IJPB Plugins user manual

David Legland and Ignacio Arganda-Carreras

October 8, 2014

Contents

1 Overview **1**

2 Installation instructions **2**
2.1 Installation in ImageJ 2
2.2 Installation in Fiji 2

3 Morphological filters **5**

4 Morphological reconstruction **7**
4.1 Principle 7
4.2 Applications 7
4.3 Usage 8

5 Watershed segmentation **9**
5.1 Classic Watershed plugin 9
5.1.1 Introduction 9
5.1.2 Usage 10
5.1.3 Over-segmentation 11
5.2 Marker-controlled Watershed plugin 12
5.2.1 Introduction 12
5.2.2 Usage 12
5.3 Morphological Segmentation plugin 13
5.3.1 Introduction 13
5.3.2 Usage 14
5.3.3 Macro language compatibility 16

6 Measurements **18**

7 Binary image utilities **19**

8 Label image utilities **21**

Abstract

The «ijpb-plugins» library for ImageJ/Fiji contains a variety of plugins for processing and analysing 2D or 3D images. It was originally designed for the study of plant cell images, but the algorithms are generic and can be applied to any type of image.

Chapter 1

Overview

The library implements several functionalities that were missing in the ImageJ software (Schneider et al. [2012]), and that were not or only partially covered by other plugins.

- morphological filtering for 2D/3D and binary or grey level images: erosion & dilation, closing & opening, morphological gradient & laplacian, top-hat...
- morphological reconstruction, for 2D/3D and binary or grey level images, allowing fast detection of regional or extended extrema, removing of borders, or hole filling
- watershed segmentation + GUI, making it possible to segment 2D/3D images of cell tissues
- 2D/3D measurements: volume, surface area, inertia ellipse/ellipsoid...
- utilities for manipulation of binary and label images

The home page of the project is hosted on GitHub: http://github.com/ijpb/ijpb-plugins.

Chapter 2

Installation instructions

2.1 Installation in ImageJ

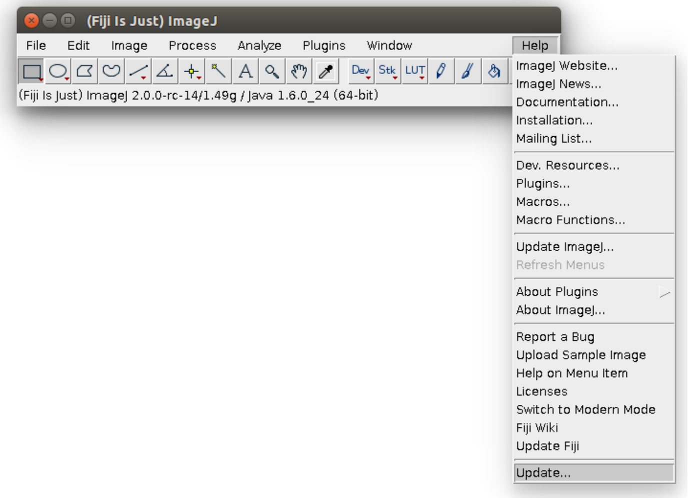
To install the IJPB-plugins toolkit in ImageJ, you only need to download the latest release (in the form of a JAR file) into ImageJ's plugins folder and restart ImageJ.

All released versions can be found at https://github.com/ijpb/ijpb-plugins/releases.

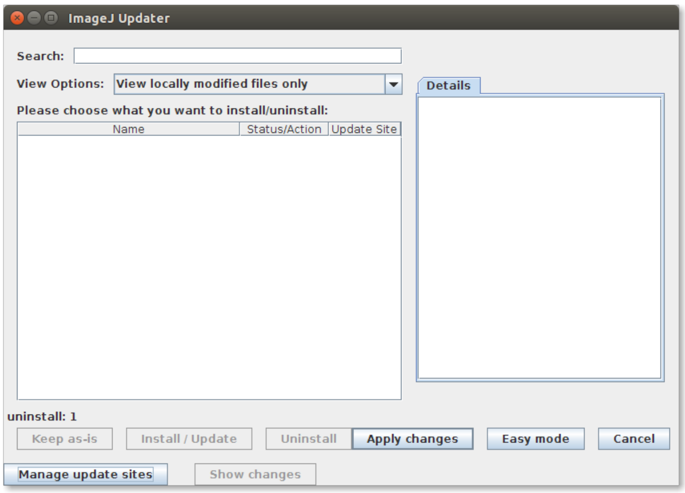
2.2 Installation in Fiji

If you use Fiji (Schindelin et al. [2012]), the IJPB-plugins toolkit can be easily installed by adding its update site to Fiji's list of update sites:

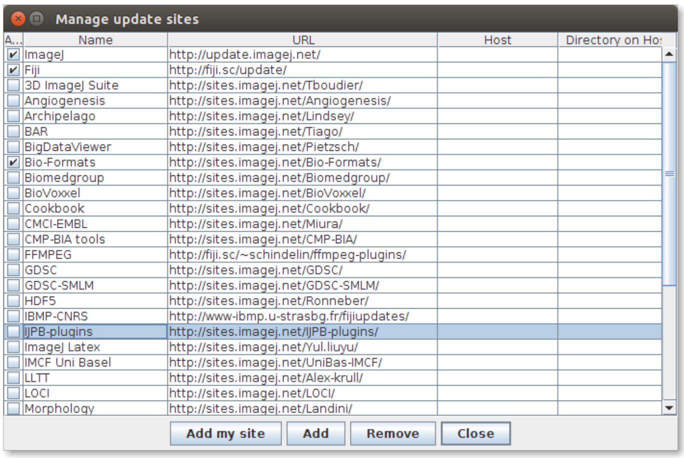
1. Open Fiji and select *Help > Update...* from the Fiji menu to start the updater.



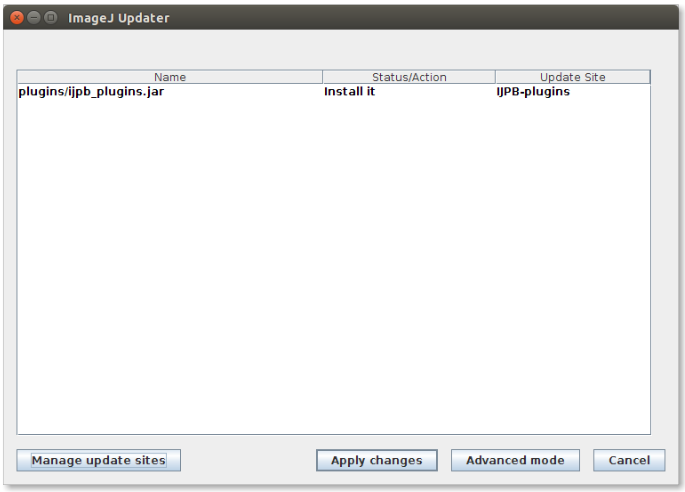
2. When the updater is open, click on *Manage update sites*.



3. This brings up a dialog where you can activate additional update sites:



4. Activate the IJPB-plugins update site and close the dialog. Now you should see an additional jar file for download:



5. Finally, click on *Apply changes* and restart Fiji. All the plugins should be available after restarting.

Chapter 3

Morphological filters

Morphological filters consider the neighborhood of each pixel/voxel according to a structuring element of a given size and shape.

Figure 3.1: Some example of morphological filters. Original image (black over white), result of dilation, result of erosion.

Figure 3.2: Some example of composed morphological filters. Result of morphological gradient, result of morphological closing, result of morphological opening.

The following filters are implemented both for 2D and 3D images, and work for binary as well as grey level images:

erosion keeps the minimum value within the neighborhood defined by the structuring element.

dilation keeps the maximum value within the neighborhood defined by the structuring element.

closing consists in the succession of a dilation with an erosion. Morphological closing makes dark structures smaller than the structuring element disappear.

opening consists in the succession of an erosion with a dilation. Morphological opening makes bright structures smaller than the structuring element disappear.

morphological gradient is defined as the difference of a morphological dilation and a morphological erosion with the same structuring element, and enhances edges of the original images.

morphological laplacien is defined as half the sum of a morphological dilation and a morphological erosion with the same structuring element, minus the original image, and enhances edges of the image.

black top-hat consists in subtracting the original image from the result of a morphological closing, and results in the enhancement of dark structures smaller than structuring element.

white top-hat consists in subtracting the result of a morphological opening from the original image, and results in the enhancement of bright structures smaller than structuring element.

The following structuring elements can be used for 2D images:

- disk
- square
- octagon
- diamond
- line with angle of 0, 90, 45 or 135 degrees

For 3D images, the same structuring elements can be used (the operation is simply repeated on each slice), as well as a cubic structuring element.

Chapter 4

Morphological reconstruction

The morphological reconstruction is at the basis of many useful algorithms, such as border removing, hole filling, or detection of regional minima or maxima in grey level images.

4.1 Principle

The principle of geodesic reconstruction is to repeat conditional dilations or erosions until idempotence. Conditional dilation is the result of a dilation, combined with a mask image using a logical operation. Conditional dilations are repeated until no more modification occur (idenpotence condition).

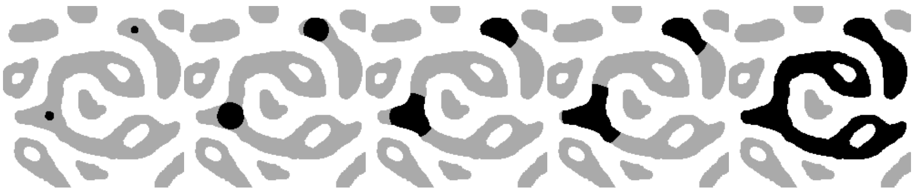


Figure 4.1: Principle of the geodesic reconstruction algorithm. Original image in gray with marker superimposed in black, and result of conditional dilations with increasing sizes.

4.2 Applications

By choosing the marker image, several operations may be automatised. For example, computing geodesic reconstruction with image of borders, and combining with original image will remove particles or regions touching the borders. In a similar way, combuting geodesic reconstruction by using the border of the complement of the image makes it possible to fill holes that may appear in particles.

Figure 4.2: Some applications of geodesic reconstruction. From left to right: original image, result of kill borders, result of fill holes.

4.3 Usage

The following operations are available in the "Plugins->Fast Morphology" menu:

Geodesic Reconstruction compute the geodesic reconstruction by erosion or dilation using a marker image and a mask image, and a specified connectivity. The same operation is available for 3D images.

Kill Borders remove the particles touching the border of a binary or grey level image, in 2D or 3D

Fill Holes remove holes inside particles in binary images, or remove dark regions surrounded by bright crests in grey level images. Works for both 2D and 3D images.

Regional Min / Max compute regional minima or extrema in grey level image, with specified connectivity

Extended Min / Max compute extended minima or extrema in grey level image, with specified connectivity

Impose Min / Max impose minima or maxima on a grey level image.

Chapter 5

Watershed segmentation

The watershed algorithm assimilates the grey level image to a digital elevation model, and aims at detecting the different catchment basins. In the grey-level image, the catchment basins correspond to dark regions surrounded by bright structures (the "crests"). It is a very popular technique specially used to segment touching objects. The IJPB-plugins suite contains several implementations of the algorithm, described in the following sections.

5.1 Classic Watershed plugin

5.1.1 Introduction

Classic Watershed is an ImageJ/Fiji plugin to perform watershed segmentation of grayscale 2D/3D images using flooding simulations as described by Soille and Vincent [1990].

The basic idea consists of considering the input image as topographic surface and placing a water source in each regional minimum of its relief. Next the entire relief is flooded from the sources and dams are placed where the different water sources meet.

All points in the surface at a given minimum constitute the **catchment basin** associated with that minimum. The **watersheds** are the zones dividing adjacent catchment basins.

The first image points that are reached by water are the points at the lowest grayscale value h_{min}, then all image pixels are progressively reached up to the highest level h_{max} (see Fig. 5.1).

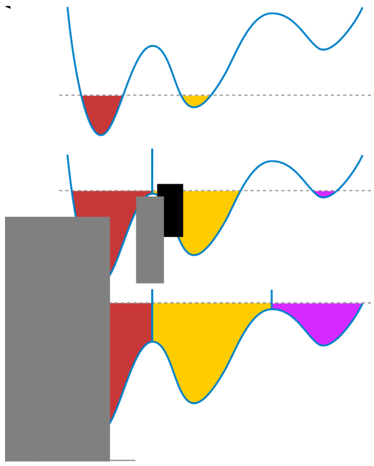
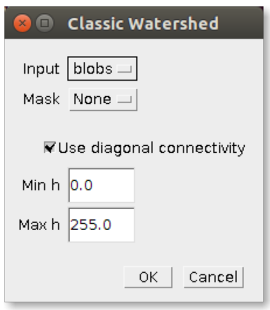


Figure 5.1: Schematic overview of watershed flooding in 1D

5.1.2 Usage

The Classic Watershed plugin runs on any grayscale image (8, 16 and 32-bit) in 2D and 3D. At least one image needs to be open in order for the plugin to run. If that's the case, a dialog like the following will pop up:



Let's have a look at the different parameters:

- Image parameters:
  - **Input image**: grayscale image to flood, usually the gradient of an image.
  - **Mask image** (optional): binary image of the same dimensions as the input image which can be used to restrict the areas of application of the algorithm. Set to "None" to run the method on the whole input image.
- Morphological parameters:
  - **Use diagonal connectivity**: select to allow the flooding in diagonal directions (8-connectivity in 2D and 26-connectivity in 3D).

– **Min h**: minimum grayscale value to start flooding from (by default, set to the minimum value of the image type).

– **Max h**: maximum grayscale value to reach with flooding (by default, set to the maximum value of the image type).

Output:

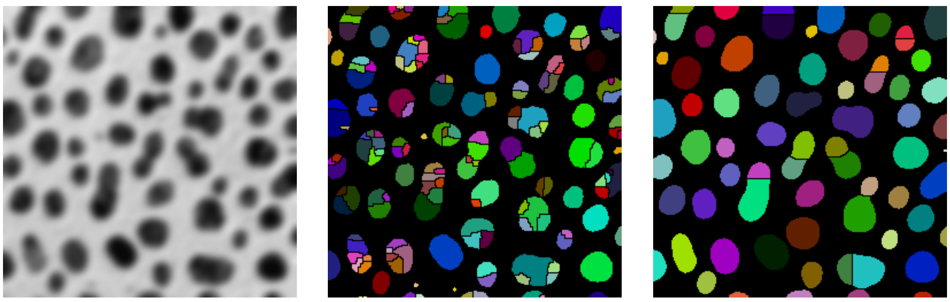
- **Labeled image** containing the resulting catchment basins (with integer values 1, 2, 3...) and watershed lines (with 0 value).

5.1.3 Over-segmentation

Normally, Classic Watershed will lead to an over-segmentation of the input image, especially for noisy images with many regional minima. In that case, it is recommended to either **pre-process the image before** running the plugin, **or merge regions based on a similarity criterion afterwards**. Several denoising methods are available in Fiji/ImageJ, namely: median filtering, Gaussian blur, bilateral filtering, etc.

Example: This short macro runs the plugin twice in the blobs sample, first without pre-processing and then after applying a Gaussian blur of radius 3:

```
// load the Blobs sample image
run("Blobs (25K)");
// invert LUT and pixel values to have dark blobs
run("Invert LUT");
run("Invert");
// run plugin on image
run("Classic Watershed", "input=blobs mask=None use min=0 max=150");
// apply LUT to facilitate result visualization
run("3-3-2 RGB");
// pre-process image with Gaussian blur selectWindow("blobs.gif");
run("Gaussian Blur...", "sigma=3");
rename("blobs-blur.gif");
// apply plugin on pre-processed image
run("Classic Watershed","input=blobs-blur mask=None use min=0 max=150");
// apply LUT to facilitate result visualization
run("3-3-2 RGB");
```



(a) Gaussian-blurred blobs image used as input (radius = 3).

(b) Watershed segmentation on original image ($h_{min} = 0$, $h_{max} = 150$).

(c) Watershed segmentation on Gaussian-blurred original image (radius = 3, $h_{min} = 0$, $h_{max} = 150$).

Figure 5.2: Input and result images from the macro example.

5.2 Marker-controlled Watershed plugin

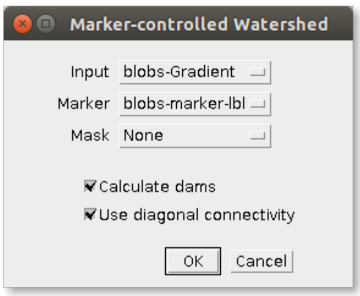
5.2.1 Introduction

Marker-controlled Watershed is an ImageJ/Fiji plugin to segment grayscale images of any type (8, 16 and 32-bit) in 2D and 3D based on the marker-controlled watershed algorithm by Meyer and Beucher [1990]. As the previous method, this algorithm considers the input image as a topographic surface (where higher pixel values mean higher altitude) but it simulates its flooding from **specific seed points or markers**. A common choice for the markers are the local minima of the gradient of the image, but the method works on any specific marker, either selected manually by the user or determined automatically by another algorithm (see Fig. 5.3).

Figure 5.3: Example of marker-controlled watershed segmentation on nucleus of Arabidopsis thaliana (image by courtesy of Javier Arpon, INRA-Versailles).

5.2.2 Usage

Marker-controlled Watershed needs at least two images to run. If that's the case, a dialog like the following will pop up:



Let's have a look at the different parameters:

- Image parameters:
  - The **Input image**: a 2D or 3D grayscale image to flood, usually the gradient of an image.
  - The **Marker image**: an image of the same dimensions as the input containing the seed points or markers as connected regions of voxels, each of them with a different label. They correspond usually to the local minima of the input image, but they can be set arbitrarily.
- And it can optionally admit a third image:

– The **Mask image**: a binary image of the same dimensions as input and marker which can be used to restrict the areas of application of the algorithm. Set to "None" to run the method on the whole input image.

- Rest of parameters:

  – **Calculate dams**: select to enable the calculation of watershed lines. Use diagonal connectivity: select to allow the flooding in diagonal directions.

Output:

- **Labeled image** containing the catchment basins and (optionally) watershed lines (dams).

5.3 Morphological Segmentation plugin

5.3.1 Introduction

Morphological Segmentation is an ImageJ/Fiji plugin that combines morphological operations, such as extended minima and morphological gradient, with watershed flooding algorithms to segment grayscale images of any type (8, 16 and 32-bit) in 2D and 3D.

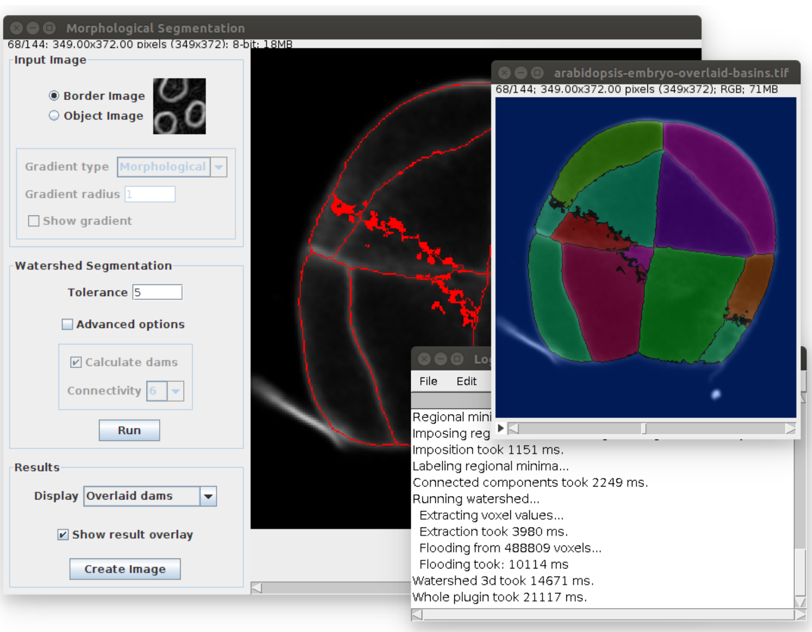


Figure 5.4: Overview of the Morphological Segmentation plugin.

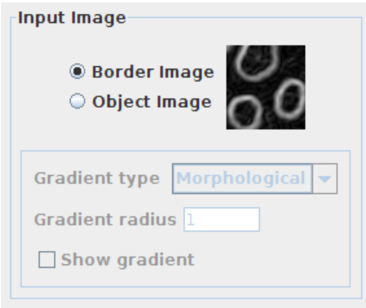
5.3.2 Usage

Morphological Segmentation runs on any open grayscale image, single 2D image or (3D) stack. If no image is open when calling the plugin, an Open dialog will pop up.

The user can pan, zoom in and out, or scroll between slices (if the input image is a stack) in the main canvas as if it were any other ImageJ window. On the left side of the canvas there are three panels of parameters, one for the input image, one with the watershed parameters and one for the output options. All buttons, checkboxes and input panels contain a short explanation of their functionality that is displayed when the cursor lingers over them.

Image pre-processing: some pre-processing is included in the plugin to facilitate the segmentation task. However, other pre-preprocessing may be required depending on the input image. It is up to the user to decide what filtering may be most appropriate upstream.

5.3.2.1 Input Image panel



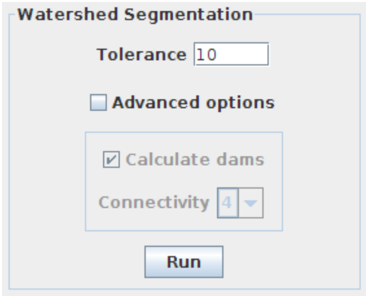
First, you need to indicate the nature of the input image to process. This is a **key parameter** since the watershed algorithm is expecting an image where the boundaries of objects present high intensity values (usually as a result of a gradient or edge detection filtering).

You should select:

- **Border Image**: if your input image has highlighted object boundaries.
- **Object Image**: if the borders of the objects do not have higher intensity values than the rest of voxels in the image.

When selecting "Object Image", an additional set of options is enabled to choose the type of gradient and radius (in pixels) to apply to the input image before starting the morphological operations. Finally, a checkbox allows displaying the gradient image instead of the input image in the main canvas of the plugin (only after running the watershed segmentation).

5.3.2.2 Watershed Segmentation panel



This panel is reserved to the parameters involved in the segmentation pipeline. By default, only the tolerance can be changed. Clicking on "Advanced options" enables the rest of options.

- **Tolerance**: dynamic of intensity for the search of regional minima (in the extended-minima transform, which is the regional minima of the H-minima transform, value of h). Increasing the tolerance value reduces the number of segments in the final result, while decreasing its value produces more object splits.
- **Calculate dams**: un-check this option to produce segmentations without watershed lines.
- **Connectivity**: voxel connectivity (4-8 in 2D, and 6-26 in 3D). Selecting non-diagonal connectivity (4 or 6) usually provides more rounded objects.

Finally, **click on "Run" to launch the segmentation**.

If your segmentation is taking too long or you want **to stop it** for any reason, you can do so by clicking on the same button (which should read "STOP" during that process).

5.3.2.3 Results panel

Only enabled after running the segmentation.

- **Display**: list of options to display the segmentation results.
  - **Overlaid basins**: colored objects overlaying the input image (with or without dams depending on the selected option in the Watershed Segmentation panel).
  - **Overlaid dams**: overlay the watershed dams in red on top of the input image (only works if "Calculate dams" is checked).

- **Catchment basins**: colored objects.
- **Watershed lines**: binary image showing the watershed lines in black and the objects in white (only works if "Calculate dams" is checked).

- **Show result overlay**: toggle result overlay.
- **Create image button**: create a new image with the results displayed in the canvas.

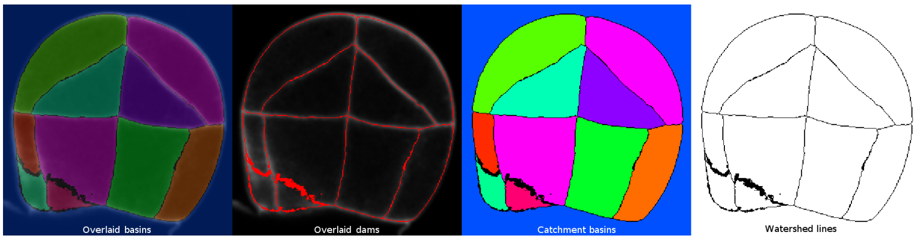


Figure 5.5: Examples of the 4 different display options.

5.3.3 Macro language compatibility

Morphological Segmentation is completely compatible with the popular ImageJ macro language[1]. Each of the buttons in the GUI are macro-recordable and their commands can be reproduced later from a simple macro file.

The complete list of commands is as follows:

- Start the plugin:

```
run("Morphological Segmentation");
```

- Select input image:

```
// select as object image
call("inra.ijpb.plugins.MorphologicalSegmentation.
    setInputImageType", "object");
// select as border image
call("inra.ijpb.plugins.MorphologicalSegmentation.
    setInputImageType", "border");
```

- Run segmentation with specific options:

```
call("inra.ijpb.plugins.MorphologicalSegmentation.segment","
    tolerance=10", "calculateDams=true","connectivity=6");
```

- Toggle result overlay:

```
call("inra.ijpb.plugins.MorphologicalSegmentation.toggleOverlay");
```

- Set option to display gradient image:

[1]http://imagej.net/developer/macro/macros.html

```
call("inra.ijpb.plugins.MorphologicalSegmentation.setShowGradient", "true");
```

- Select display format:

```
// Overlaid basins
call("inra.ijpb.plugins.MorphologicalSegmentation.setDisplayFormat", "Overlaid basins");
// Overlaid dams
call("inra.ijpb.plugins.MorphologicalSegmentation.setDisplayFormat", "Overlaid dams");
// Catchment basins
call("inra.ijpb.plugins.MorphologicalSegmentation.setDisplayFormat", "Catchment basins");
// Watershed lines
call("inra.ijpb.plugins.MorphologicalSegmentation.setDisplayFormat", "Watershed lines");
```

- Create new image with the current result:

```
call("inra.ijpb.plugins.MorphologicalSegmentation.createResultImage");
```

Complete macro example

The following macro opens the Blobs sample image, applies the plugin with a tolerance value of 32 and displays the result as overlaid dams.

```
// load the Blobs sample image
run("Blobs (25K)");
// run the plugin
run("Morphological Segmentation");
// wait for the plugin GUI to load
wait(1000);
// select input image as "object"
call("inra.ijpb.plugins.MorphologicalSegmentation.setInputImageType", "object");
// set gradient radius as 1
call("inra.ijpb.plugins.MorphologicalSegmentation.setGradientRadius", "1");
// run segmentation with tolerance 32, calculating the watershed dams,
// 4-connectivity
call("inra.ijpb.plugins.MorphologicalSegmentation.segment", "tolerance=32", "calculateDams=true", "connectivity=4");
// display the overlaid dams
call("inra.ijpb.plugins.MorphologicalSegmentation.setDisplayFormat", "Overlaid dams");
```

Chapter 6

Measurements

Several measurements are provided for binary or label 2D or 3D images. The results are provided in a ResultsTable, whose name contains the name of the original image.

Bounding box returns the minimal and maximal coordinates in each direction for each label

Volume / surface area computes the volume, the surface area, and a sphericity index, defined as $36\pi V^2/S^3$. Surface area is computed using a discretized version of the Crofton formula.

Inertia ellipse / ellipsoid returns the centroid (center of gravity) as well as the size and the orientation of the inertia ellipse or ellipsoid of each particle. Radiuses are sorted in decreasing order. Angles are given in degrees, and correspond to the azimut ("yaw"), the elevation ("pitch"), and the roll around the main axis.

Chapter 7

Binary image utilities

Some functions specific for the processing of binary images.

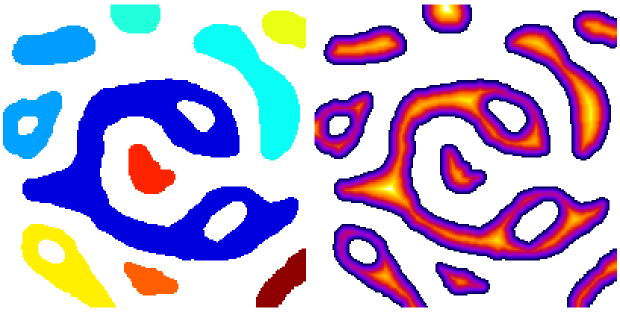


Figure 7.1: Connected components labeling, and computation of distance maps for binary image.

Connected Components Labeling transforms the binary image into a label image by assigning a specific number (label) to each connected component

Chamfer Distance Map computes an approximate distance map between each foregrond pixel to the nearest background pixel

Keep / Remove Largest Region identifies the largest connected component, and keep it or remove it

Size Opening computes the size (area in 2D, volume in 3D) of each connected component, and remove all particles whose size is below the value specified by the user.

Algorithms work for both 2D or 3D images. Default connectivity 4 (resp. 6) is used for 2D (resp. 3D) images.

Figure 7.2: Utilities for binary images. From left to right: keep largest region, remove largest region, apply size opening for keeping only regions with at least 1000 pixels.

Chapter 8

Label image utilities

Some functions specific for the processing of label images, in which the pixel/voxel values is used to identify the particle it belongs to. The value 0 is assumed to correspond to the background.

Figure 8.1: Utilities for label images. From left to right: original label image, remove border labels, remove largest region, apply size opening for keeping only regions with at least 1000 pixels.

Remove Border Labels similar to "kill borders" function, but operates faster as no morphological reconstruction is required.

Select Label(s) enters a set of labels, and creates a new label image containing only the selected labels.

Crop Label creates a new binary image containing only the label specified by the user. The size of the new image is fitted to the region.

Replace Value replaces the value of a region by another value. Can be used to "clear" a label, by replacing its value by 0, or to merge to adjacent regions.

Label Boundaries creates a new binary image containing value 255 for pixels/voxels having a neighbour with a different value.

Keep / Remove Largest Label identifies the largest label, and keep it or remove it

Label Size Opening computes the size (area in 2D, volume in 3D) of each region, and remove all labels whose size is below the value specified by the user.

Assign Measure To Label combines a label image with a results table, and creates a new image for which each pixel/voxel is assigned the measurement value corresponding to the label it belongs to.

Set Label Map allows to choose the color map used to display a label image. In particular, suffling the color map and/or choosing a specific color for background allows better visualisation that only grey levels

Label To RGB converts a label image to true RGB image. Similar to ImageJ native conversion, but this plugin avoids confusion between background pixels and regions with low labels.

Figure 8.2: Assign result of a measurment to a label image. In this example, the elongation is represented using a color code, between dark purple (circular) to yellow (very elongated).

Bibliography

Fernand Meyer and Serge Beucher. Morphological segmentation. *Journal of visual communication and image representation*, 1(1):21–46, 1990.

Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, et al. Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7):676–682, 2012.

Caroline A Schneider, Wayne S Rasband, Kevin W Eliceiri, Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, et al. 671 nih image to imagej: 25 years of image analysis. *Nature methods*, 9(7), 2012.

Pierre Soille and Luc M Vincent. Determining watersheds in digital pictures via flooding simulations. In *Lausanne-DL tentative*, pages 240–250. International Society for Optics and Photonics, 1990.