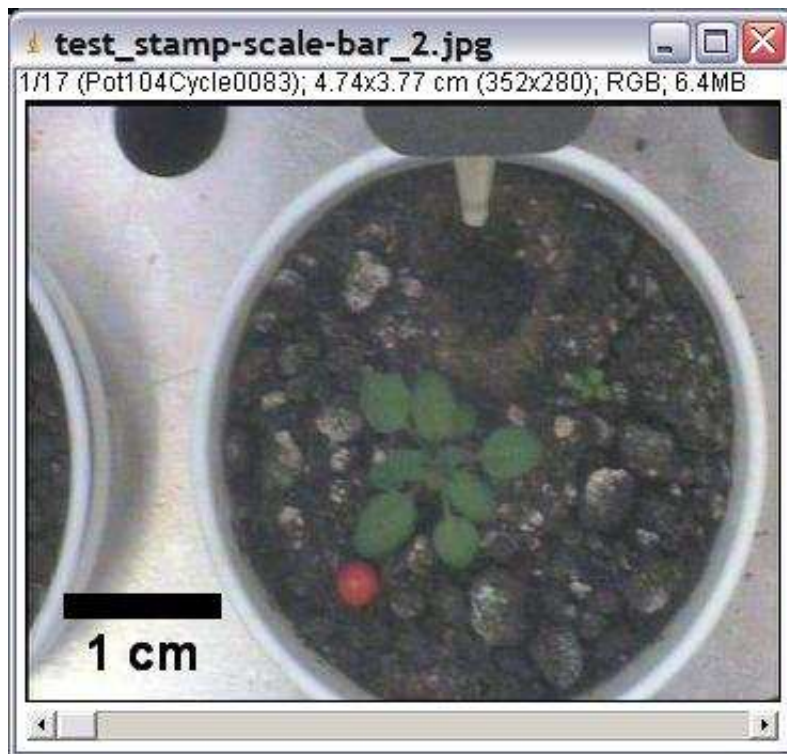


Author:

The operation calls the "Scale Bar" command from ImageJ.

Example

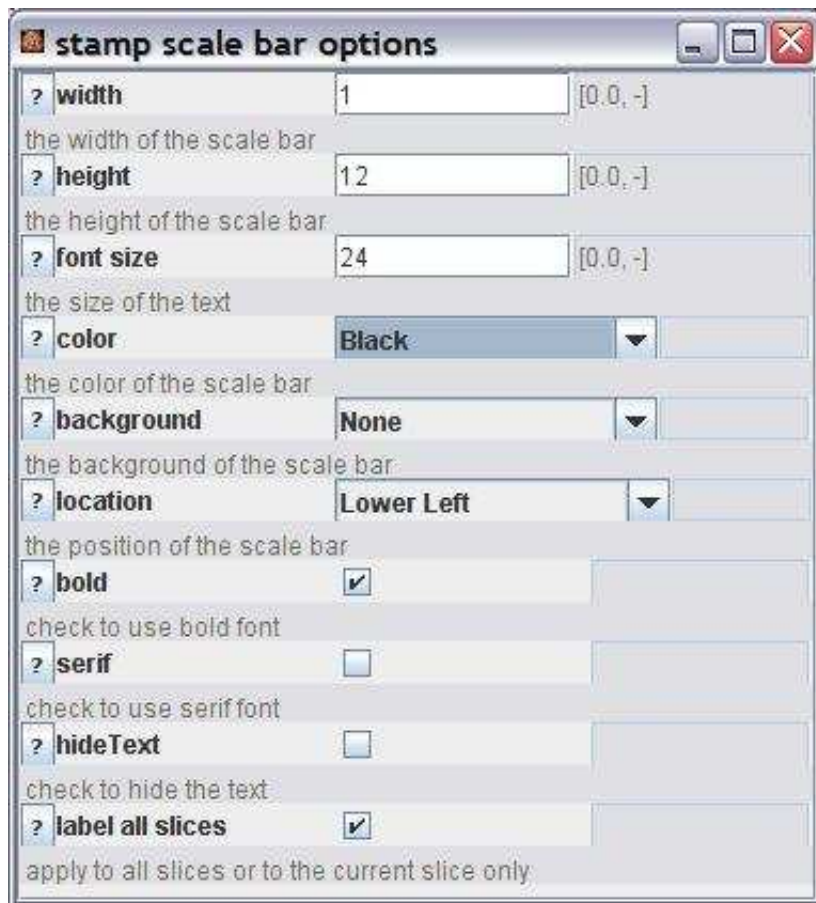




Description

Stamps a scale bar, indicating the spacial scale, into an image. If the image is a stack the scale bar can be stamped to the current slice or to all slices.

Options



width: The width of the scale bar in the unit of the scale (for example cm).

height: The height of the scale bar in pixel.

font size: The size of the font used for the text indicating the length and the unit.

color: The color of the scale bar.

background: The color of the background of the scale bar and the text. If none is selected the background is the image.

location: The location of the scale bar. This can be either one of the corners or at selection. In the later case the upper left corner of the scale bar will be in the upper left corner of the selection (or its bounding box).

bold: If checked the text will be bold.

serif: If checked the text will have serifs.

hide text: If checked the scale bar will be displayed without text.

label all slices: If checked the scale bar will be stamped into all sclices of the stack, otherwise it will only be stamped into the current slice.

Parameter

The input image is the only parameter.

Results

The result image with the scale bar stamped into it.

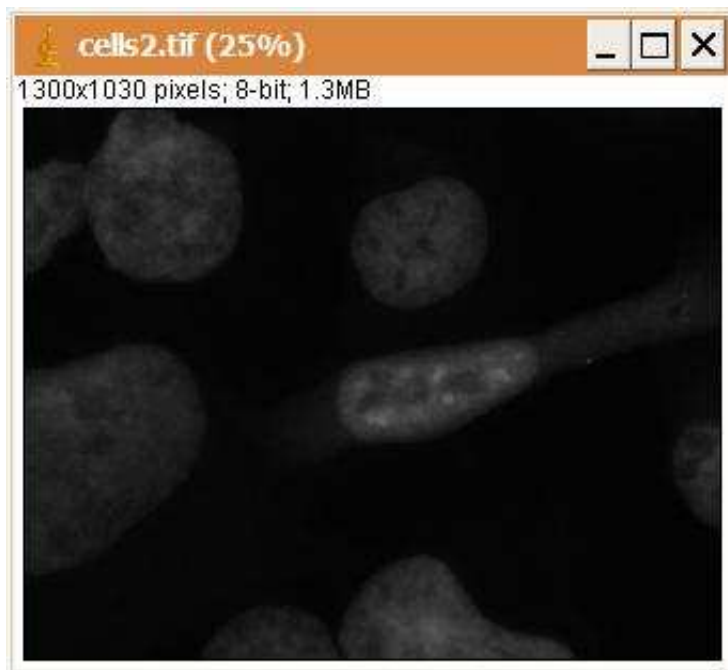
Operation: subtract

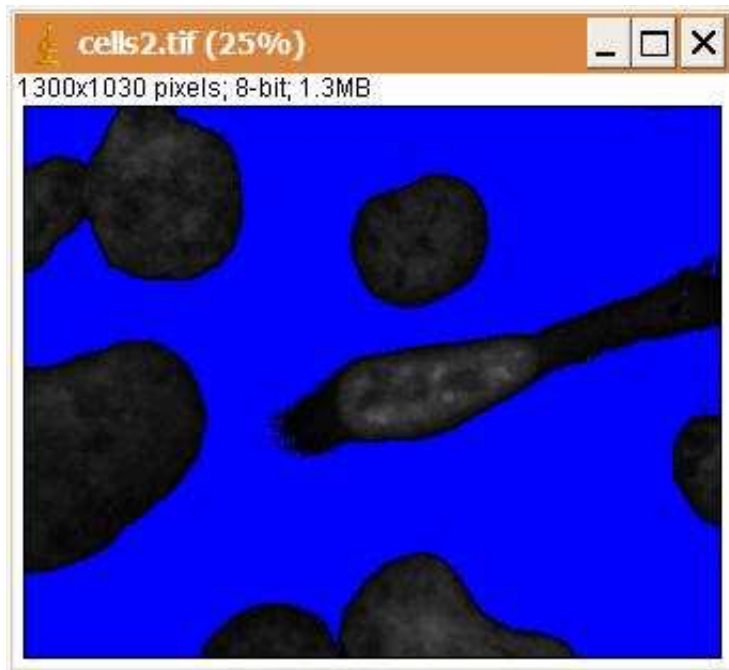


Author:

The operation uses the method add of the class ImageProcessor from ImageJ.

Example

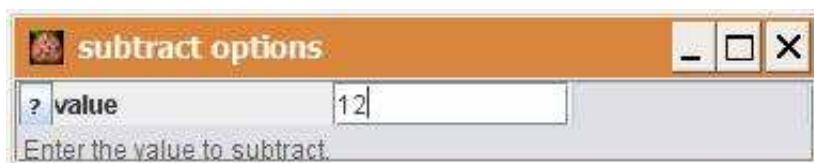




Description

Subtracts a constant value from the intensity value of each pixel in the image.

Options



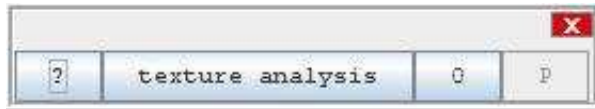
Parameter

The only parameter is the input image.

Results

The only result is the result image.

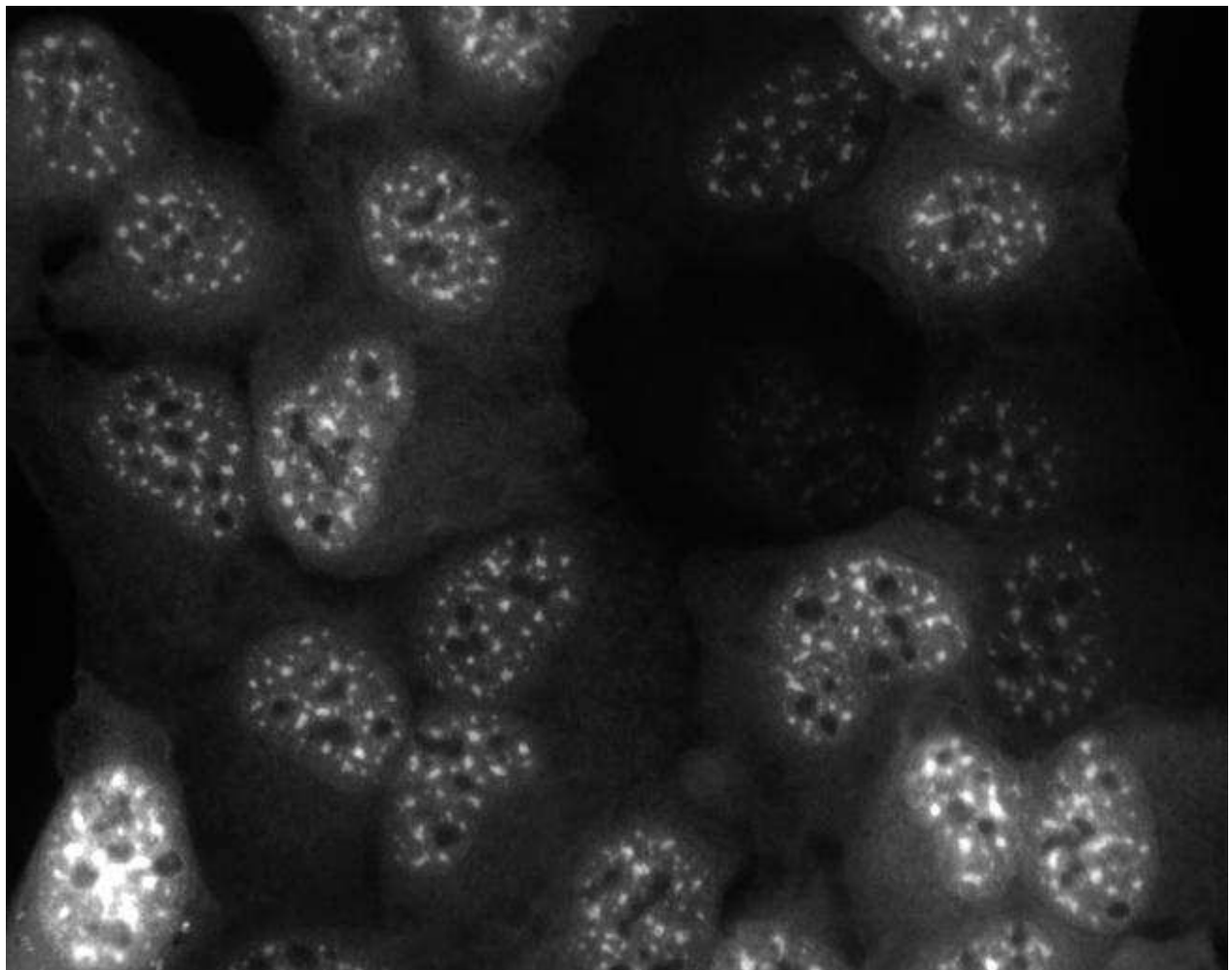
Operation: texture analysis

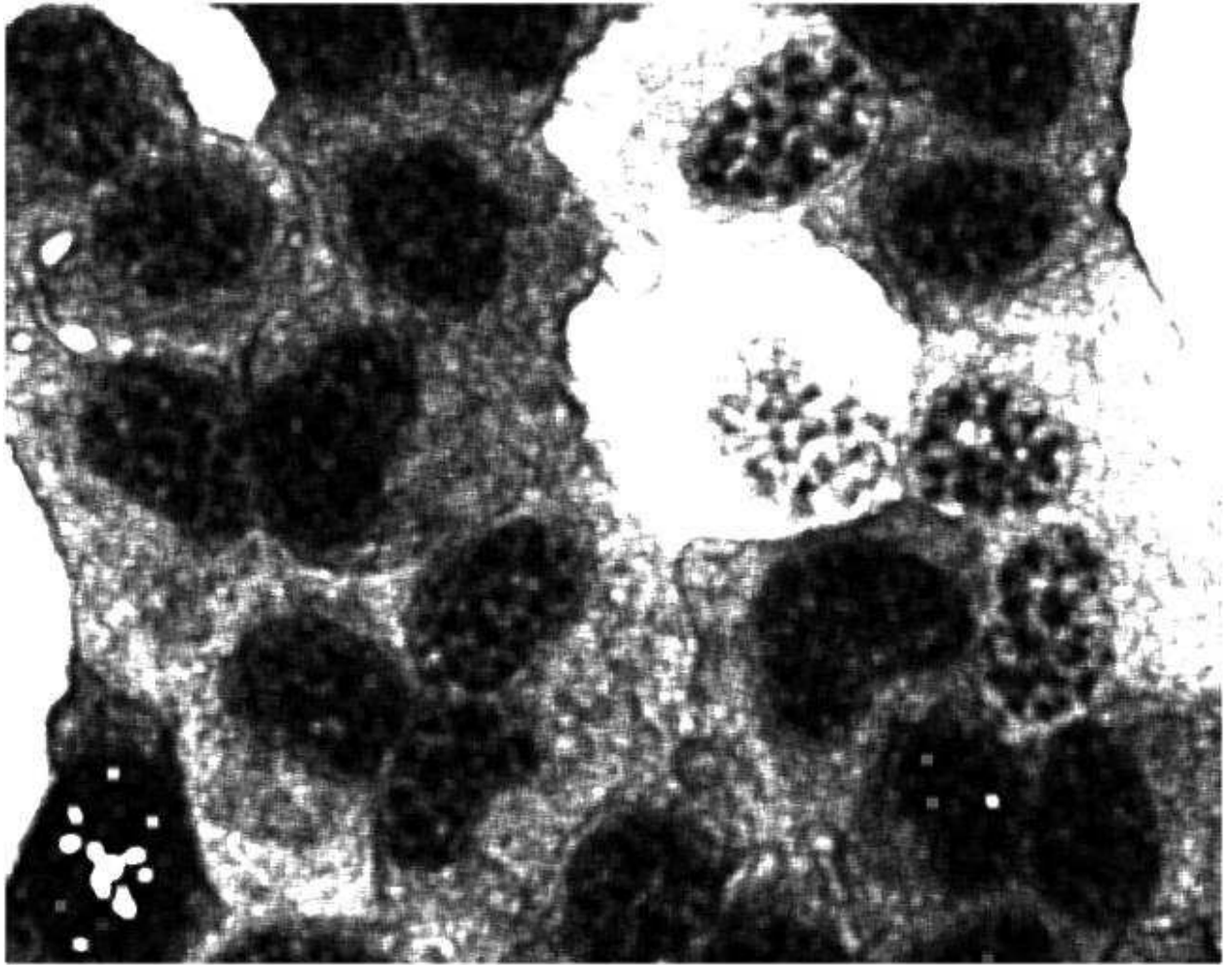


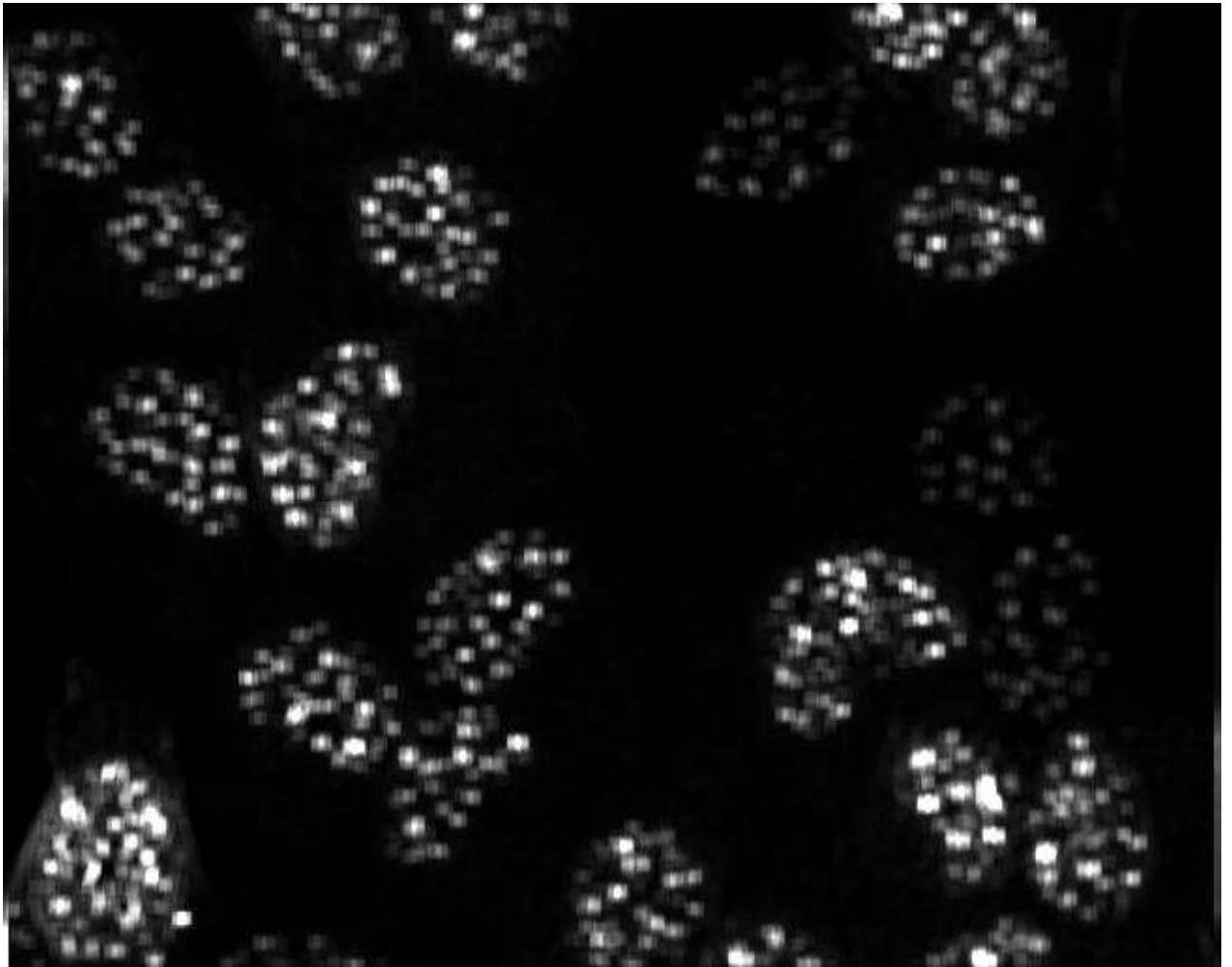
Author:

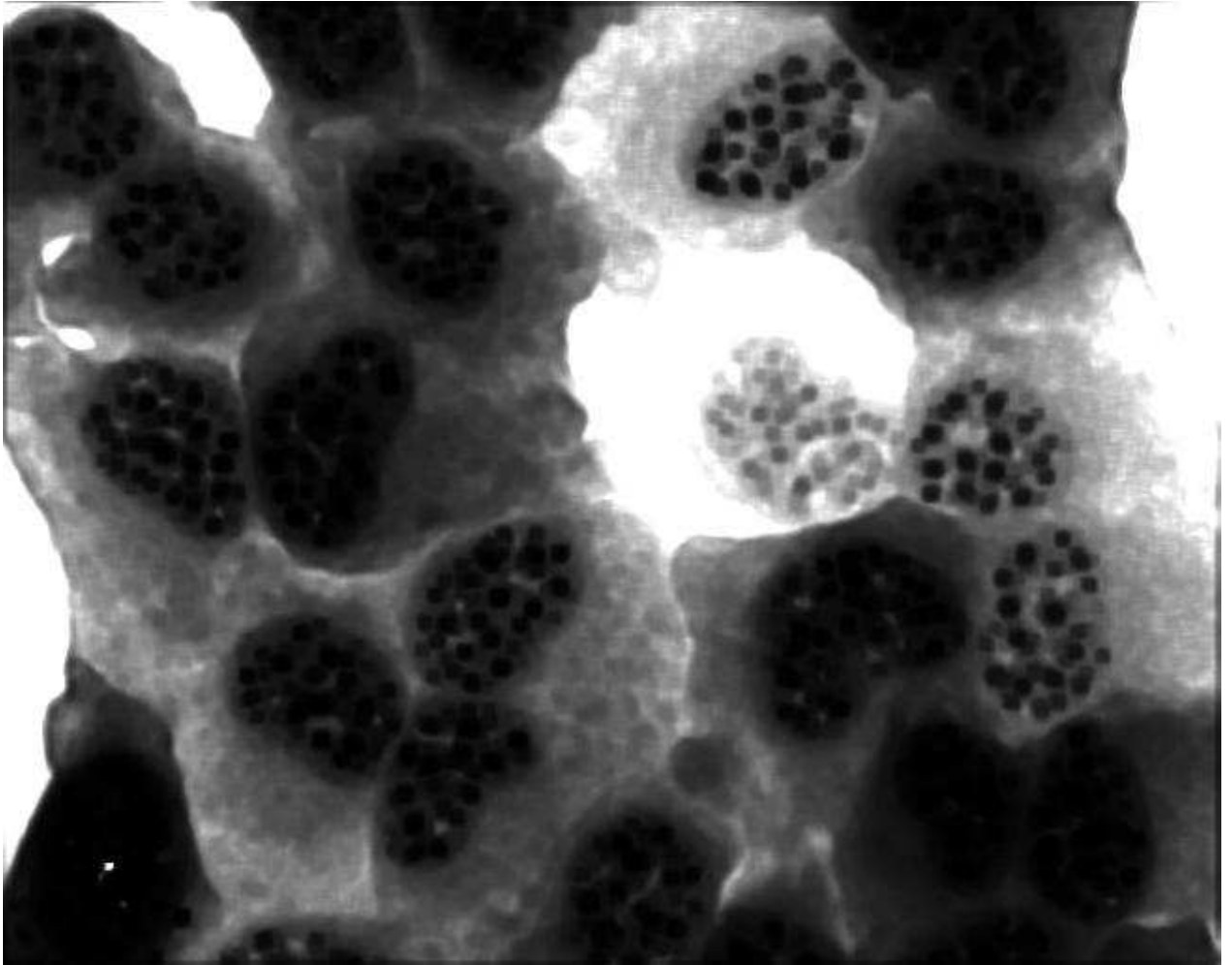
The operation uses code from the plugin texture analyzer by Julio E. Cabrera.

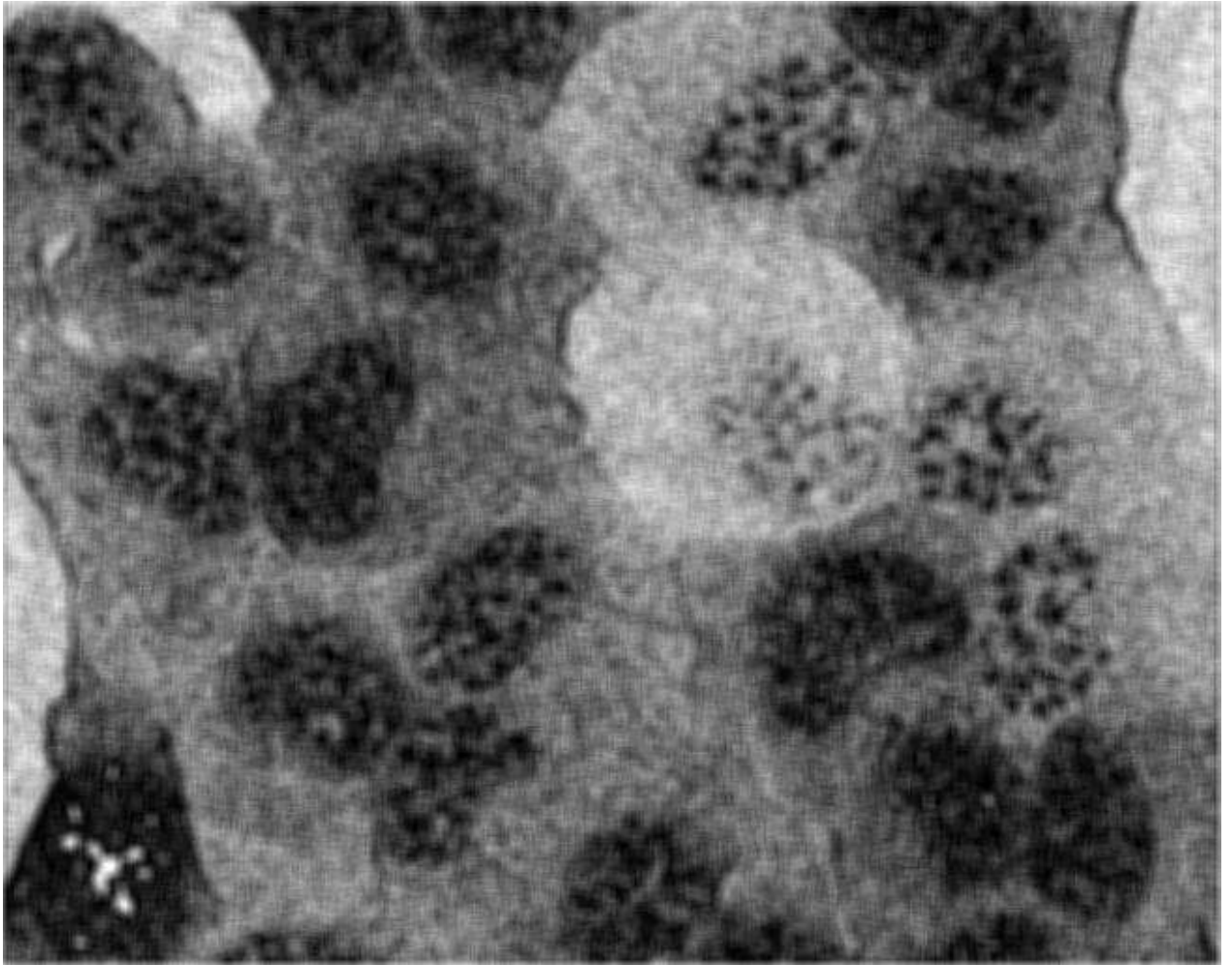
Example:

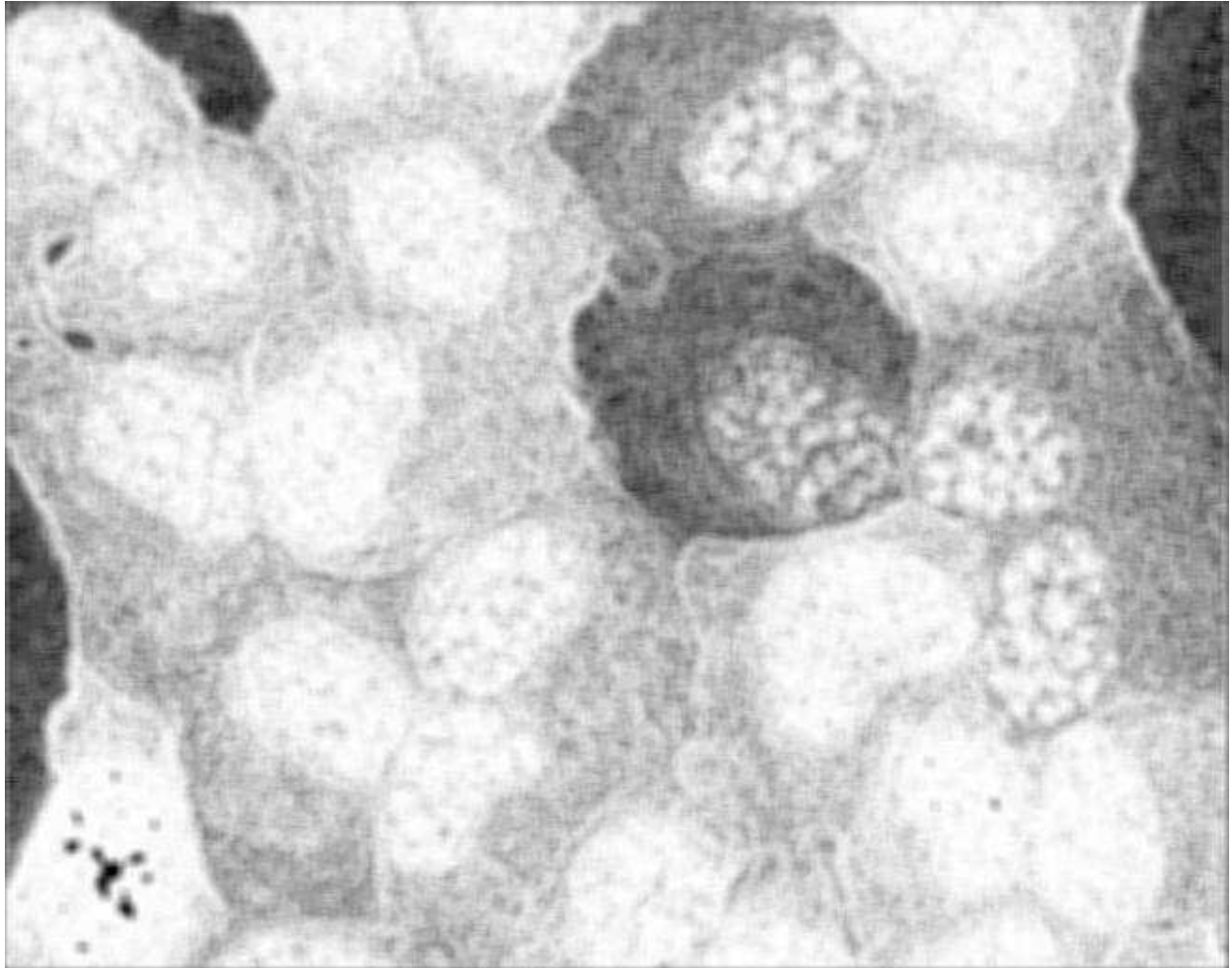












Description

Calculates a stack with one slice for each of the glcm texture analysis features.

Options

The screenshot shows a dialog box titled "texture analysis options" with three settings:

? radius	3	[1.0, -]
the radius of the neighborhood in which the texture is analyzed		
? step	1	[1.0, -]
the step size for the texture analysis		
? angle	0 degrees	▼
select the direction for the texture analysis		

radius: The radius of the neighborhood in which the texture features are calculated.

step: The step size for the correlation analysis.

angle: The direction into which the step for the correlation analysis is done.

Parameter

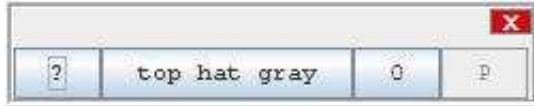
The only parameter is the input image. It must be a 8-bit greyscale image.

Results

The result is a 32-bit stack with 5 slices:

1. angular second moment
2. contrast
3. correlation
4. inverse difference moment
5. entropy

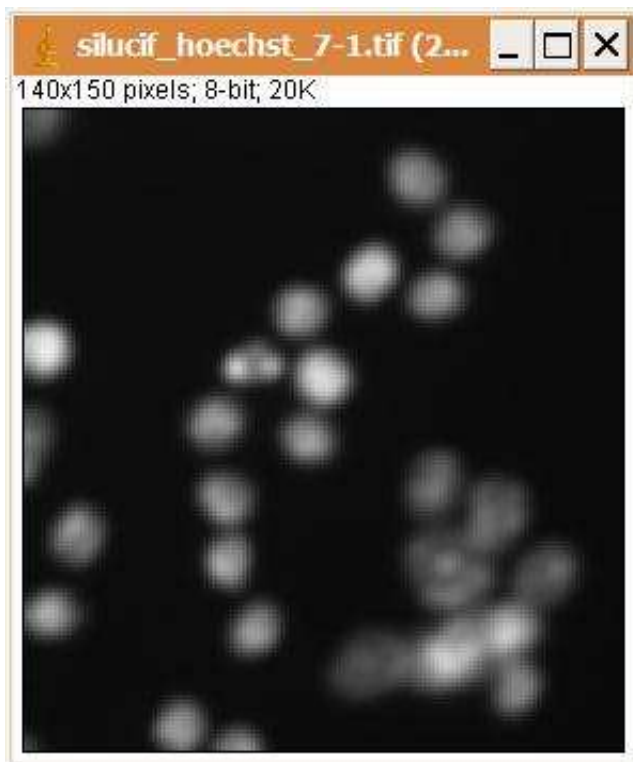
Operation: top hat gray

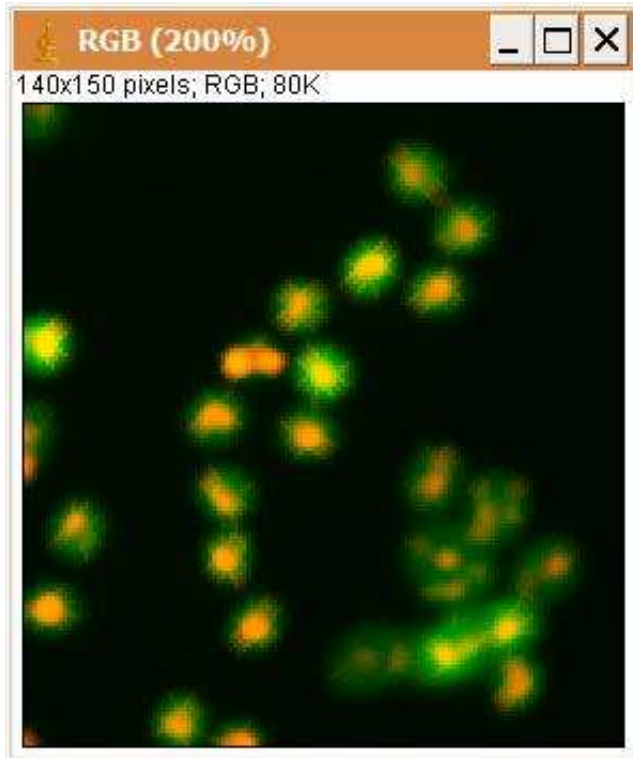
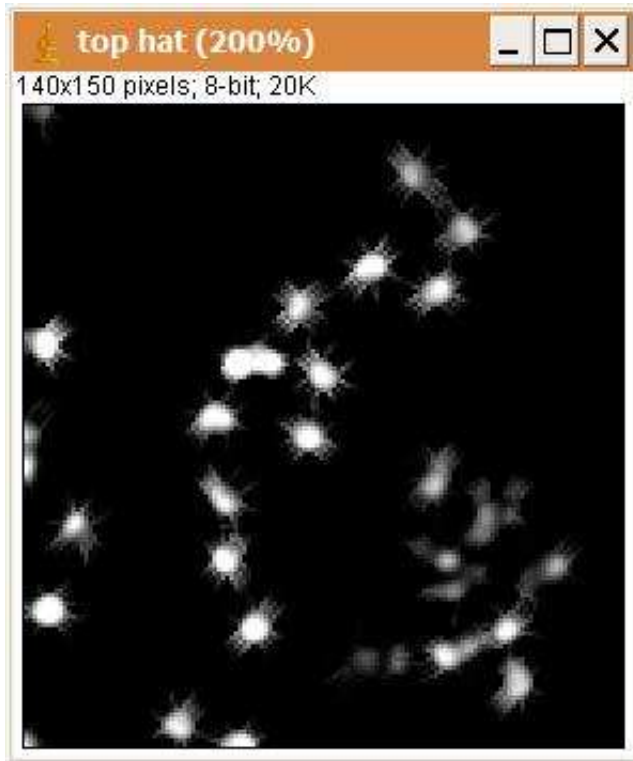


Author:

The operation calls the Grayscale Morphology plugin by Dimitar Prodanov.

Example





Description

The operation extracts bright regions from the image. It subtracts the result of the open operation from the input image.

Options

morphology type: Choose whether the 8-bit or the 32-bit morphology plugins are used.

radius: The radius of the structuring element.

structuring element type: Select the form of the structuring element. If free form is selected the structuring element can be entered manually.

structuring element: Enter the structuring element directly. This is used when free form is selected in the field structuring element type.

Parameter

The only parameter is the input image.

Results

The only result is the result image. The result image is a 32-bit or an 8-bit image, depending on the morphology type used.

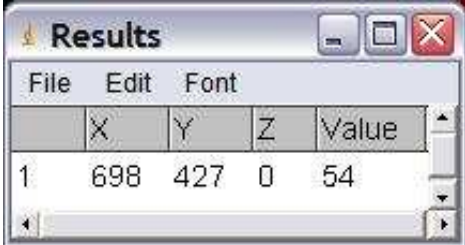
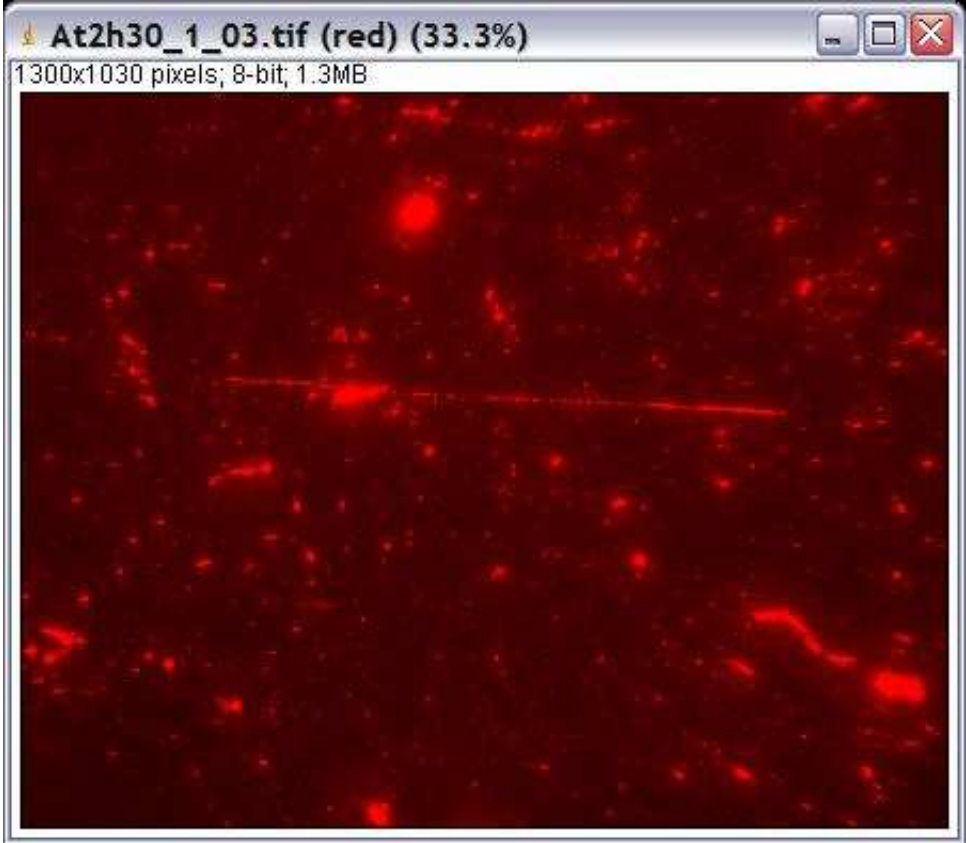
Operation: trace lines



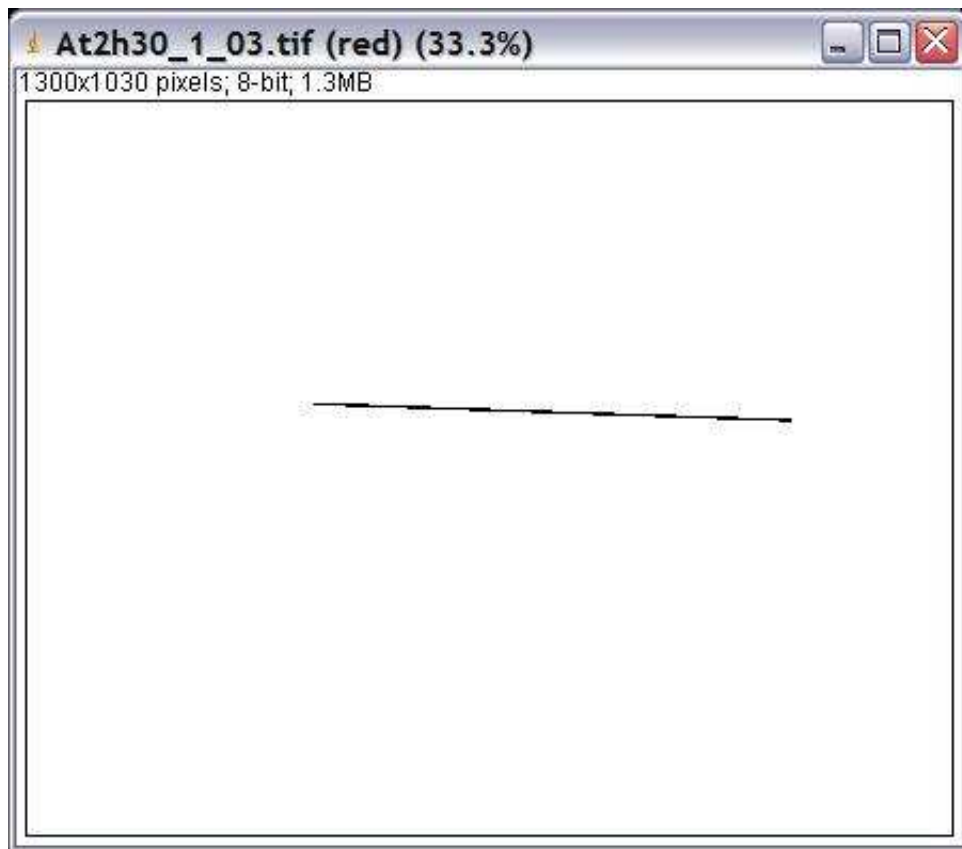
Author:

Volker Bäcker

Example



	X	Y	Z	Value
1	698	427	0	54



Description

The operation traces lines in a possibly noisy image, starting from a point on each line. A line can have gaps. Only lines with a given maximum angle against the horizontal are taken into account. Only lines with minimal length and a minimum distance from the image borders are taken into account.

Options



min stick length: To determine the next direction and to evaluate the stop condition a line segment of this length will be used. The tracing stops when the intensity under the segment of this length in the best next direction is smaller than in the direction perpendicular to the next best direction.

max angle: The next direction is searched within the angles -max angle to +max angle around the x-axis..

distance border: The minimum distance from the border.

min segment length: The minimum length of the line segments taken into account.

Parameter

inputImage (ImagePlus) : A 2-dimensional input image.

startPoints (ResultsTable): A results table containing the start points for the tracing in the columns X and Y.

Results

result (ImagePlus): An image showing the lines found.

resultLines (Vector of Line2D): A list of the lines found.

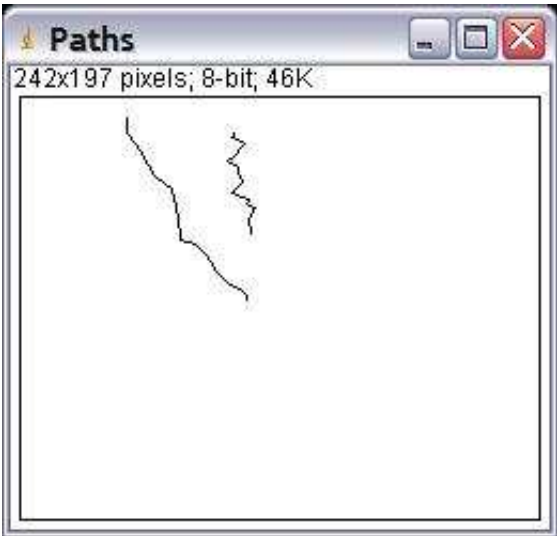
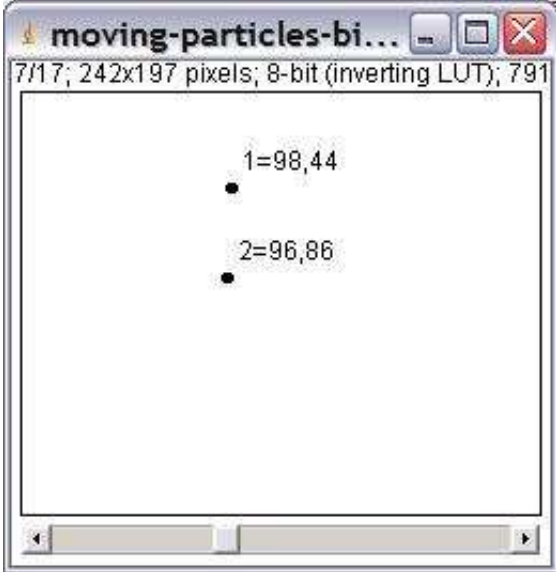
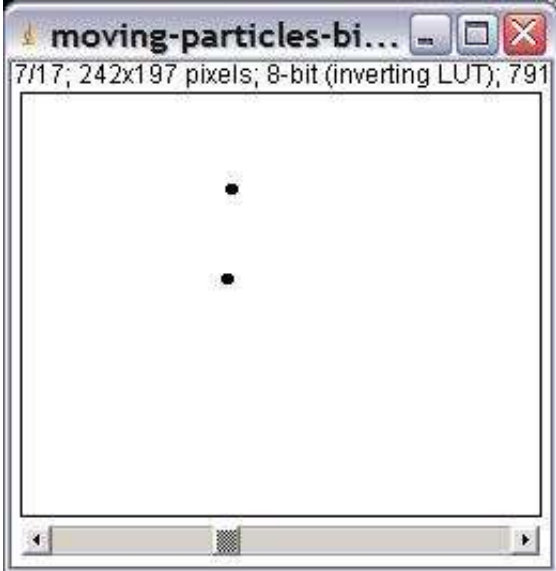
Operation: track particles



Author:

The operation uses the plugin MTrack2 from Nico Stuurman.

Example



Frame	X1	Y1	Flag1	X2	Y2	Flk
15	101.0	19.5		55.0	24.5	
16	98.0	18.5		49.0	16.5	
17	99.0	16.5		49.0	9.5	
Track	Length	Distance traveled	Nr of Frames			
1:	68.94808	47.67599	17			
2:	109.067215	101.788994	14			

Description

Track moving particles in a binary image and measure the distance traveled and the length of the path for each particle.

Options

track particles options	
? min size	1
? max size	999999
? min track length	2
? max velocity	30
? result filename	<input type="text"/> <input type="button" value="browse"/>
? show labels	<input checked="" type="checkbox"/>
? show positions	<input checked="" type="checkbox"/>
? show paths	<input checked="" type="checkbox"/>
? show path lengths	<input checked="" type="checkbox"/>
? max columns	75

min size: The minimal size of the particles.

max size: The maximal size of the particles.

min track length: Shorter paths are not taken into account.

max velocity: The maximum velocity of the particles.

result filename: If the result filename is set the results are not shown in a results table. Instead the user is asked for a filename when the operation is started and the measurements are written to that file.

show labels: If checked the labels of the objects will be displayed in the result stack.

show positions: If checked the positions of the objects are displayed in the result stack.

show paths: If checked an image showing the paths of all particles is created.

show paths lengths: If checked the lengths of the paths are reported, otherwise only the coordinates of all points on the paths are reported.

max columns: The maximal number of tracks that are reported next to each other. Further tracks are reported below.

Parameter

The only parameter is the input image. It must be a binary stack.

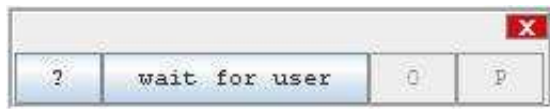
Results

Result (ImagePlus): The result stack showing the particles with their labels and positions.

PathsImage (ImagePlus): A 2-dimensional image showing the paths.

TrackMeasurements (TextPanel): The measured paths.

Operation: wait for user

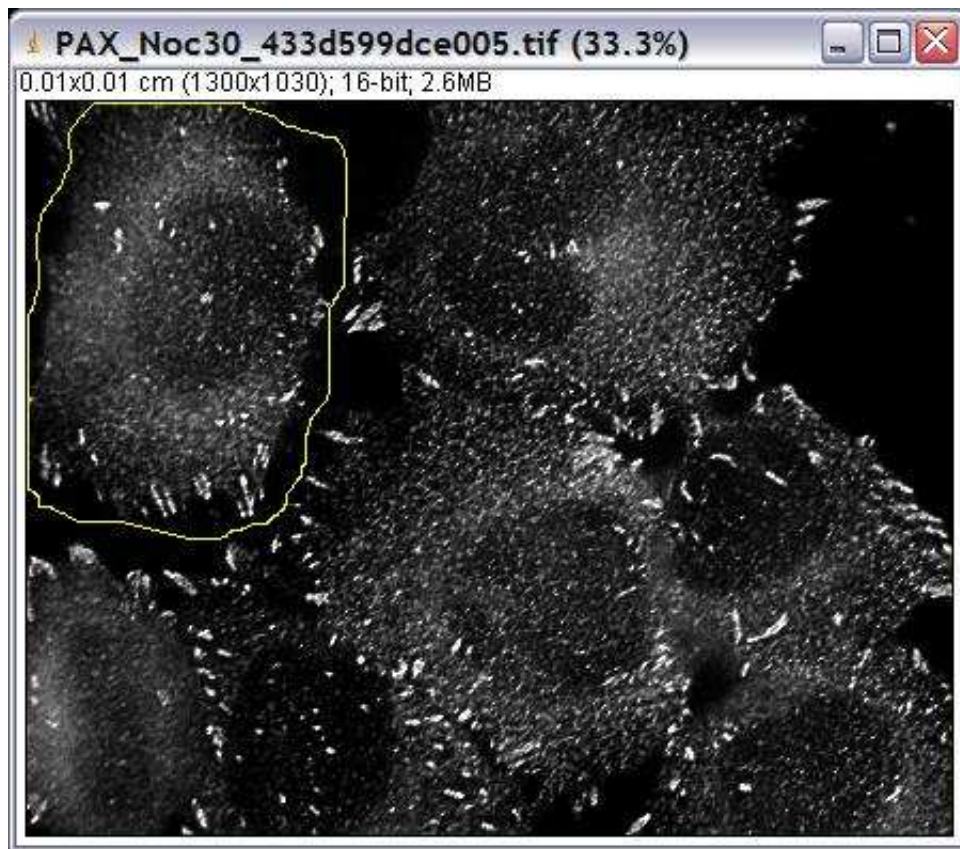


Author:

Volker Baecker

Example





Description

The operation pauses the application until the user presses the continue-button.

Options

The operation has no options.

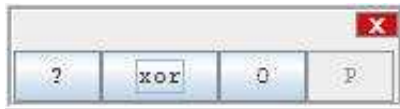
Parameter

The operation has no parameters.

Results

The operation has no results.

Operation: xor



Author:

Uses the method xor of the class ImageProcessor from ImageJ.

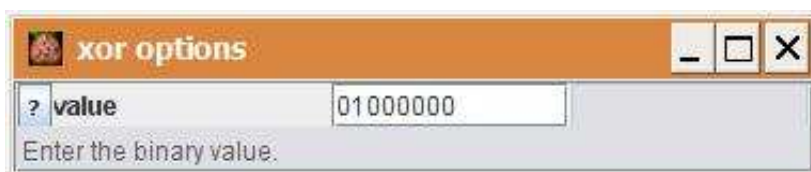
Example



Description

The operation replaces each intensity value in the image with the result of an exclusive or of the value and a constant.

Options



value: The value in binary form.

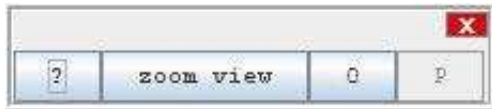
Parameter

The only parameter is the input image.

Results

The only result is the result image.

Operation: zoom view



Author:

Volker Bäcker

Example



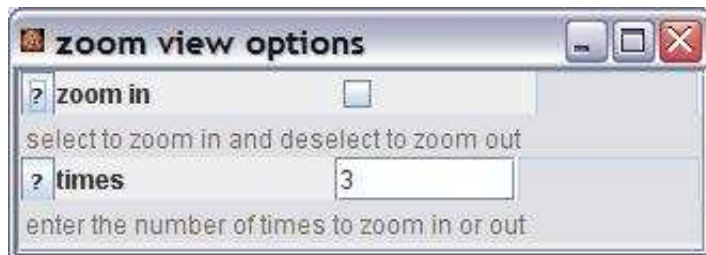
Zoom n-times into or out of the image.



Description

The operation zooms n-times into or out of the image.

Options



zoom in: The operation will zoom into the image if zoom in is checked and out of the image otherwise.
times: The number of times the zoom-in or zoom-out will be applied.

Parameter

The only parameter is the input image.

Results

The operation has no results.