Rigaku

EPISODE

02

IMAGE PROCESSING WORKSHOP

*Companion eBook*

# REFINING SEGMENTATION USING IMAGEJ

*Presenter: Angela Criswell*

# You'll learn:

Segmentation of CT data can often result in errors, or falsely labeled pixels/voxels. Due to noise and other artifacts So, you'll need a set of tools that help you correct the errors and fine tune segmentation results prior to quantitative analysis.

In this workshop, you will learn how to refine your segmentation results using tools like morphological operations and further segment regions of interest in your segmented data.

You can also download the open-source image processing program ImageJ and try different morphological operations on sample data to have hands-on experience and deepen your understanding of image processing.

Here is the recording of the workshop.

# CONTENTS


04 [1. Why we need to refine segmentation?]

07 [2. How do we refine segmentation?]

14 [3. ImageJ hands-on exercises]

14 [About the tool]

24 [Take-aways]

# PRESENTERS

*Presenter: Angela Criswell*
*Senior Scientist*

Angela holds a PhD from Rice University and has been with Rigaku for 20 years. She started in the Macromolecular Crystallography Applications lab focusing on X-ray techniques to study structural biology. She has gained expertise in a number of X-ray methods in her tenure at Rigaku, including small angle X-ray scattering and X-ray computed tomography. Angela likes working with customers to find the best fit for their samples while addressing their specific experimental questions.

*Co-presenter: Aya Takase*
*Director of X-ray Imaging*

Aya holds a PhD in engineering from Osaka University and a MA in physics from Tokyo University of Science and has been with Rigaku for 23 years. She started in the X-ray Diffraction Application Lab and transitioned to X-ray Imaging in 2017. Her goal: Help non-expert X-ray users achieve expert results with less time and effort. She has worked on many projects designing automated and user-friendly X-ray instruments and analysis software. She is very passionate about helping people learn more about X-rays and working with X-ray users to solve their specific problems.

*Host: Tom Concolino*
*Southeast Regional Account Manager*

Tom holds a PhD in Chemistry from Mississippi State University and has been with Rigaku for 20 years. He started out in the Small Molecule Crystallography Applications Lab before transitioning to the sales team in 2002. He has been on the front lines helping clients save on time, cost, and effort while pushing forward to support the never-ending need to innovate and explore new materials and structures. From academia to mining to pharmaceutical research, Tom has learned the value of bringing a fresh perspective to each customer application while utilizing his vast experience to collaborate on the best fit solution for each and every customer.

1

# Why do we need to refine segmentation?

The ultimate goal of X-ray CT (computed tomography) data analysis is to quantify information about features we observe in CT images. To quantify features, we first need to segment the CT images. During segmentation, each pixel within an image is given a label that assigns it to a specific region of interest (ROI).

There are many tools that are used for segmentation, including thresholding, machine learning and deep learning. Though some are more powerful than others, it can be said that no segmentation tool is perfect. As a result, some pixels are incorrectly labeled. So, additional tools are required to fine tune segmentation results.

# EXAMPLE

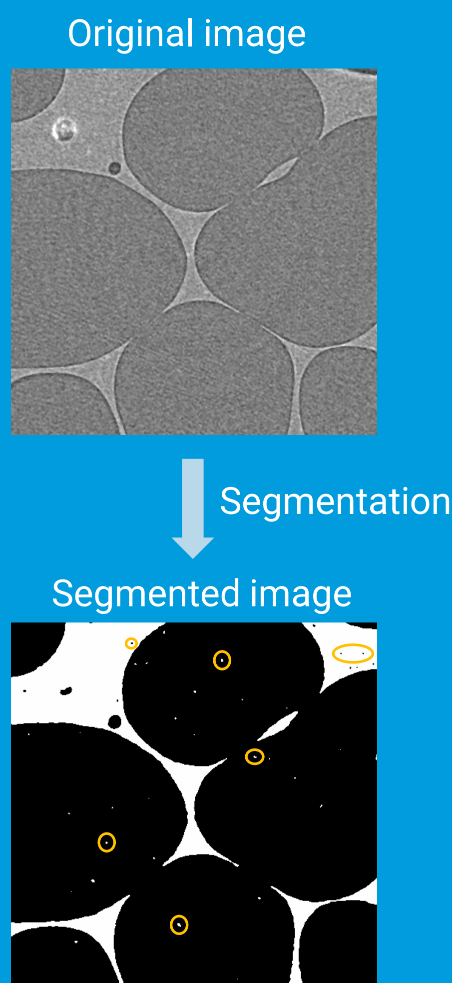


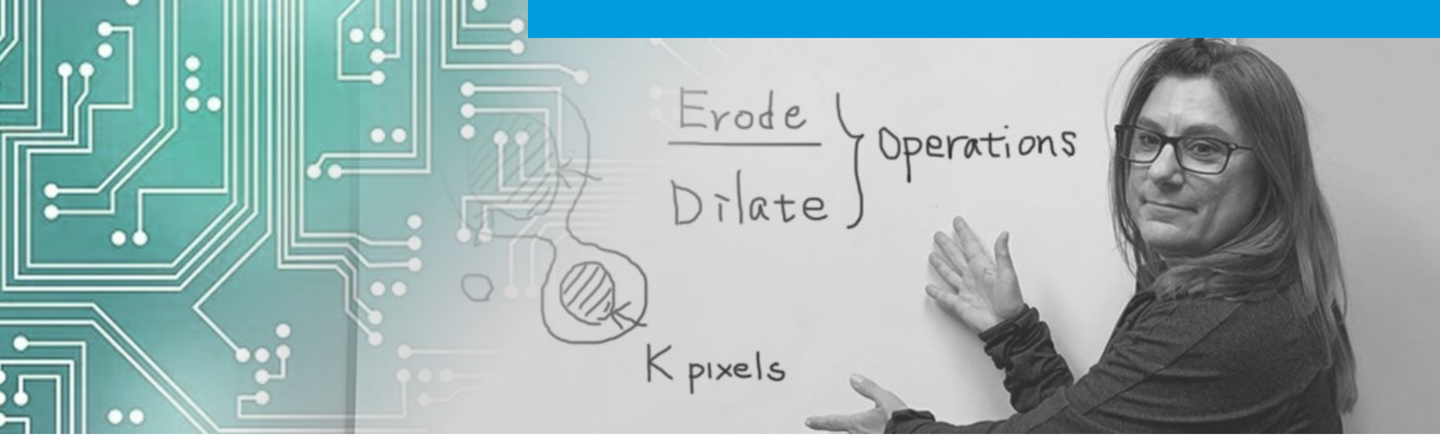

Figure 1: Examples of incorrectly labeled pixels (yellow circles) in a segmented image.

To illustrate incorrect labeling, Figure 1 shows an image of a foam sample where thresholding is used to assign pixels as either polymer (white) or void (black). Yellow circles indicate incorrectly labeled pixels where some void pixels are incorrectly assigned as polymer pixels and some polymer pixels are incorrectly assigned as void pixels. So, further refinement of these segmentation results is required.

In this presentation, we share some tools that are used to fine tune results to more accurately fit reality.

## Example

In Figure 1, we see an original image and the result of segmentation. In this example, there are several incorrectly labeled pixels, denoted by the yellow circles.



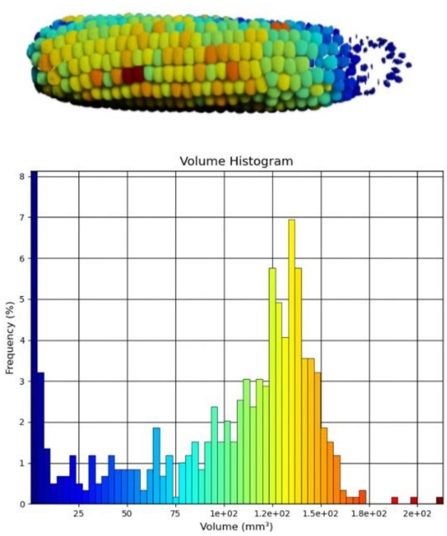




Figure 2: A 3D volume of maize with kernels colored according to their calculated volume (top) and the volume histogram for labeled kernels (bottom).

Additionally, we often want to further partition groups of pixels within ROIs. For example, we may want to examine a group of ROIs with respect to their size distribution. Figure 2 shows such an example where maize kernels are separated, and their size distribution can be quantified.

In other cases, we may want to exclude regions from our analysis, for example regions that intersect the image border. Luckily, many tools are available for post-segmentation refinement.

2

# How do we refine segmentation?

There are several tools used to refine segmentation results. These include operations that reassign incorrect pixels in ROIs, operations that look for connections within ROIs, tools that allow further segmentation of ROIs, and others. Here we will look at some of these tools and illustrate how they are used.

## Morphological operations

Morphological operations are commonly used tools for refining segmentation results. They work by probing an image with a structuring element (SE) to test how it intersects with an ROI. The structuring element probes the image at all locations and makes conclusions based on whether the element fits or does not fit within a region's shape.

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects, while erosion removes pixels on object boundaries. The number of pixels added or removed from objects depends on the size and shape of the SE used to process the image. Both dilation and erosion change the size and the shape of the object.

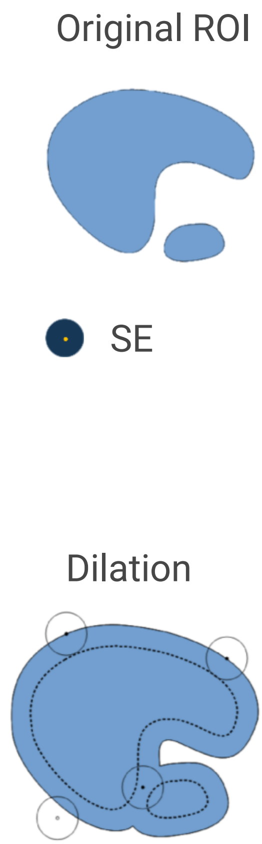



Figure 3: Dilation of an ROI using a disk-shaped structuring element.

## Dilation and Erosion

Dilation probes an ROI with the SE to test whether the SE intersects with pixels within the ROI. It results in shapes that are larger than those in the original ROI. Additionally, it creates connections between components and fills holes within the shape. Figure 3 illustrates an example of dilation. In this case an original image of light blue shapes is probed with a disc-shaped structuring element (SE).



Images from https://imagej.net/plugins/morpholibj

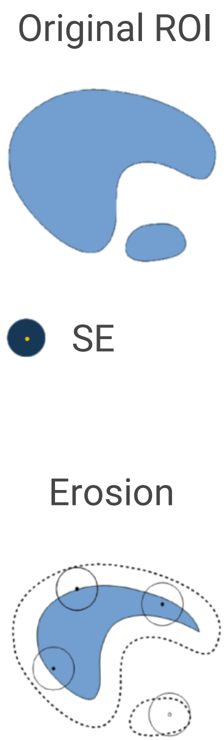
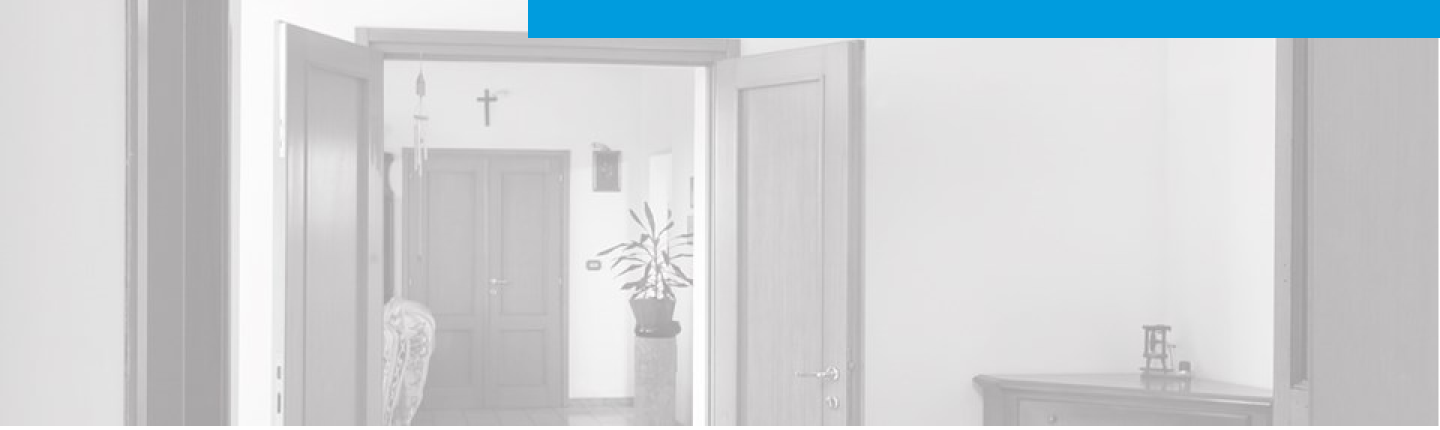



Figure 4: Erosion of an ROI using a disc-shaped structuring element.

Erosion probes the ROI to tests whether the SE is fully contained within the labeled area. It results in an ROI that is smaller than that of the original ROI. Additionally, it removes thin lines and small isolated objects that are thinner or smaller than the SE. It can also separate two objects that are connected by a thin line or “neck”. Figure 4 illustrates an example of how erosion works.



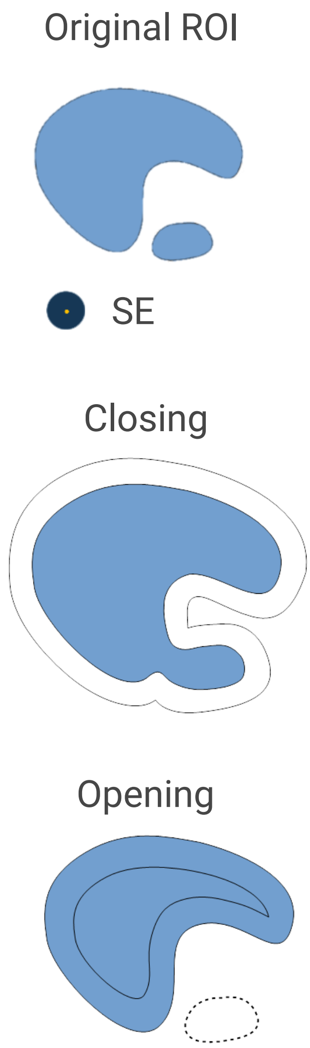



Figure 5: Closing and opening an ROI using a disc-shaped structuring element.

## Closing and Opening

Often, erosion and dilation are used in combination. For example, the result of dilation followed by erosion is called closing. Closing removes small holes within an ROI. It can also connect structures that were separated by a thin space. In a symmetric way, the result of erosion followed by dilation is called opening. Opening removes structures smaller than the SE. It also separates two objects that are connected by a thin line or a “neck”.

Figure 5 illustrates the results of opening and closing.



Images from https://imagej.net/plugins/morpholibj

## Kill Borders

The kill borders tool removes objects touching the border of an image. This can be useful when you want to eliminate the objects that do not represent their accurate shape or size because they are truncated by the image border.

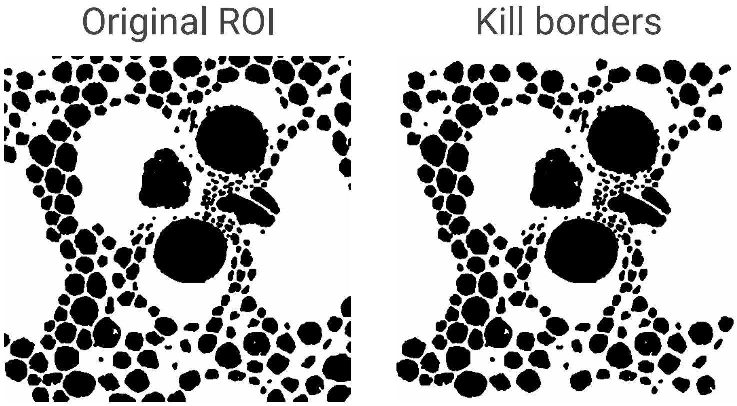



Figure 6: An example of the kill borders operation.

## Fill Holes

Fill holes removes holes inside shapes in an image. Fill Holes has the benefit that the ROI shape is not changed, however you should be careful when using the tool because this tool could fill large "real" holes that shouldn't be filled. Figures 7 illustrate how fill holes works and illustrates how “real” holes are also incorrectly filled.

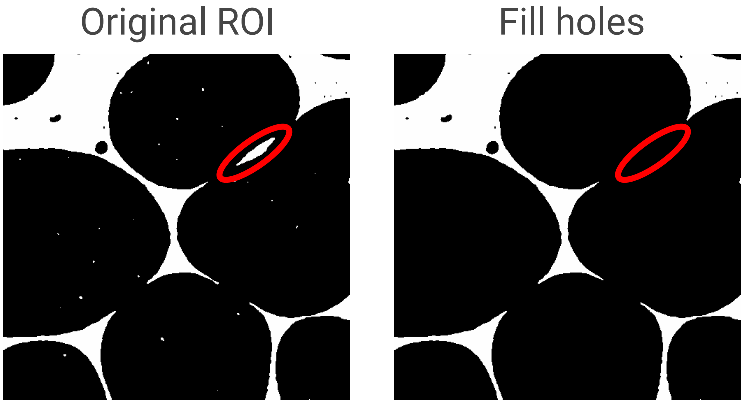
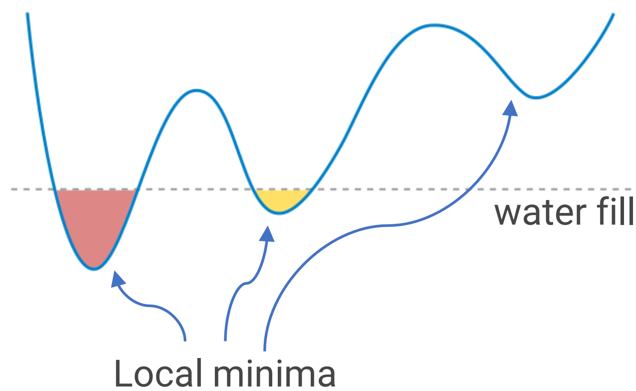


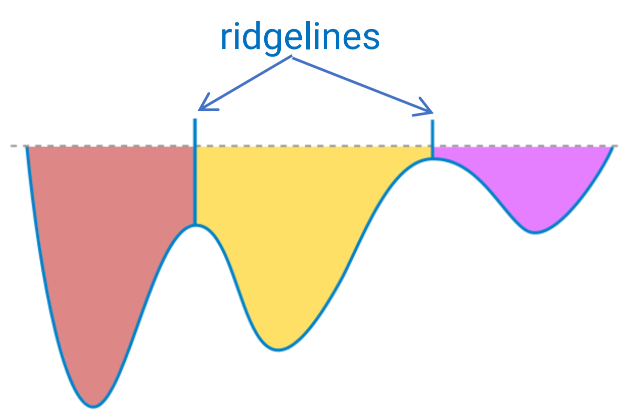

Figure 7: An example of the fill holes operation.

## Watershed Transformation

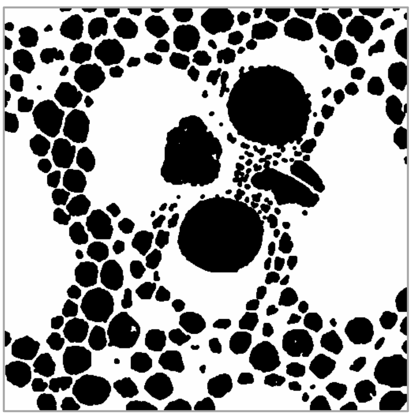
The watershed transformation is used to separate connected components within an ROI into separate objects. Let's consider the picture below where each object in an ROI is a basin. In the image below, there are three basins corresponding to three adjacent objects within the ROI. Now imaging that it starts raining and the basins start to fill.




Eventually the basins fill and bodies of water between neighboring objects will start to touch. This point of contact is marked as a ridgeline, or edge, of the object.




Segmented image

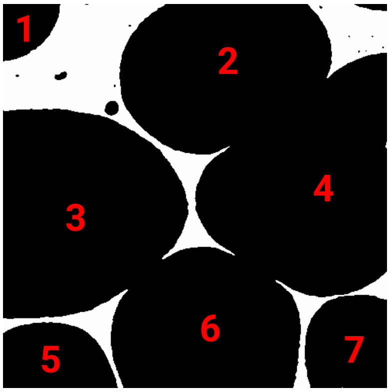


Following watershed

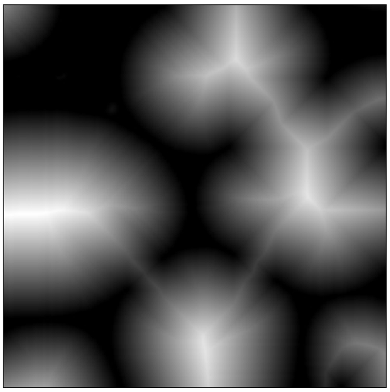
Figure 8: An example of object separation using the watershed transformation and recoloring ROIs.



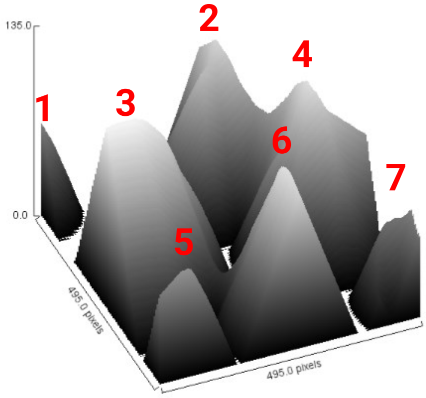
Images adapted from https://imagej.net/plugins/classic-watershed original image




distance transform



3D intensity plot



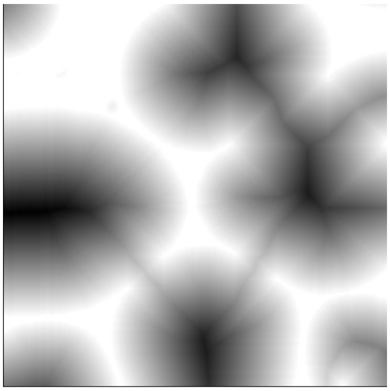

To perform watershed, we use a distance transform. As we saw earlier, the watershed operation requires that each object in the ROI be a ‘basin.’ So, how do we do transform objects in our ROI into basins?

Let’s consider our segmented foam image shown on the top left, where 7 objects within the ROI are labeled. What we need to do is to transform this segmented image into one containing basins. To do that, we use the distance transform.

The distance transform computes the distance between each labeled void pixel and the nearest foam pixel. On the middle left, we see the result of this calculation.

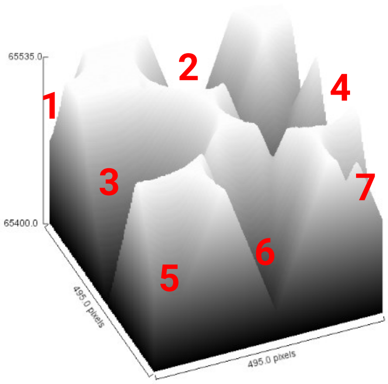
If we then plot the 3D intensity of this image, we get the result on the bottom left. This plot shows that rather than basins, each of the objects corresponds to peaks. So, we need to invert the distance transform before we can perform the watershed operation.

inverted distance transform



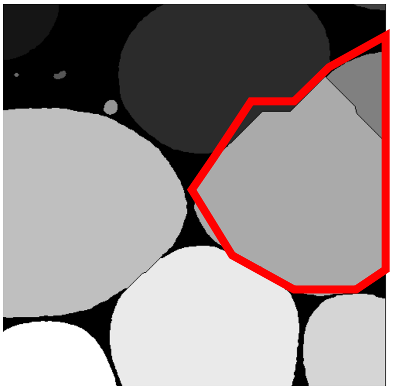
The top left image shows an inverted distance transform.

3D intensity plot




If we plot the 3D intensity of this image, we get the result on the middle left. This plot shows each of the objects now corresponds to a basin. Now, we're ready to perform the watershed operation.

watershed result



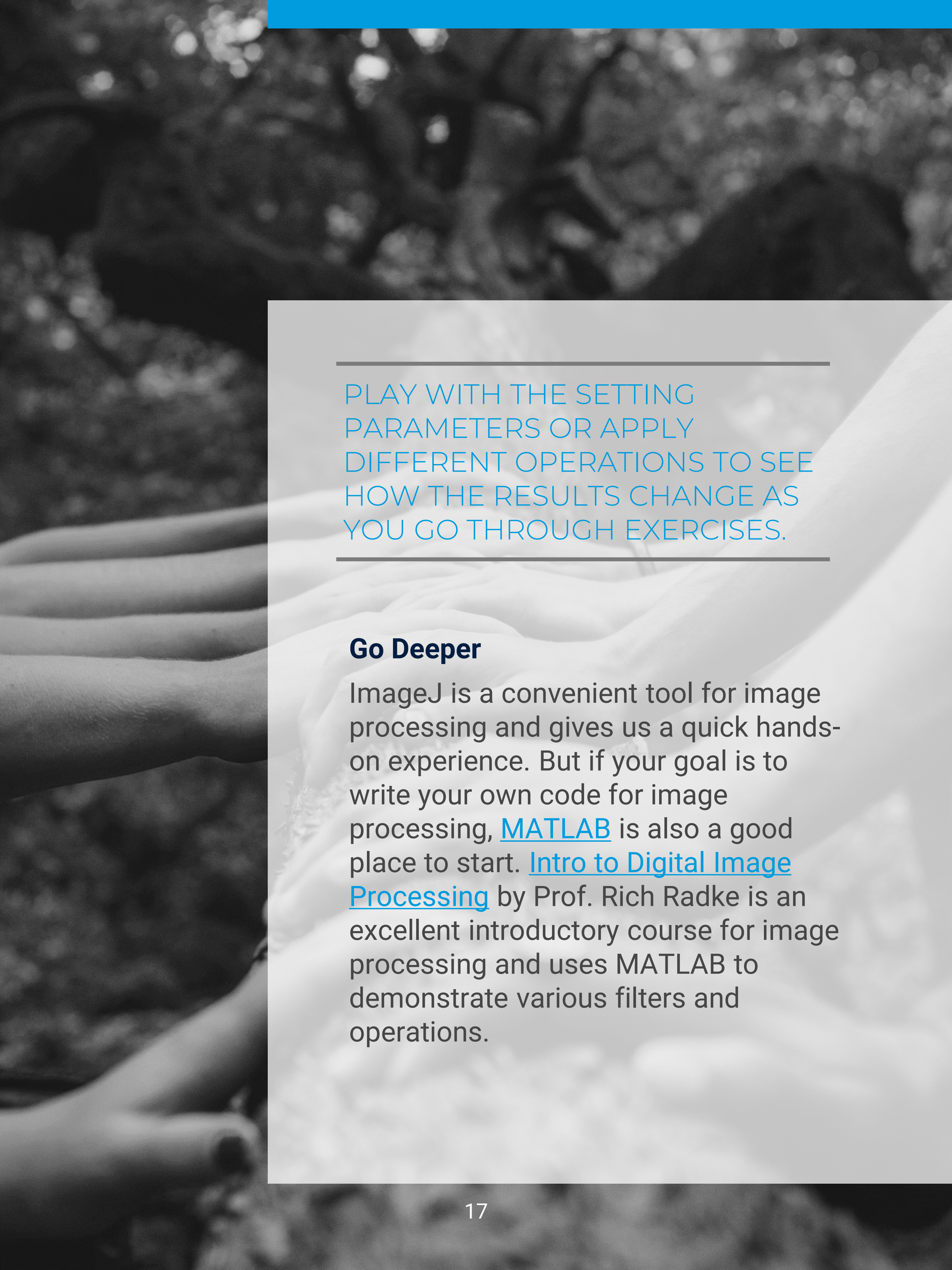
The bottom left image shows the result of the watershed operation on the inverted distance transform. Notice that the results of the watershed are not perfect. The red circle shows where there should be a single object however it is divided into to parts This is called oversegmentation. This can easily be fixed by merging objects.

# 3 ImageJ hands-on exercises

Hands-on exercises help us understand how various tools can be used to refine our segmentation results.

In this section, we will use the MorphoLibJ plugin to show how to use morphological operations, perform watershed transformation and use other tools to refine segmented images using an open-source Fiji distribution of ImageJ.

If you are new to ImageJ, watch this Mini Tutorial: ImageJ Getting Started Guide.



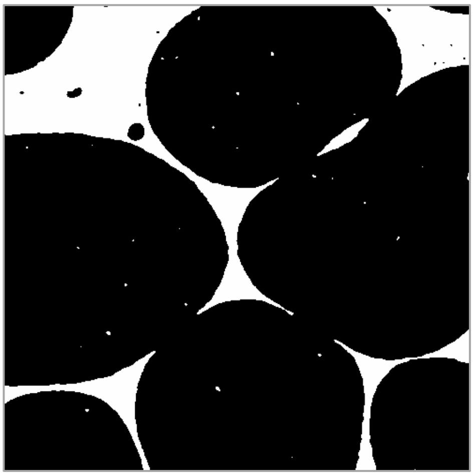
PLAY WITH THE SETTING PARAMETERS OR APPLY DIFFERENT OPERATIONS TO SEE HOW THE RESULTS CHANGE AS YOU GO THROUGH EXERCISES.

## Go Deeper

ImageJ is a convenient tool for image processing and gives us a quick hands-on experience. But if your goal is to write your own code for image processing, MATLAB is also a good place to start. Intro to Digital Image Processing by Prof. Rich Radke is an excellent introductory course for image processing and uses MATLAB to demonstrate various filters and operations.

## Dilation and Erosion

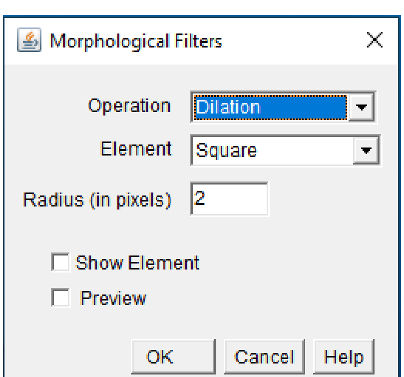
Let's look at dilation and erosion filters and how they change the original binary image.



**Original binary image**: Segmented CT image of a foam material. 495 x 495 pixels, 1.06 microns/pixel

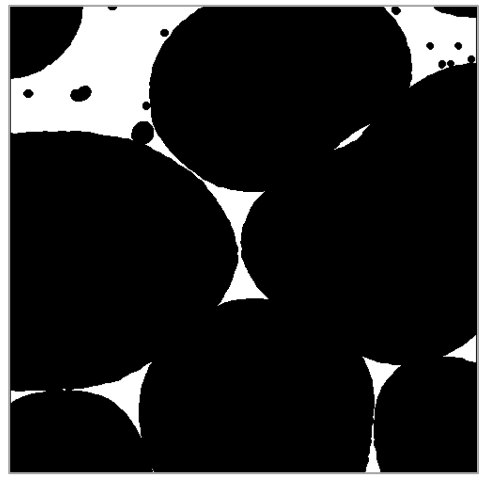
## Applying custom filters

(Plugins → MorphoLibJ → Filtering → Morphological Filters)

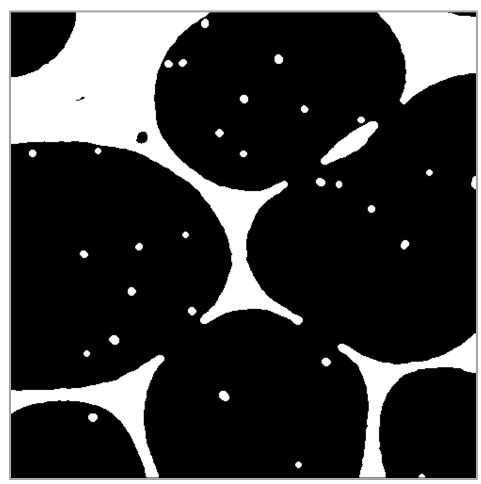



Select Dilation or Erosion operation from drop-down menu and select a structuring element. Turn on "Preview" to see the effect.

### 1. Dilation, disk, 3 pix radius



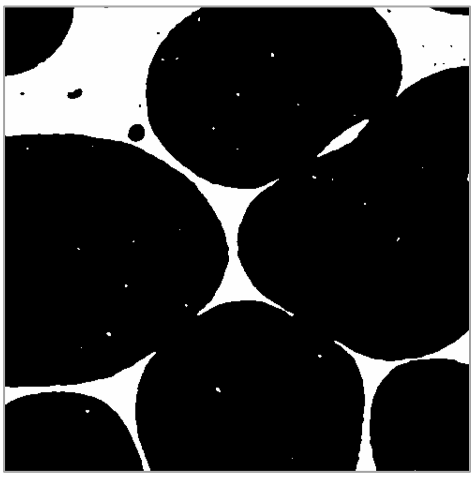
### 2. Erosion, disk, 3 pix radius



Try different operations and structuring element types/sizes to get a feel of how they work.

## Closing and Opening

Now let's have a look at Closing and Opening to see how they work on the same image.

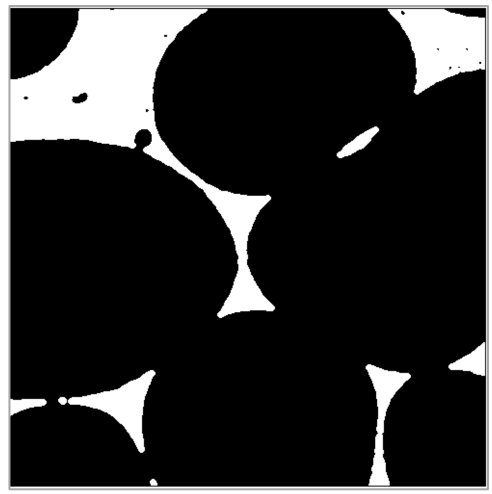


**Original binary image**: Segmented CT image of a foam material. 495 x 495 pixels, 1.06 microns/pixel

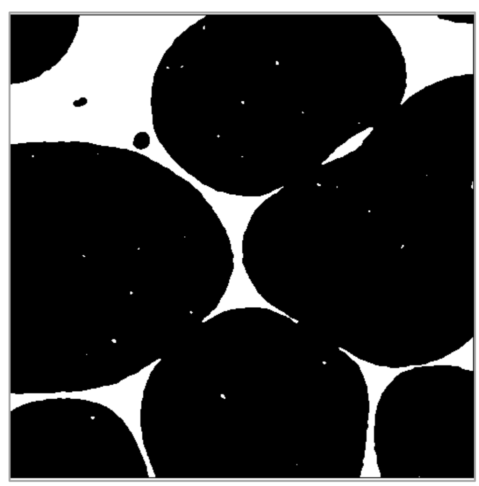
We'll use the same Morphological Filters dialog and select Closing or Opening operations.

Select Closing or Opening operation from drop-down menu and select a structuring element. Turn on "Preview" to see the effect.

**1. Closing, Square, 2 pix radius**



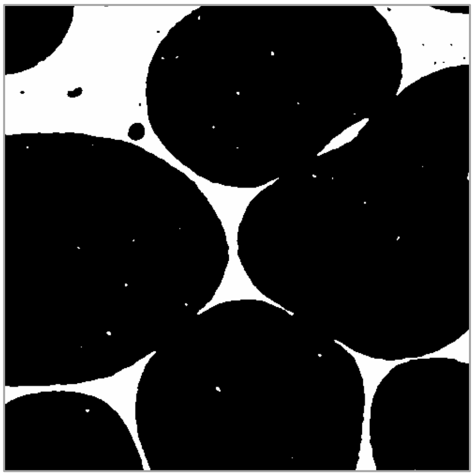
**2. Opening, Square, 2 pix radius**



Try different operations and structuring element types/sizes to get a feel of how they work.

## Kill Borders

Now let's have a look at the effect of Kill Borders on the same image.

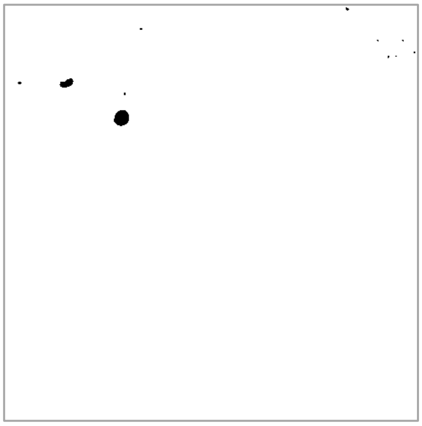


**Original binary image**: Segmented CT image of a foam material. 495 x 495 pixels, 1.06 microns/pixel

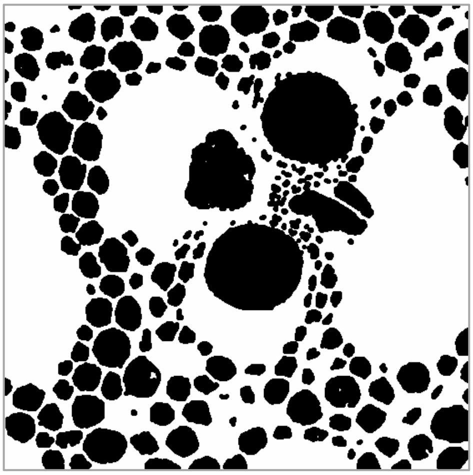
### Applying Kill Borders

(Plugins → MorphoLibJ → Filtering → Kill Borders)

Because all large voids intersect with the image edge, most are removed.

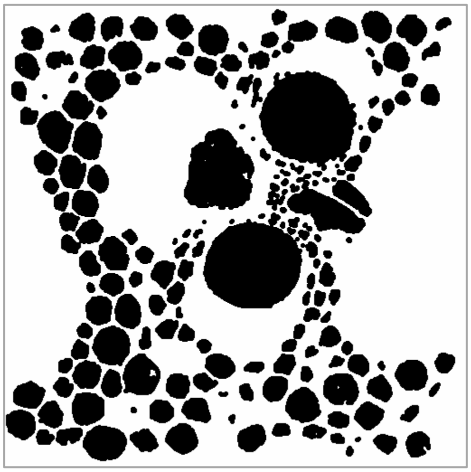


Let's have a look at a different image.



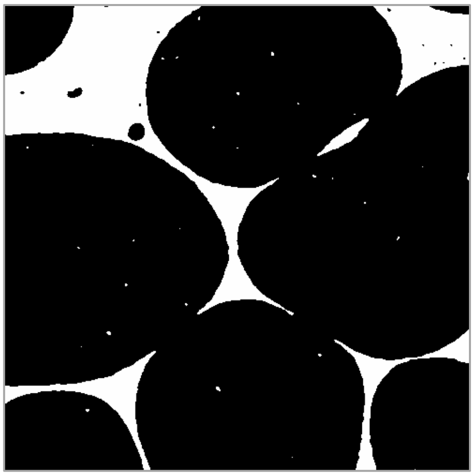
**Original image**: Segmented CT image of bamboo. 647 x 647 pixels, 1.27 microns/pixel

In this case, fewer voids intersect with image edges, so most voids are retained. This tool is useful in that it eliminates truncated objects so that any quantitative analyses are not biased by partial objects.



## Fill Holes

Now let's have a look at the effect of Fill Holes on the segmented foam image.

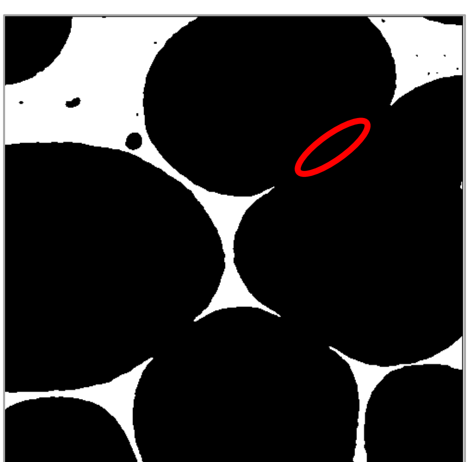


**Original binary image**: Segmented CT image of a foam material. 495 x 495 pixels, 1.06 microns/pixel

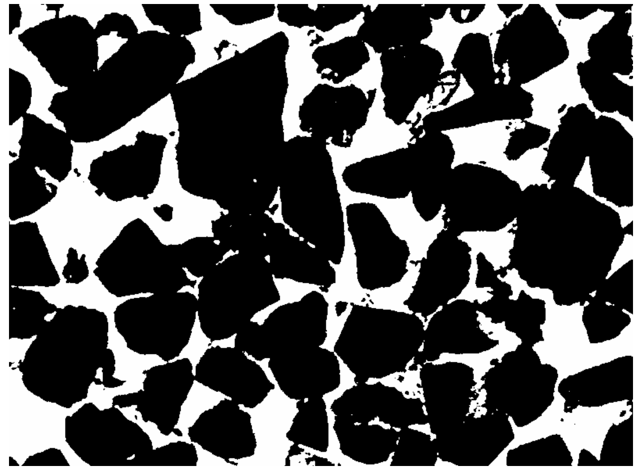
### Applying Fill Holes

(Plugins → MorphoLibJ → Filtering → Kill Borders)

This tool fills holes without changing the shape of voids but can fill "real" holes (red).



Let's have a look with a different image.

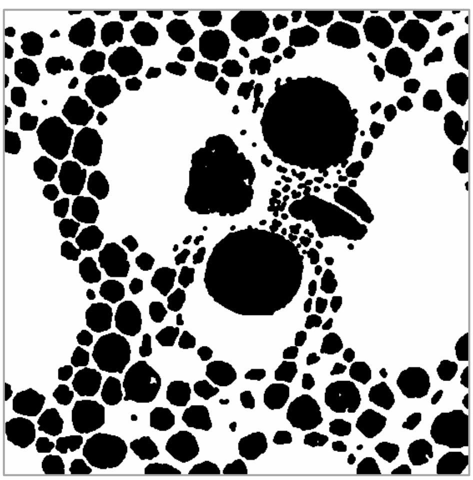


**Original binary image**: Segmented CT image of sandstone. 799 x 559 pixels, 5 microns/pixel

Note that fill holes has a deleterious effect when used on this image because several quartz grains are merged together. In this case, you might try closing instead.

## Watershed

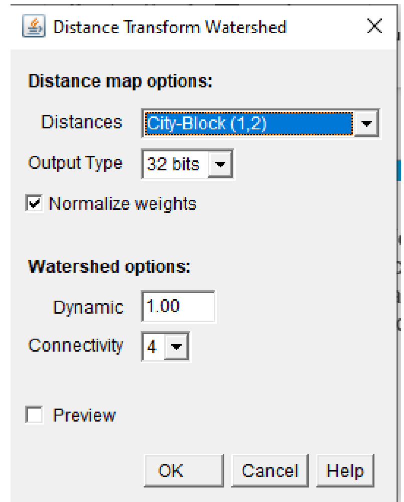
First, let's try watershed on an easy example. 'we will use the Distance Transform Watershed that works on binary images and allows separation of touching objects by combining the distance transform and watershed methods



**Original binary image**: Segmented CT image of bamboo. 647 x 647 pixels, 1.27 microns/pixel

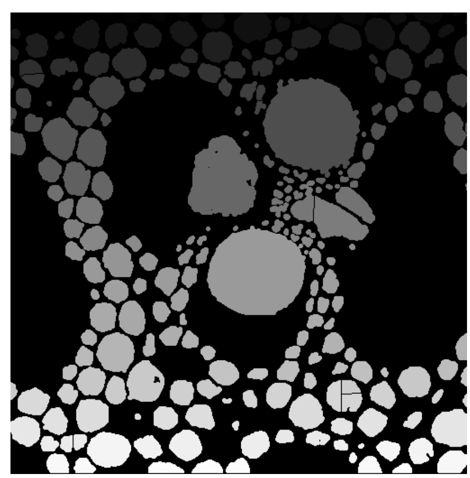
### Applying distance transform watershed

(Plugins → MorphoLibJ → Binary → Distance Transform Watershed)



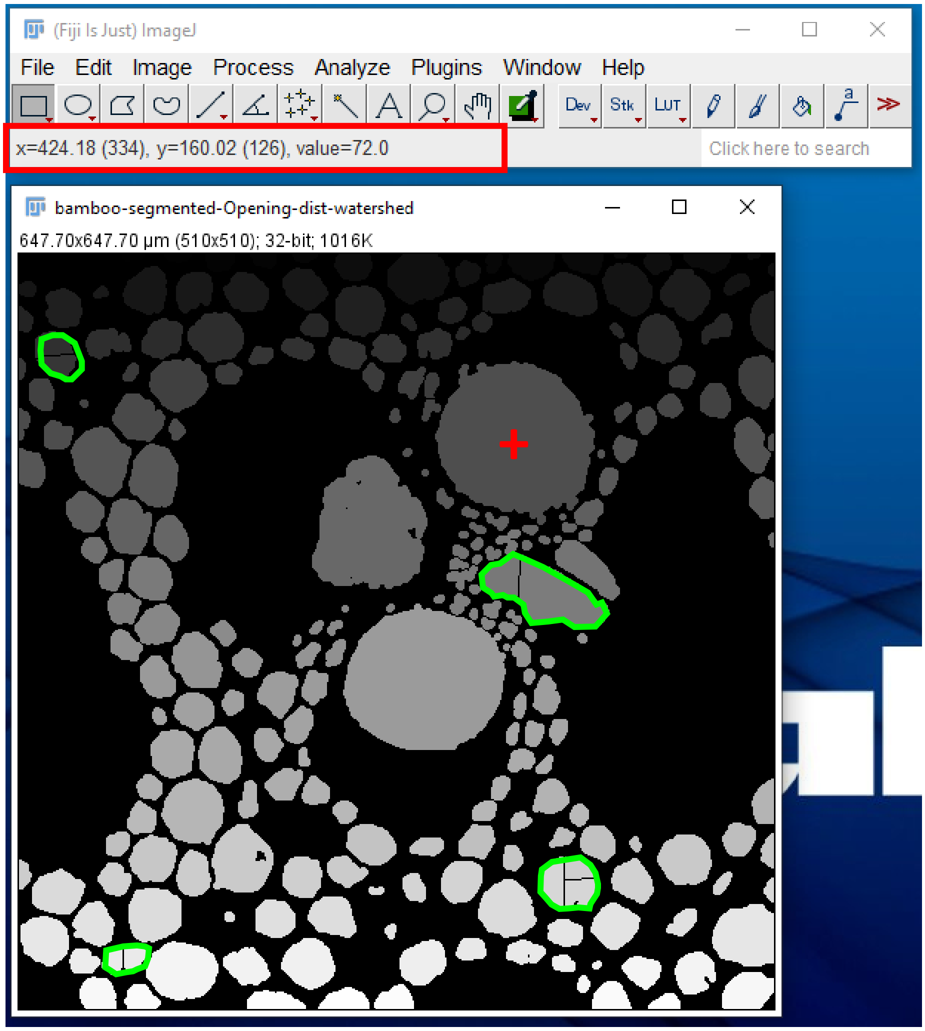

Distance map options allow you to select among pre-defined weights to compute the distance transform. Different options will cause changes to the shape of object borders.

### City-Block (1-2) result



The default coloring for segmented objects is a 2D 32-bit labeled image where each object is assigned a separate index.

To identify the index # for each object, simply hover over an object (indicated by red +) and the corresponding index value is displayed in the ImageJ status bar (indicated by red rectangle). After sorting, we can hover our mouse over an object (red cross) and it's label will be shown in the ImageJ status bar. This object is #72. We can then look at the table to inspect the calculated area.

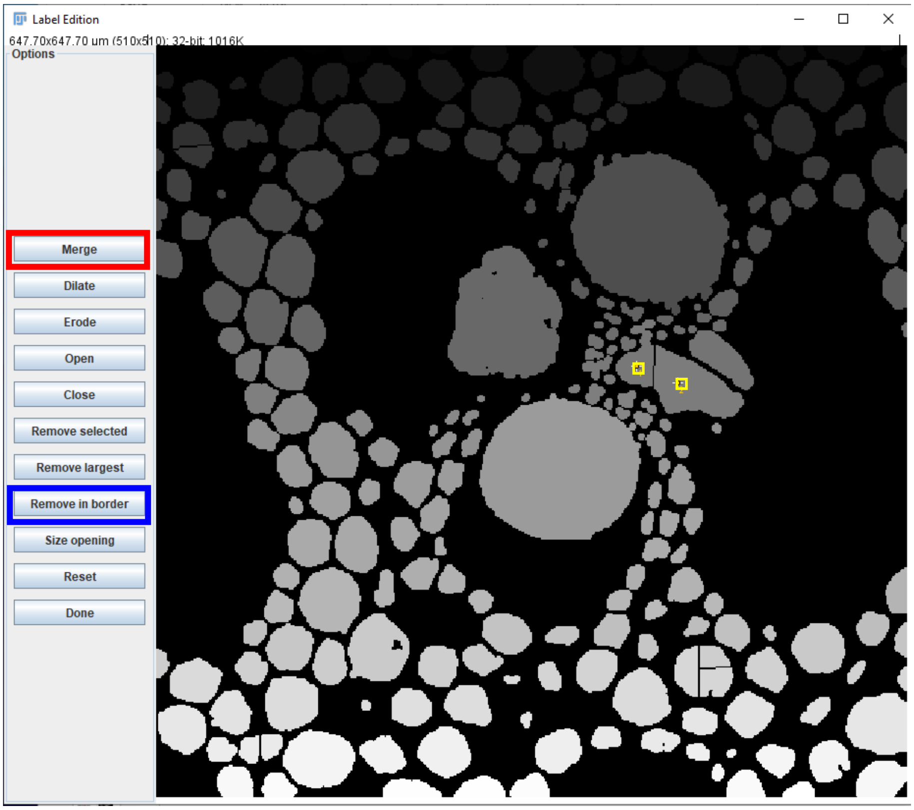



You'll notice in this case there are examples of oversegmentation (indicated by green outlined objects). The next section will describe how to merge these into a single object.

## Merging oversegmented objects

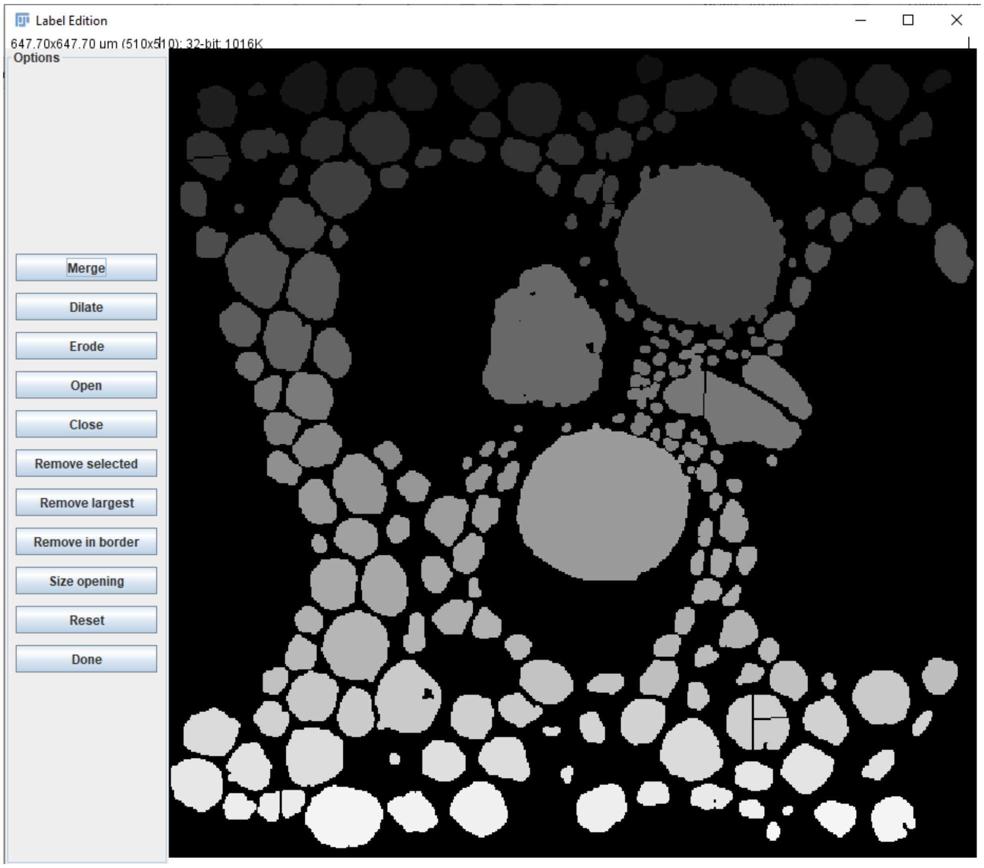
(Plugins → MorphoLibJ → Labels →Label Images → Label Edition)

To merge an oversegmented object, click on the objects in the image portion of the Label Images window. You will see a small yellow box and numbers appear. Then, click the Merge button (red rectangle) on the top left to merge these selections into a single object.




Repeat for each of the oversegmented objects in the image. When you've finished, click the 'Remove In Border' button (blue rectangle) to run the Kill Borders operation.

You should see the following in the Label Edition window.




Don’t be concerned that the previously merged objects still look separate. If you have merged them properly, then when you hover over the ImageJ status bar, the objects should have the same index #.

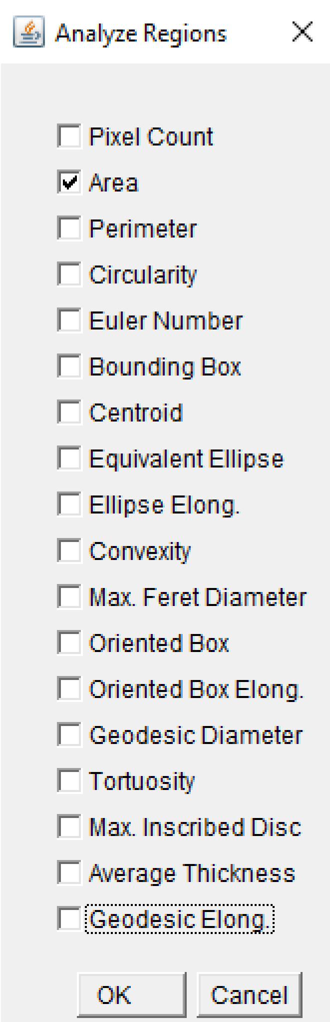
Now you’re ready to perform quantitative analyses of your objects.

Let's plot the area for each object.

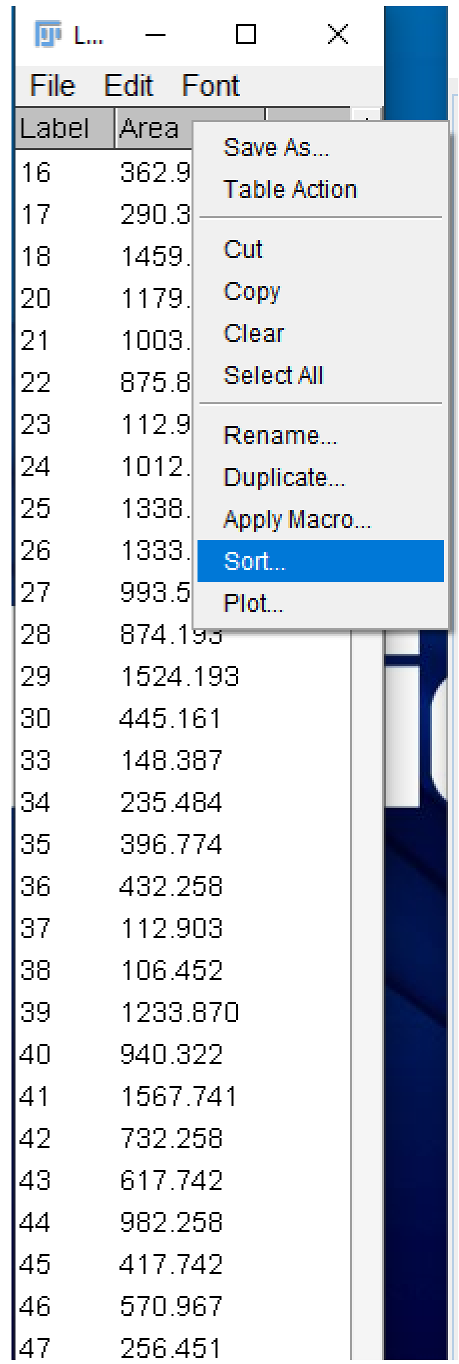
**Quantitative analysis of objects**

(Plugins → MorphoLibJ → Analyze → Analyze Regions)

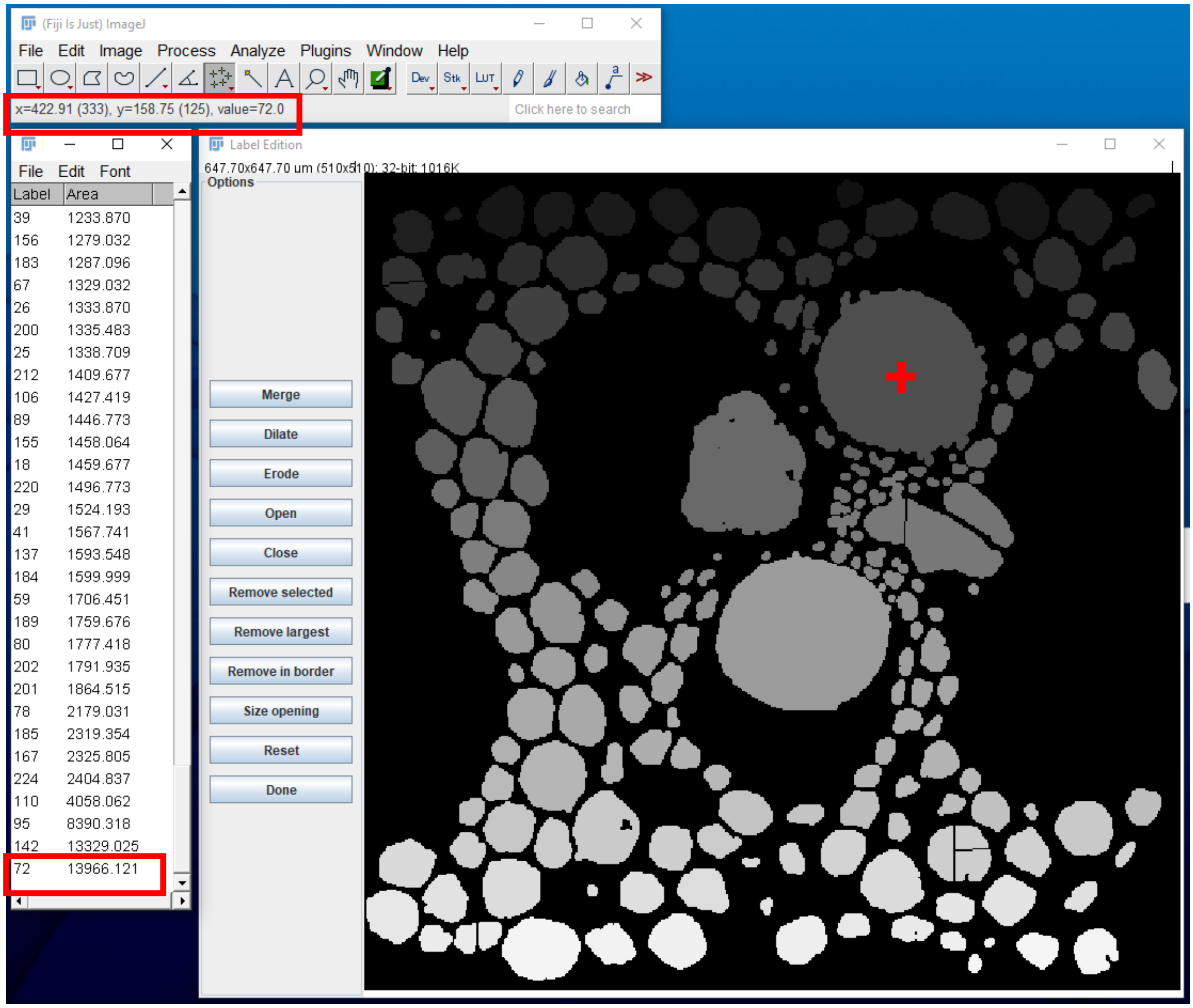
Uncheck all boxes except the Area checkbox.




Let's look at the result of Area Calculation. We can sort the results by right-clicking on the gray table header.



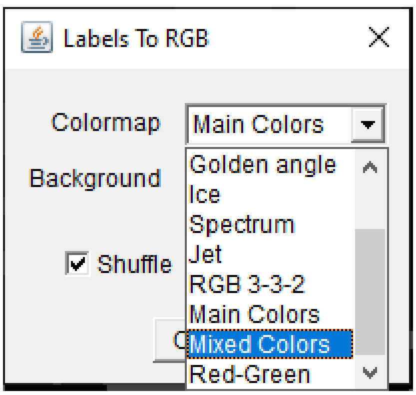
After sorting, we can hover our mouse over an object (red cross) and it's label will be shown in the ImageJ status bar. This object is #72. We can then look at the table to inspect the calculated area.

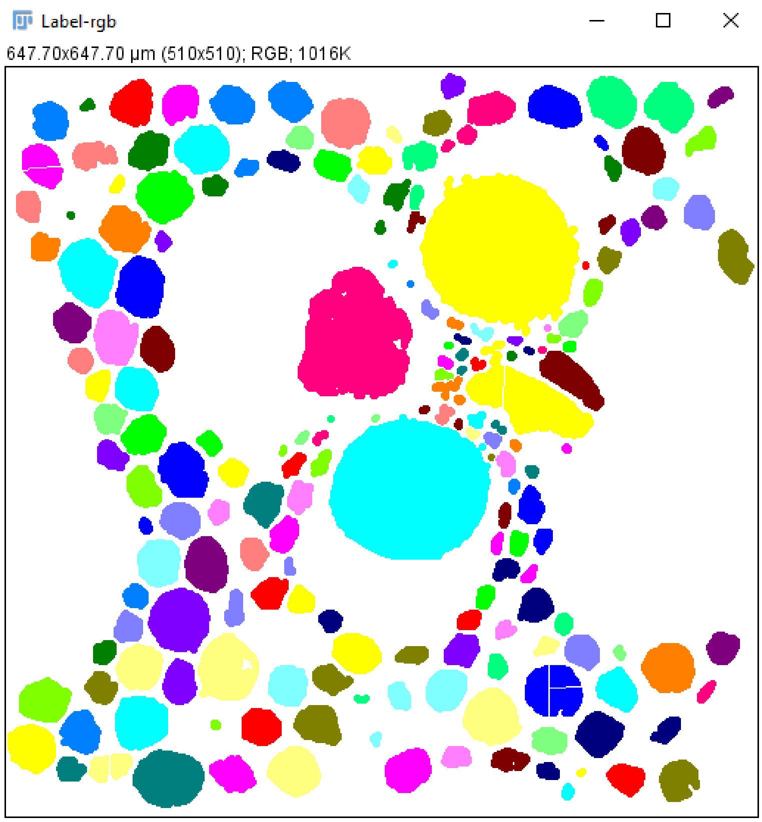


The default coloring for of the segmented objects can be changed from grayscale to color. Let’s try Mixed Colors to recolor segmented objects.

### Recolor segmented objects

Plugins → MorphoLibJ → Label Images → Set Label Map.




An RGB image will open to display the image with the selected colormap as shown above on the right.

# ABOUT THE TOOLS

## Fiji: A distribution of ImageJ

ImageJ is an open-source image processing program for scientific multidimensional images. We used Fiji distribution in this workshop. You can download Fiji for Windows, Mac, or Linux from this link:

https://imagej.net/software/fiji/downloads

## MorphoLibJ plugin

The MorphoLibJ plugin used in this workshop is available from this link:

https://imagej.net/plugins/morpholibj

Original work: Legland, D. *et al*. doi: 10.1093/bioinformatics/btw413

# Takeaways

Segmentation is always the first step of data analysis. During segmentation, each pixel within an image is given a label that assigns it to a specific ROI. Though we have very powerful tools to use for segmentation, no tool is perfect. As a result, some pixels are incorrectly labelled. So, additional tools are required to fine tune segmentation results.

Here we shared some tools for refining our segmentation results, including Dilation and Erosion, Closing and Opening, Kill Borders and Fill Holes. We also looked at how we can separate objects within an ROI using the watershed operation.

## LET'S LEARN TOGETHER

Many people have learned what X-ray computed tomography (CT) is, how it works, and where it can be helpful in our webinar and workshop series. All recordings, application examples, a publication list, and blog articles are available at imaging.rigaku.com.

Subscribe to the email updates to stay informed about new articles, recommended publications and books, and upcoming learning events.


**CONTACT US**

imaging@rigaku.com