



Image processing and analysis with ImageJ

Workshop:
Image processing and analysis with ImageJ

Montpellier RIO Imaging

Volker Baecker

02.03.2012



Table of Contents

1. Introduction.....	4
2. Installing and setting up ImageJ.....	4
2.1 Installing ImageJ.....	4
2.2 Memory Settings.....	4
2.3 Upgrading ImageJ.....	5
2.4 Installing loci-bioformats	5
2.5 Associating file types with ImageJ.....	5
3. A quick tour.....	6
3.1 The tool-buttons.....	6
3.1.1 The rectangular selection tool.....	7
3.1.2 Measuring and the ImageJ Results Table.....	7
3.1.3 Selection brush and tool options.....	7
3.1.4 Polygon and freehand-selection tools.....	8
3.1.5 Line Selections	8
3.1.6 Profile plots.....	8
3.1.7 Arrow tool, annotations, roi-manager and overlays.....	9
3.1.8 Point Selections.....	10
3.1.9 The magic wand tool.....	11
3.1.10 The particle analyzer.....	12
3.1.11 Magnifying glass and scrolling.....	13
3.1.12 Foreground and Background Color, Clear, Fill, Draw and Revert.....	13
3.2 Macros.....	14
3.2.1 Macro Recorder and batch processing.....	15
3.3 Plugins.....	16
3.4 Image stacks.....	17
3.4.1 The ImageJ 3D Viewer.....	18
4. Help, Documentation and Extensions.....	18
4.1 Help and Documentation.....	18
4.2 Plugins.....	19
4.3 Macros.....	20
4.4 Tools.....	20
4.5 Update Macros and Tools.....	21
5. Display and image enhancements.....	22
5.1 What is a digital image.....	22
5.2 Brightness and Contrast adjustment.....	24
5.3 Non linear display-adjustments.....	26
5.3.1 Gamma correction.....	27
5.3.2 Enhance Contrast.....	29
5.3.2.1 Normalization or contrast stretching.....	29
5.3.2.2 Histogram Equalization.....	31
5.4 Changing the palettes.....	32
5.5 Overlay of multiple channels.....	35
5.5.1 Aligning channels.....	38
5.5.2 Overlay of volume images.....	38
5.6 Noise suppression.....	39



- 5.6.1 Convolution filter.....40
 - 5.6.1.1 Mean filter.....40
 - 5.6.1.2 Gaussian blur filter.....43
 - 5.6.1.3 Edge enhancing filter.....46
- 5.6.2 Rank Filter.....47
- 5.6.3 Filtering in the frequency domain.....48
- 5.7 Background subtraction.....53
 - 5.7.1 Inhomogeneous background.....54
- 5.8 Increasing the apparent sharpness.....56
- 6. Segmentation.....57
 - 6.1 Manual threshold selection.....57
 - 6.2 Measurements and the ROI-Manager.....60
 - 6.3 Computed thresholds.....63
 - 6.4 Segmentation of more then 2 object classes.....63
 - 6.5 Watershed segmentation.....64
- 7. Image types and formats.....66
 - 7.1 Spatial resolution.....66
 - 7.2 Image types.....66
 - 7.2.1 Conversion from 16 to 8 bit.....67
 - 7.2.2 RGB Images.....67
 - 7.2.3 File Formats.....69
- 8. Image Analysis.....69
 - 8.1 Counting objects.....69
 - 8.1.1 Counting objects manually.....69
 - 8.1.2 Counting and measuring separated objects automatically.....70
 - 8.1.3 Counting and measuring objects touching each other automatically.....71
 - 8.1.3.1 Statistical approach.....71
 - 8.1.3.2 Local Threshold.....72
 - 8.1.3.3 Watershed.....72
 - 8.1.3.4 Other methods.....73
 - 8.2 Comparing intensities.....73
 - 8.3 Classifying objects.....74
- 9. Annotating images.....75
 - 9.1 Scale bar, time stamper and event stamper.....75
 - 9.2 Calibration bar.....77
- 10. Working with multi-dimensional data.....78
 - 10.1 Time series.....78
 - 10.2 Measuring velocities.....79
- 11. Biological applications.....80
 - 11.1 Colocalization analysis.....80
 - 11.1.1 Visualizing colocalization using overlays.....80
 - 11.1.2 Quantification of colocalization.....84
 - 11.1.2.1 Intensity correlation coefficient based methods.....84
 - 11.1.2.2 Pearson's coefficient and scatter plot.....85



1. Introduction

In this workshop you will learn how to apply image analysis and processing techniques, using the public domain software ImageJ and some additions (plugins) made by Montpellier RIO Imaging and others. ImageJ has been written and is maintained by Wayne Rasband at the National Institute of Mental Health, Bethesda, Maryland, USA.

ImageJ is the successor of the Macintosh software NIH Image written by Wayne Rasband.

ImageJ is written in Java, which means that it can be run on any system for which a java runtime environment (jre) exists. It can be run under Windows, Mac, Linux and other systems. It can be run as a browser plugin, on a website or as a standalone application.

Because of the easy way in which ImageJ can be extended, using macros and plugins, a lot of functionality is available today, especially in the fields of microscopy and biology.

2. Installing and setting up ImageJ

2.1 Installing ImageJ

The ImageJ homepage is <http://imagej.nih.gov/ij/>. Go to the download page and download the appropriate version for your operating system. In this workshop we will only use windows.

For windows there are three versions available:

- ImageJ bundled with a 32-bit java runtime environment
- ImageJ bundled with a 64-bit java runtime environment
- ImageJ without a java runtime environment

To use the last version, the java runtime environment must already be installed on your computer. For this tutorial we download the ImageJ bundle with the 64-bit java included. Run the installer and follow the instructions on the screen. The first time ImageJ is started, a configuration file for your installation will be created.

The installer will create a quick-start icon and a desktop icon. You can use these to start ImageJ. You can open images by dropping them onto the desktop icon.

2.2 Memory Settings

Java applications will only use the memory allocated to them. Under *Edit>Options>Memory & Threads...* you can configure the memory available to ImageJ.



Figure 1: The memory and threads settings of ImageJ



Image processing and analysis with ImageJ

The maximum memory should be set to $\frac{3}{4}$ of the available memory on your machine. To find out how much memory your machine has, open the properties of *My Computer* and look for the amount of available RAM.

A dialog tells you that the change will be applied the next time you start ImageJ. The configuration is stored in the file *ImageJ.cfg* in the ImageJ folder. Should ImageJ not start after you changed the memory settings, delete the file *ImageJ.cfg*, restart ImageJ and set the maximum memory to a lower value. A double-click on the lower part of the ImageJ launcher window displays the amount of used and available memory. You can use *Plugins>Utilities>Monitor Memory...* to monitor the memory usage.

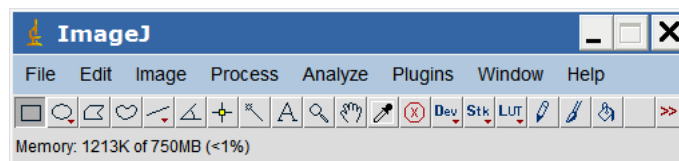


Figure 2: A double click on the window displays memory information.

The number of *parallel threads for stacks* is by default set to the number of processors or cores available in your system.

2.3 Upgrading ImageJ

To upgrade ImageJ, start ImageJ and go to *Help>Update ImageJ...* Select the latest version from the list. ImageJ will be closed. Open it again and look at *help>about ImageJ* to see the current version number.

Alternatively you can download the latest pre-release version of ImageJ from <http://imagej.nih.gov/ij/upgrade/>. Download the file *ij.jar* and replace the version in the ImageJ home folder with the new version.

2.4 Installing loci-bioformats

ImageJ can read a number of image formats like tiff (and tiff stacks), dicom, fits, pgm, jpeg, bmp or gif images. It can not read other formats like ics or lsm by itself. If you want to work with these file formats you need plugins that can handle them.

Loci bio-formats is a project that implements a library which can read many biological image formats. “*Bio-Formats is a standalone Java library for reading and writing life sciences image file formats. It is capable of parsing both pixels and metadata for a large number of formats, as well as writing to several formats.*”. ImageJ can take advantage of the bioformats library. You only have to download the file *loci_tools.jar* from <http://www.loci.wisc.edu/bio-formats/downloads> and to put it into the plugins folder of your ImageJ installation. ImageJ will automatically use it when it can not open an image itself. The easiest way to install the plugin is to drag the link to the jar file directly from the website onto the ImageJ launcher window. This will download the plugin and ask you where under the plugins folder you want to save it. In this case you can save it directly into the plugins folder.

2.5 Associating file types with ImageJ

To open images by double-clicking we have to associate the file type with the ImageJ program. Download the zip-archive of example images ([images.zip](#)) from the mri wiki <http://dev.mri.cnrs.fr/wiki/imagej-workshop> and unzip the archive.



Image processing and analysis with ImageJ

Right-click on a tif-image, select “*open with*” and “*choose the program*” from the context menu. On the dialog click the browse button and select *ImageJ.exe* from the ImageJ home directory. Check the “*always use this program to open files of this type*” option and click *ok* on the dialog. In the same way associate jpg-images with ImageJ. If on recent versions of windows you have problems to associate ImageJ with file-types, download and double-click the file [imagej.reg](http://dev.mri.cnrs.fr/projects/list_files/imagej-workshop) from the resources-folder of the page http://dev.mri.cnrs.fr/projects/list_files/imagej-workshop.

This will add some information about ImageJ to the registry.

If for each image you open a new ImageJ instance is started, go to the menu *Edit>Options>Misc...* and select the “*Run single instance listener*” checkbox. Instead of running a new ImageJ for each image, all images will now be opened in the same Image instance.

3. A quick tour

Before explaining the image analysis tools in ImageJ in more detail, we will make a quick tour to get an overview of the most important tools. We will use the images you downloaded in section 2.5. Open the image *A4 dapi 1.tif*. You can drag an image from your operating system browser to the ImageJ window to open it. We will use this image as an example in the whole chapter 3.

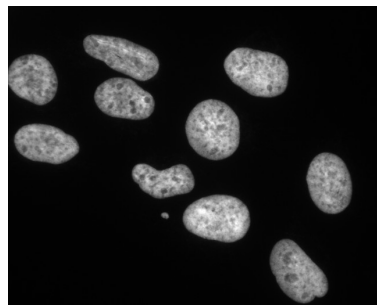


Figure 3: The example image for chapter 3.

Move the mouse-pointer over the image and watch the ImageJ window. The position of the mouse-pointer and the intensity at that position are displayed in the lower part of the window. If it is difficult to see the mouse over the image, you have probably the crosshair cursor activated. You can switch back to a pointer cursor in the menu *Edit>Options>Misc...*

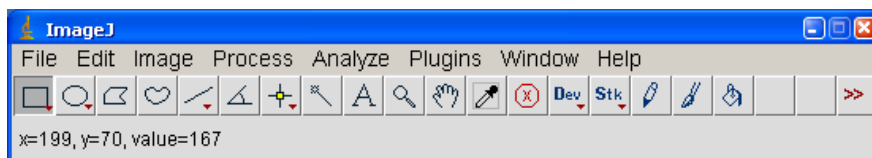


Figure 4: Position and intensity are displayed in the launcher window.

3.1 The tool-buttons

Now look at the tool-buttons in the ImageJ window. The twelve tools upto the *color picker* are basic tools that are always displayed. The remaining tools belong to a toolset that can be changed using the rightmost button. Try this and than switch back to the *startup macros* toolset. When you click on a tool button the button remains pressed and the corresponding

tool will be active until you press another tool button. Notice that the name of the tool is displayed when the mouse pointer is over a tool button.

3.1.1 The rectangular selection tool

Make sure that the rectangular selection tool is active. You can now make a rectangular selection in the image by clicking at a point and dragging the mouse. While doing this the width, the height and the aspect ratio of the selection are displayed. Once you release the mouse button you made a selection on the image. You can still change its position by clicking into it and dragging and its size by using the handles.

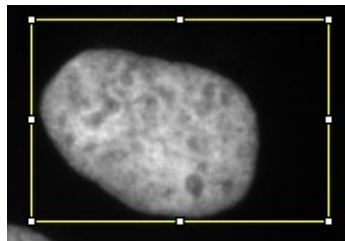
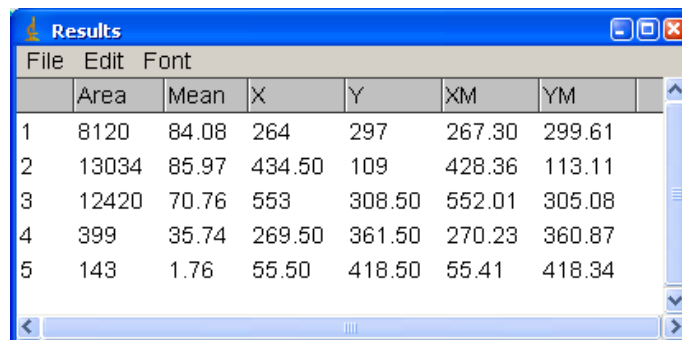


Figure 5: A rectangular selection.

3.1.2 Measuring and the ImageJ Results Table

Try to make a selection that contains exactly one nucleus. Then press the key *m* on the keyboard. This is the shortcut for the *measure* command that can be found in the *Analyze* menu. The ImageJ *results table* will be opened. The table contains the results of the measurements. All measurements will be stored in this global table. In the *File* menu you have the possibility to duplicate the table to keep measurements. From the *Results* menu you can remove everything from the system table with the command *Clear Results*. The *Set Measurements* command allows to define which properties will be measured. Note that the change will only be active for future measurements, the corresponding columns for already existing measurements will be filled with zero values.



	Area	Mean	X	Y	XM	YM
1	8120	84.08	264	297	267.30	299.61
2	13034	85.97	434.50	109	428.36	113.11
3	12420	70.76	553	308.50	552.01	305.08
4	399	35.74	269.50	361.50	270.23	360.87
5	143	1.76	55.50	418.50	55.41	418.34

Figure 6: The ImageJ results table.

3.1.3 Selection brush and tool options

Behind tool buttons that have a small, red triangle in the lower right corner, you can find a list of tools by right-clicking on the button. Remove all selections from the image. You can do this by clicking somewhere in the image while the rectangular selection tool is active or by pressing *shift a* (*a* is the shortcut for *select all*, *shift-a* for *select none* from the menu *Edit-*

>Selection). Right-click on the *elliptical-selection tool button* and select the *selection brush tool*. Click in the image and move the mouse, keeping the button down. You can paint a selection onto the image. Once you released the button, you can add to the selection by holding down the *shift* key while painting. You can remove from a selection by holding down the *alt* key. These two modifier keys work for all kinds of selections in ImageJ. Note that on some systems you have to use *ctrl+alt* instead of *alt*, since the operating system is using the modifier key itself.

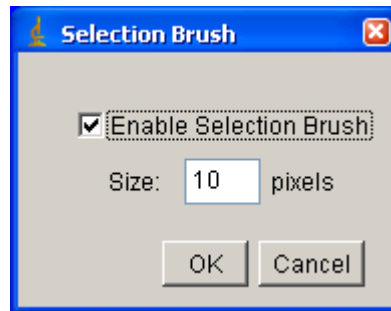


Figure 7: The options of the selection brush tool.

Tools can have options. You can access them by double-clicking on the tool button. Change the size of the brush used by the selection brush tool.

3.1.4 Polygon and freehand-selection tools

Try the polygon and freehand-selection tools. Note that you do not have to close the selections at the end. For the polygon selection you can finish the process by right clicking and the selection will automatically be closed. The freehand selection will be closed as soon as you release the mouse button.

3.1.5 Line Selections

The line selection tools: *straight line*, *segmented line* and *freehand line* can be used to measure lengths. Besides the length the straight line tool measures the angle with the horizontal x-axis. Use a right click to finish a segmented line. The *angle tool* can be used to measure arbitrary angles. The line tools have the line width as a parameter. If the line width is bigger than one, the pixels within the line width will be taken into account for some measurements and operations. A line selection can be transformed into an area selection and vice versa using the commands *Edit>Selection>Line to Area* and *Edit>Selection>Area to Line*.

3.1.6 Profile plots

You can use the line selection tools to create profile plots. First make a line selection across a nucleus, then press the key *k* (the shortcut for *Analyze->Plot Profile*). Note that, when the line width is bigger than one, the profile will be averaged over the width.

If you want averaged line profiles of horizontal or vertical lines, you can use the rectangular selection tool instead. If you want to use vertical lines you have to press *alt-k* instead of *k* in this case.

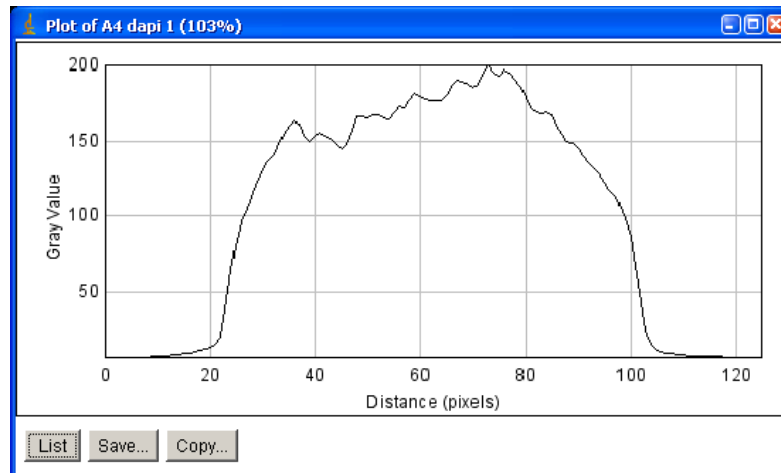


Figure 8: A profile plot along a line crossing a nucleus.

A profile plot can be used to measure properties like, for example, the diameter of an object by taking into account only intensity values above a baseline, that is visible in the plot. Note that the plot is an image itself and that you can make selections on it and measure them. Furthermore the plot-image has a spacial calibration that corresponds to the values displayed on the axis of the diagram. This allows to measure values corresponding to the image from which the plot has been created, independent of the size of the selection that was used to create it.

3.1.7 Arrow tool, annotations, roi-manager and overlays

Select the *arrow tool* from the *straight line* tool button. Double click to open the options dialog and set magenta as foreground color. Click in the image and drag the mouse to create an arrow. With the *shift* modifier key you can restrain the arrows to horizontal and vertical ones.

If you want to have multiple arrows you can add them to the overlay by pressing the *b* key (shortcut for *Image>Overlay>Add Selection*). You can use the *text tool* to add an annotation to an arrow. Use *ctrl+alt+b* to add it to the overlay as well.

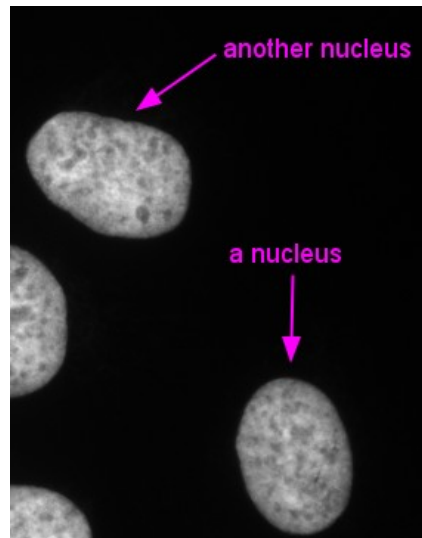


Figure 9: A non-destructive overlay of arrows and annotations.

If you want to be able to modify objects in the overlay, you have to add them to the roi-manager. Roi stands for region of interest and the roi-manager is the tool that allows to manage multiple selections. Use *Image->Overlay->To ROI Manager* to add the objects in the overlay to the roi-manager. Note that you find the roi-manager under *Analyze->Tools*. Make sure that *Show All* and *Edit Mode* are selected. You can now select a selection object by clicking on its number in the image or by selecting it in the list in the roi-manager. You can modify the selected selection object and make the change permanent by pressing *update* on the roi-manager. To create an rgb-snapshot of your image with overlay use the *flatten* command from the roi-manager.

Note that overlay and roi-manager can be used with all kinds of selection, not only arrows and text. Delete all overlays and all selections in the roi-manager. Make some area selections on the image and add each to the roi-manager. Make sure that no roi is selected and press the *measure* button. Each selection will be measured independently.

3.1.8 Point Selections

You can use the point selection tools to manually count objects in an image. With the point selection tool use *shift* to add points. You can delete points using *alt*. Note that the numbers of the remaining points change accordingly. If you make an error and you loose your selection you can get the last selection back using *shift+e* (the shortcut for *Edit->Selection->Restore Selection*). You can add the selection to the roi-manager or measure it. The options allow to directly add each point to the manager or to measure it while you are working. If you use the multi-point selection tool, you do not have to use *shift* to add points, so you are less likely to accidentally delete the selection. As before you can use *flatten* on the roi-manager to create an image that permanently contains the point markers. Using the *more* button on the roi-manager you can save and load selections, in case you need a break in the middle of counting.

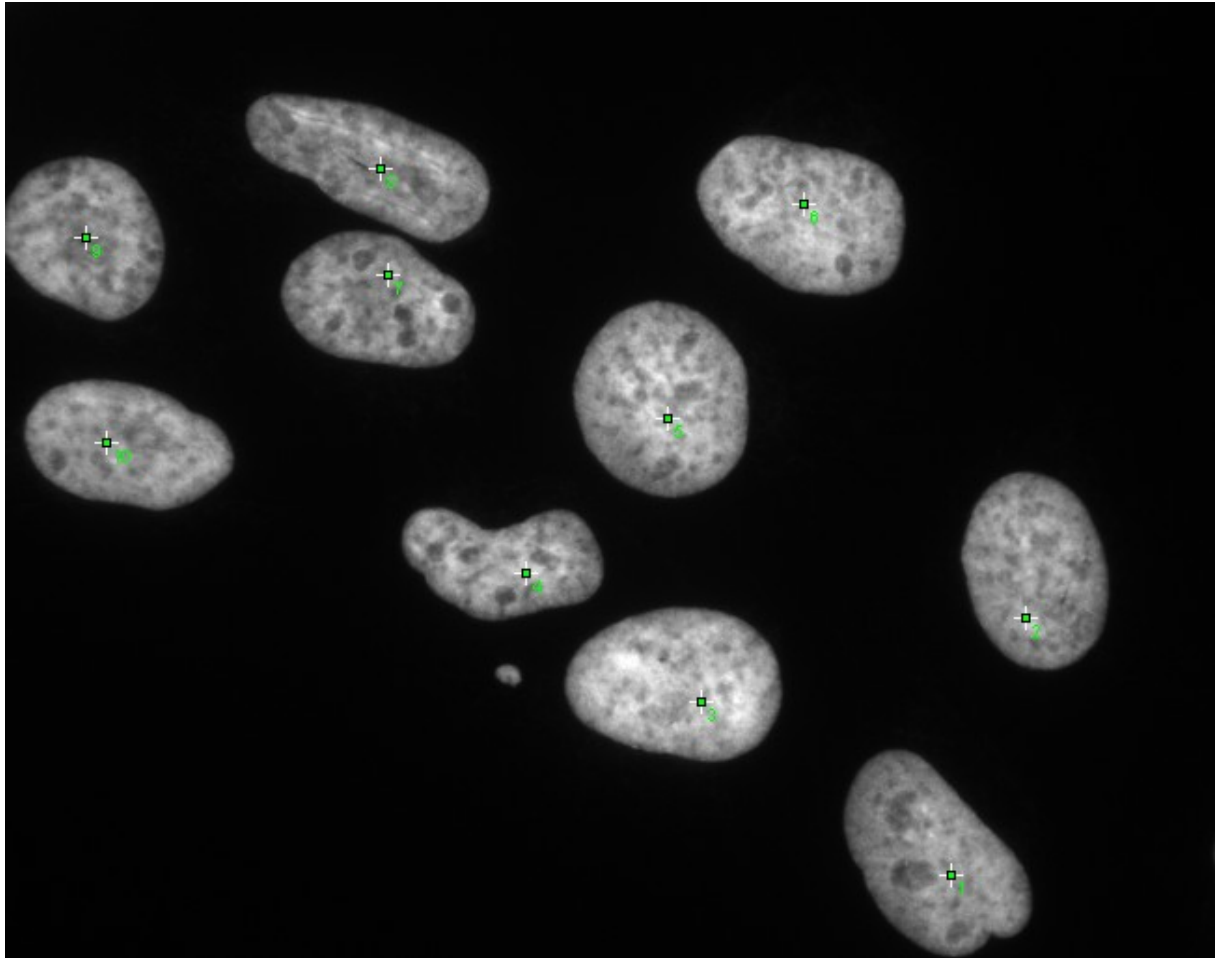


Figure 10: Counting objects with the point selection tool.

3.1.9 The magic wand tool

Use the wand tool to select all nuclei, one after the other. Add them to the roi-manager and measure them. To do this you either need to change the tolerance in the wand tool's options or you need to set a threshold on the image. If a threshold is set on the image the wand tool selects all connected pixel above the lower and below the upper threshold value. To define a threshold value press *shift-t* (the shortcut for *Image->Adjust->Threshold*) to open the threshold tool. Select *Dark background*. If you select the *Over/Under* mode in the right selection box, values below the lower threshold will be displayed in blue and values above the upper threshold will be displayed in green. Adjust the lower threshold and select the nuclei using the wand tool.

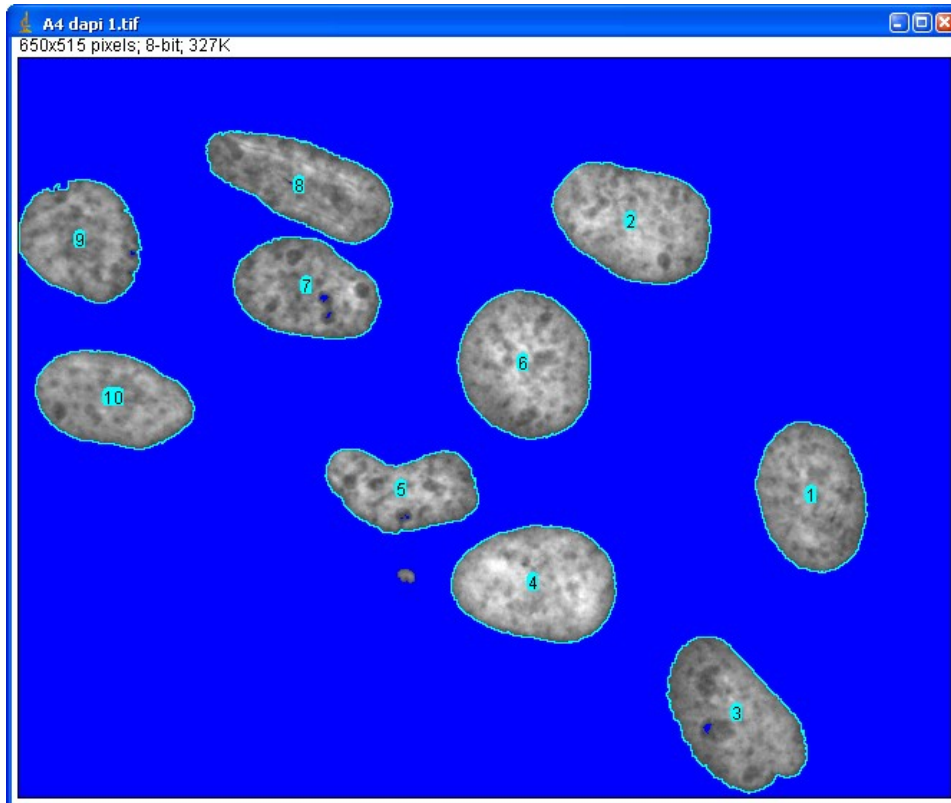


Figure 11: A lower threshold is set and the nuclei have been selected using the wand tool.

3.1.10 The particle analyzer

Instead of clicking on each object with the wand tool, you can let the particle analyzer do the work. As before, define the lower threshold, then call *Analyze Particles* from the menu *Analyze*. Make sure that only *Add to Manager* is selected in the *Analyze Particles* dialog. Then click ok. You can exclude too small or too big objects by supplying a minimal and maximal size in the *Particle Analyzer* dialog.

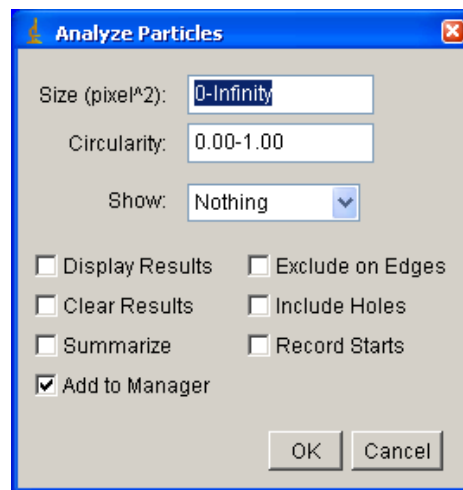


Figure 12: The Analyze Particles dialog.

3.1.11 Magnifying glass and scrolling

The magnifying glass tool allows to change the zoom of the image. A left click zooms in and a right click zooms out. When the image is bigger than the window the position of the current view is indicated in the upper left corner. The scrolling tool allows to move the image in the window by clicking and dragging. The keyboard shortcuts for zooming are + and – and the scrolling can be done by holding down space. Scrolling with the space key down allows to scroll while another tool is active.

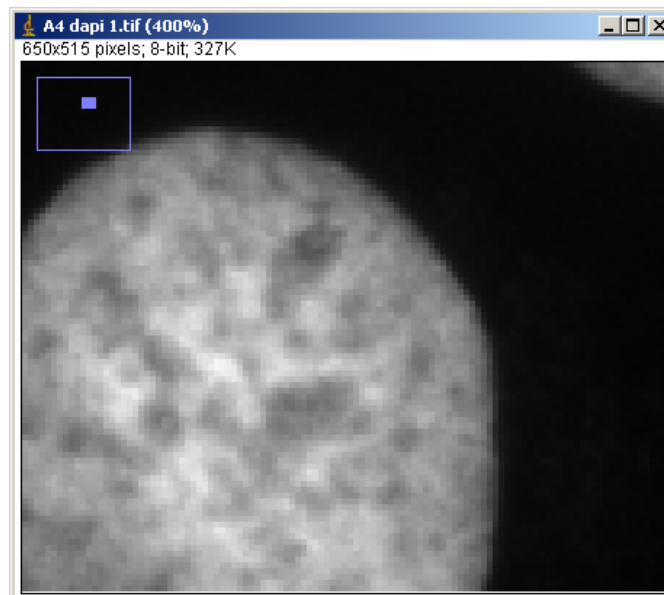


Figure 13: The position of the current view in the image is indicated.

3.1.12 Foreground and Background Color, Clear, Fill, Draw and Revert

The color picker allows to select the foreground color from an image. Select the tool and click somewhere in the image. The foreground color will be changed. In the options of the color



Image processing and analysis with ImageJ

picker the background and foreground colors can be swapped and the colors can be selected from a palette. Note however that only gray-scale values will be used as long as the image is not a color image. You can use *Image>Type>RGB Color* to convert the image into a color image. The foreground color will be used by the commands *Fill* and *Draw* from the menu *Edit* and the background color will be used by the commands *Clear* and *Clear Outside*. After you have changed your image you can reload the original version from disk by pressing the *r* key (shortcut for *File->Revert*).

3.2 Macros

Macros are small programs written either in the ImageJ macro language or in a scripting language, that can be run by ImageJ to automate image analysis tasks. Click on the *dev* menu button on the *Startup Macros* toolset and select *Macros*. This will open a list of macros on the ImageJ website. Search the macro with the name *DisplayTiffInfo.txt*. Drag the link from the web-page and drop it onto the ImageJ launcher window. This will open the macro in the ImageJ macro editor. You can run it using the command *Run Macro* from the menu *Macros* of the macro editor. This macro will display some information stored in the header of the image.

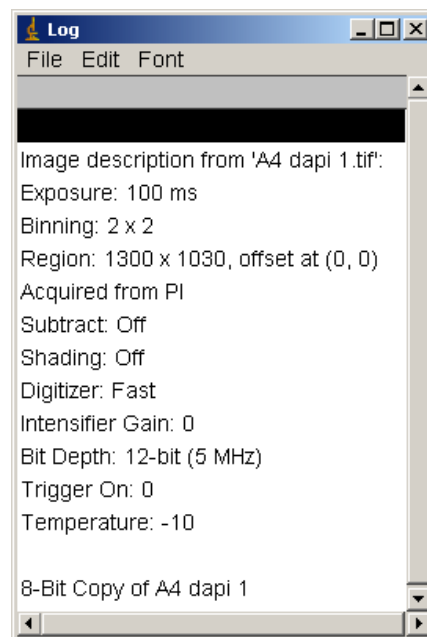


Figure 14: The macro displays information from the header of the image file.

We will try another macro. The macro *ROI_Color_Coder.ijm* can mark objects with a color according to the value of a given property, for example area or roundness. Open the macro by dropping it on the ImageJ launcher window. To use the macro we need the selections of the objects in the roi-manager and the corresponding measurements in the results table. Set the threshold and use the Particle Analyzer and the measure command from the roi-manager to achieve this, then run the macro. If the color code looks strange, make sure that you reset the line width of the line selection tool to 1.

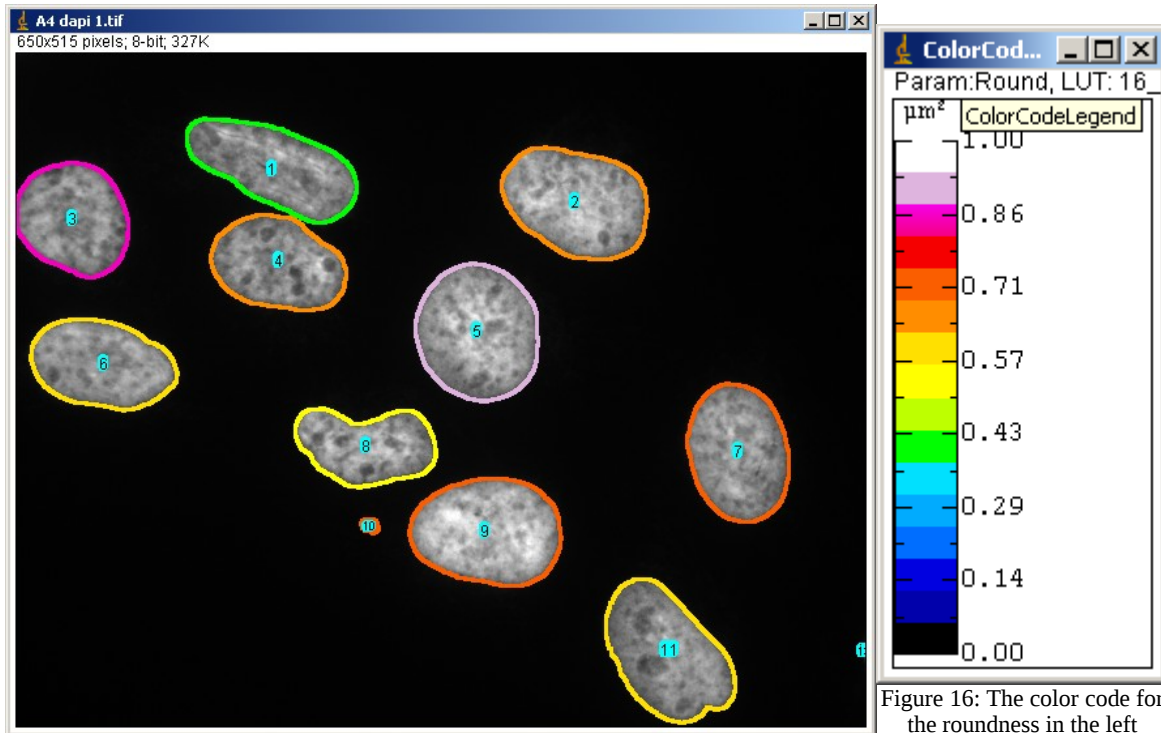


Figure 15: The roundness of the objects is marked with a color-code.

Figure 16: The color code for the roundness in the left image.

3.2.1 Macro Recorder and batch processing

Imagine we want to measure some properties like size and form of the nuclei in our image, but in many images, not only in one. We could run all the commands needed on each image, one after the other to achieve this. But we can do better: We can record the commands needed to analyze one image and create a macro from them.

Start the macro recorder under *Plugins->Macros->Record...* Under *Analyze->Set Measurements* select the properties you want to measure. Make sure that *centroid* is among the selected measurements, it will be needed later for the *label* command. Now set the threshold and run the *particle analyzer*. In the *particle analyzer* select *display results* only. Then run *Analyze->Label* to add labels to the image that indicate which objects have been found. Remove unneeded commands from the macro recorder, then click create and close the recorder window. You should now have a macro similar to the following:

```
run("Set Measurements...", "area mean standard modal min centroid center
perimeter display redirect=None decimal=3");
setAutoThreshold("Default dark");
run("Analyze Particles...", "size=10-Infinity circularity=0.00-1.00
show=Nothing display");
run("Label");
```

Open the first image from the folder *13-batch* and run the macro. Press *shift-o* (the shortcut for *File->Open Next*) to open the next image in the folder and run the macro again. Repeat this until all images are analyzed.

Of course we can still do better. We do not have to load each image manually. Copy the text of the macro, then open *Process->Batch->Macro* and paste it into the text part of the dialog.



Image processing and analysis with ImageJ

Select the input folder. Create a new folder and select it as result folder, then press the *process* button. As a result you will get the measurements in the *results table* and the labeled images in the results folder.

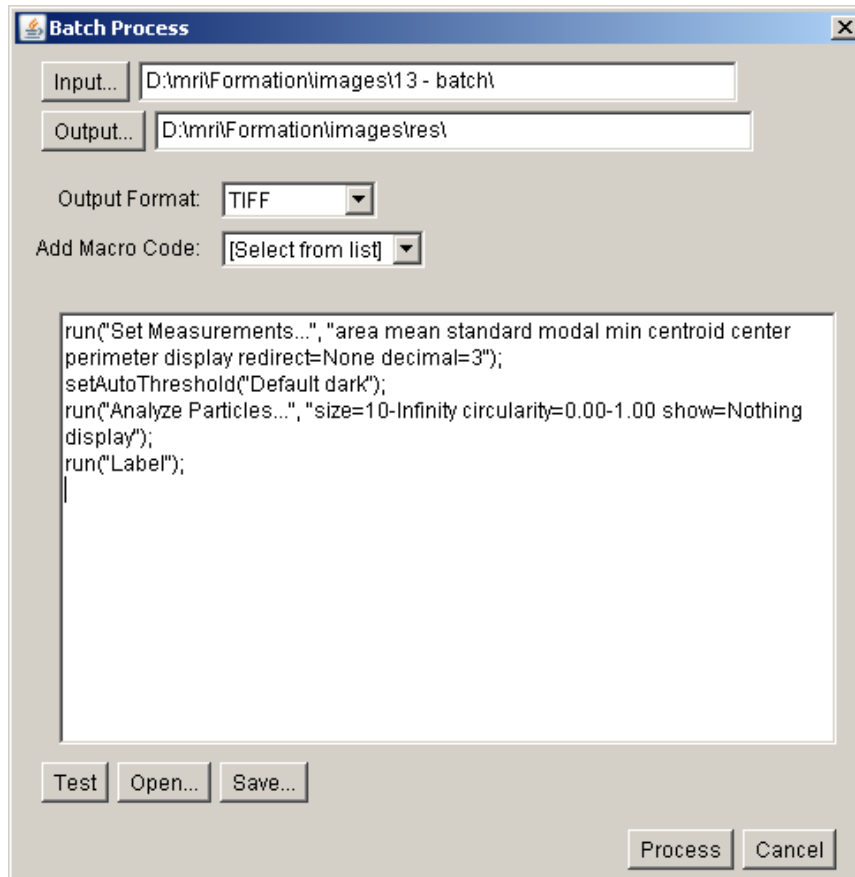


Figure 17: The ImageJ batch processor.

3.3 Plugins

Plugins are java modules that can be used to add functionality to ImageJ. A large number of plugins concerning microscopy is available. Select *Plugins* from the *Dev* menu button. This will open the plugins page on the ImageJ website in your browser. Look for the SIOX (Simple Interactive Object Extraction) plugin and click on the link. This plugin will allow to segment color images by giving examples of background and foreground areas. To install the plugin drag the *siox.jar* link from the webpage and drop it onto the ImageJ launcher window. A file-save dialog will pop-up. Use it to create a subfolder *Segmentation* of the folder *plugins* and save the plugin into this subfolder. It will now be available in the *Plugins* menu in ImageJ.

Open one of the images from the *04 – plant (roboter)* folder. Run the *siox*-plugin from the *plugins* menu. Use a selection tool and select multiple foreground zones holding down the shift key. Then switch to background and select multiple background zones. Press the segment button. If you want you can create a mask for further processing.

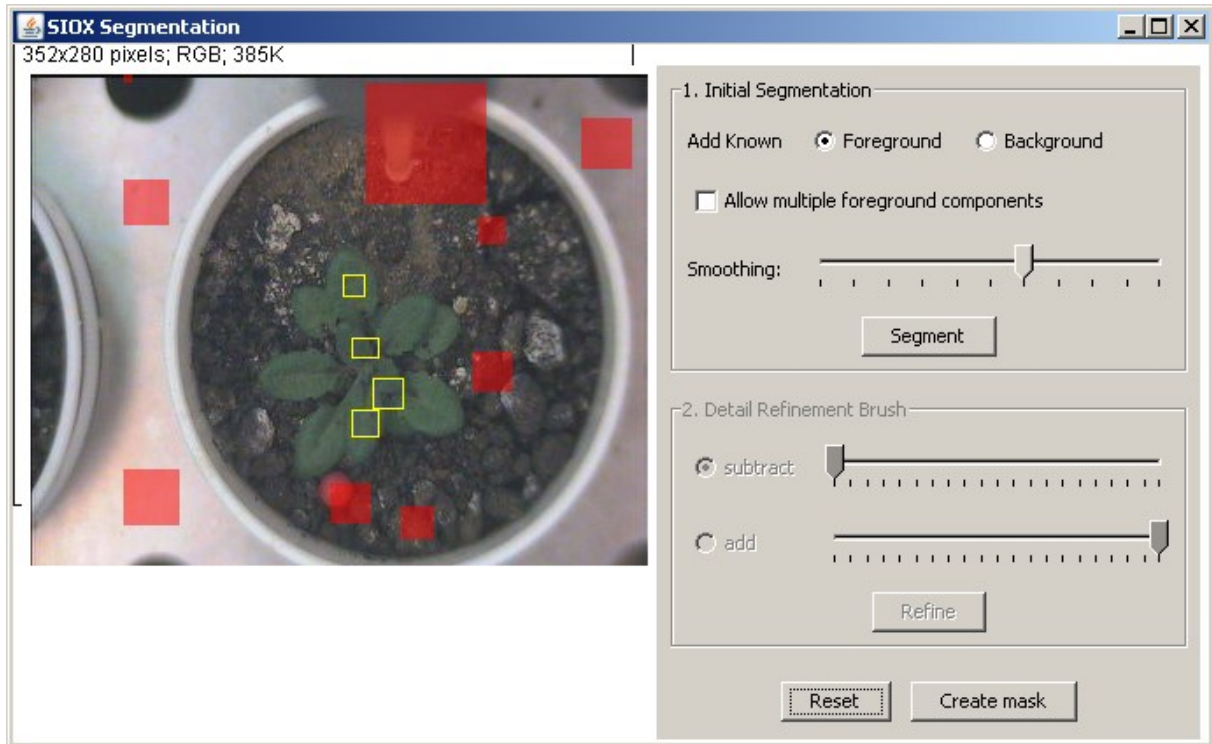


Figure 18: The SIOX Segmentation plugin.

3.4 Image stacks

Stacks can either represent volume data or time series. Open the example stack *T1 Head* from the *Stk* menu button. Note the slider below the image. You can use it to select the visible slice of the stack.



Figure 19: A stack of images in ImageJ

The play and pause button next to the slider allow to start and stop the animation of the stack. A right click on the same button opens the options-dialog for the animation. Try the commands *z project*, *3d-project* and *make montage*. *Combine* allows to combine to stacks one

next to the other or one above the other, while *Concatenate* allows to add the slices of one stack at the end of another.

3.4.1 The ImageJ 3D Viewer

Change the current tool-set to the *Plugins* tool-set. From the *3D* menu button select *ImageJ 3D Viewer*. If you do not have java3D installed, you will be asked if ImageJ should install it for you. If you get this question answer yes. You need to restart ImageJ. Open the *T1 Head* again and start the *ImageJ 3D Viewer* a second time. From the viewer window use the *File->Add Content* menu to add the head to the 3d scene. As long as the scrolling tool is selected you can turn the 3d scene by dragging with the mouse. Make a freehand-selection that covers a part of the head in the 3D-scene. Select the head-object from the *select* menu. Use *Edit->Fill selection* to remove a part of the image data. Reselect the *scrolling* tool and turn the head so that you can see the area that you cut.

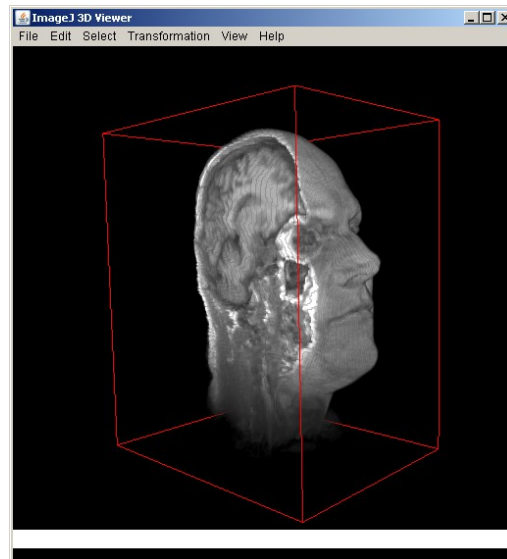


Figure 20: A 3d scene. Part of the data has been cut-off with the fill command.

4. Help, Documentation and Extensions

4.1 Help and Documentation

At the page <http://imagej.nih.gov/ij/docs/> you can find help for the menu commands in the *ImageJ User Guide* as well as a link to the MBF ImageJ manual (*ImageJ for Microscopy*). The page can be accessed from within ImageJ via the menu *Help>Documentation...*

The ImageJ user guide written by Tiago Ferreira and Wayne Rasband can be found under: <http://imagej.nih.gov/ij/docs/user-guide.pdf>

Further information can be found at the ImageJ Documentation Wiki: <http://imagejdocu.tudor.lu/doku.php>



Image processing and analysis with ImageJ

The new imagejdev.org (<http://imagejdev.org/>) is supposed to become the new home of the ImageJ community in the future.

The book “Digital Image Processing – An Algorithmic Introduction using Java” can be found here: <http://www.imagingbook.com/index.php?id=15>

Wilhelm BURGER • Mark J. BURGE

Digital Image Processing – An Algorithmic Introduction using Java

Textbook with 560 pages, 271 figures and 17 tables

© Springer 2008

ISBN: 978-1-84628-379-6

The McMaster Biophotonics Facility offers an ImageJ manual and a collection of plugins under: <http://www.macbiophotonics.ca/imagej/>

You can find more detailed explanation of some image processing techniques in the hyperlink image processing resource (hipr) at:

<http://homepages.inf.ed.ac.uk/rbf/HIPR2/index.htm>

4.2 Plugins

New functions can be added to ImageJ by installing plugins. Plugins either add commands to some menus or they can be run from the *Plugins* menu and its sub-menus once they have been installed.

You can access the ImageJ plugin-site from the menu *Help>Plugins....* If the plugin is available as a .jar or a .class file, you can install it by dragging the link directly from your web-browser onto the ImageJ launcher window. This will cause ImageJ to download the plugin. It then opens a dialog that lets you copy the plugin into the plugins folder or into one of its sub-folders.

As an example we will install a plugin that draws random ovals into a new image. Use the help menu to go to the plugin site. Scroll down to the category *Graphics* and click on the link *Random Ovals*. Drag the link *Random_Ovals.class* onto the ImageJ launcher window. Select the folder *Graphics* in the *Save Plugin...* dialog or create it if it doesn't exist. You can now directly run the newly installed plugin from the *Plugins* menu, using *Plugins>Graphics>Random Ovals*.

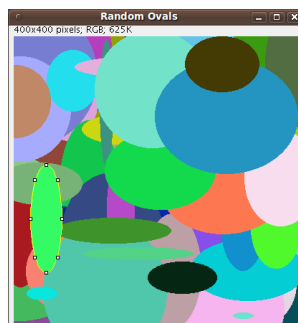


Figure 21: The result of the « Random Ovals » plugin.

Should the plugin come as a zip file, unzip it into a temporary folder. If the content consists of .jar or .class files, you can drag them to the ImageJ launcher window in the same way to

install them. Otherwise follow the instructions on the web-site from which you downloaded the plugin.

4.3 Macros

Macros are scripts in the ImageJ-macro language. A number of macros is available in the folder *macros* in the ImageJ base-folder. You can open them by dragging them onto the ImageJ launcher window or by using *Plugins>Macros>Edit...*

You can access the ImageJ macro site from the menu *Help>Macros*.

You can directly drag a link to a macro from the web-browser onto the ImageJ launcher window. This will open the macro in the ImageJ macro editor, from which you can run the macro using the menu *Macros>Run Macro* (or *ctrl+r*). Try to run the macro *Mandelbrot.txt* this way.

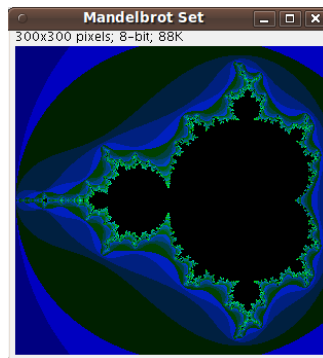


Figure 22: The result of the Mandelbrot.txt macro.

4.4 Tools

Tools are macros that add a tool button to ImageJ's toolbar. Note that the name of the tool is displayed in the status line, when you move the mouse over a tool button. There is a difference between action tools and tools. Action tools execute a single command. They do not change the active tool. When you press the button of a tool, that is not an action tool, you change the active tool. The active tool will usually do something when you click in an image. An example of a tool is the *Rectangular Selection* tool. An example of an action tool is the *Abort Macro or Plugin* tool.

There can be multiple tools hidden behind one tool-button. If this is the case a small red arrow is displayed on the button. A right-click on the button shows the list of tools and allows to change the tool. Look for example at the second button from the left. This is the *elliptical selection* tool. A right-click on the button gives access to the *selection brush* tool

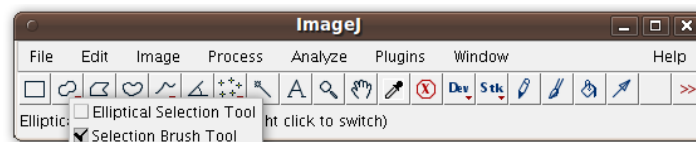


Figure 23: A right click allows to select between different tools.

A tool or a group of tools (those that share the same button) can have options. The options can be accessed by a double-click on the tool button.

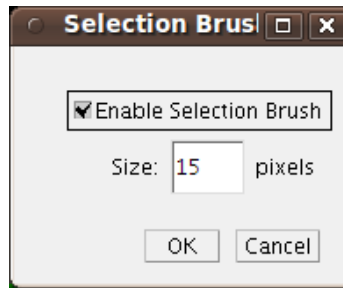


Figure 24: The options of the selection brush tool.

From the menu *Help* select *Macros*. On the web-page scroll down to the *tools* folder and enter the folder. Search for *RGBProfilesTool.txt* and drag the link onto the ImageJ toolbar. All tools right of the *color picker* will disappear and there will be a new tool button for the *rgb profiles tool*.



Figure 25: An rgb example image.

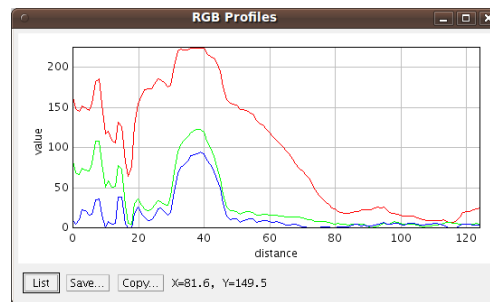


Figure 26: The rgb profile along the line in the image.

4.5 Update Macros and Tools

There is a tool available that automatically downloads and updates the macros and tools from the ImageJ website to your local machine. Select the toolset *Luts Macros and Tools Updater* from the toolset button (>>). Read the message on the dialog and close it by pressing the *ok* button. Now click on the leftmost menu-button of the toolset and select *Macro, Tools and Toolsets Updater*>. Then answer the questions in the dialog.

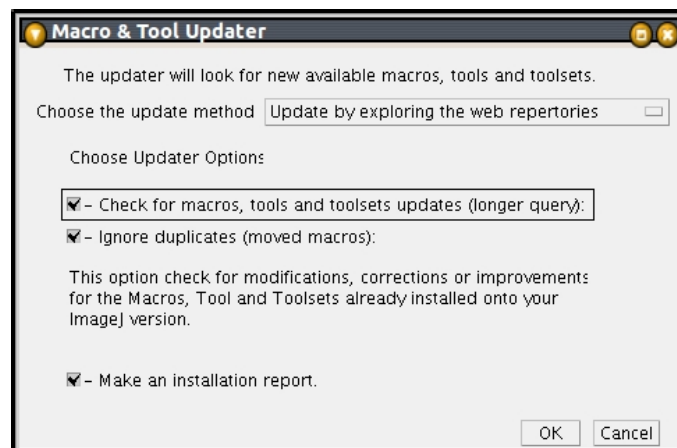


Figure 27: The macro and tool updater.

New macros, tools and toolsets will be downloaded into the *ImageJ/macros*, the *ImageJ/macros/tools* and the *ImageJ/macros/toolsets* folders. Existing macros, tools and toolsets will be updated if a newer version is available on the website.

5. Display and image enhancements

5.1 What is a digital image

Before we start to work on images we should answer a simple question: “What is a digital image?” There are different kinds of images, of course. Images can for example be two or three dimensional and they can have one or more color channels.

An image is a rectilinear array of point samples (pixels). In the simplest case the array has two dimensions and width \times height pixels. The distance between the point samples is sometimes called the pixel size. The pixel consists of the coordinates in the image and an associated sample value. The sample value can be an integer between 0 and a maximum value. The sample values can be interpreted as intensities. 0 usually means no intensity, yielding a black pixel. Higher values usually mean lighter pixels and lower values mean darker pixels.

The word pixel is build from “pix element”. Pix is an abbreviation for picture.

When a digital image is displayed on a device the elements of the image must be mapped to the elements of the output device. This can be done in different ways. One way is to let each output element display the sample value of the nearest existing image sample. This results in pixels being displayed as little squares. Better results can be achieved by using interpolated values for the output elements.

Open the image *example1.tif*. You can either double click the image or use *File>open* from the ImageJ window. When an ImageJ window is active you can use keyboard shortcuts to call commands from the ImageJ menus. To get the open dialog, click on an ImageJ window and press ‘o’.

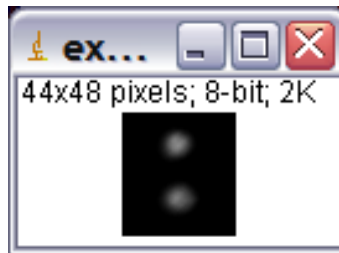


Figure 28: The first example image.

Use the magnifying glass tool from the tool-box and click multiple times on the image to get the maximum zoom. You can now see the individual pixels being displayed as little squares. Select *Interpolate Zoomed Images* in the menu *Edit>Options>Appearance* to switch to an interpolated display.

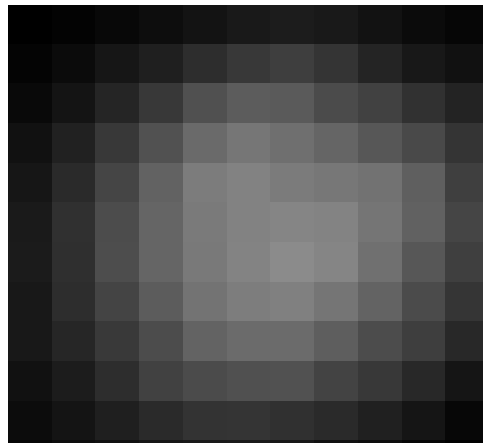


Figure 29: With a higher zoom, individual pixels become visible as little squares.

The coordinates and the intensity value of the pixel under the mouse pointer are displayed at the bottom of the ImageJ window. You can use the right mouse button to zoom out again. To scroll the image hold down space-bar, press the left mouse button and move the mouse. Remark that the origin of the coordinate system is in the upper left corner of the image with increasing coordinates from left to right and top to bottom. Save the image as a text image to the desktop and open it with a text editor or drop it on the ImageJ window. You see the image as a matrix of intensity values.

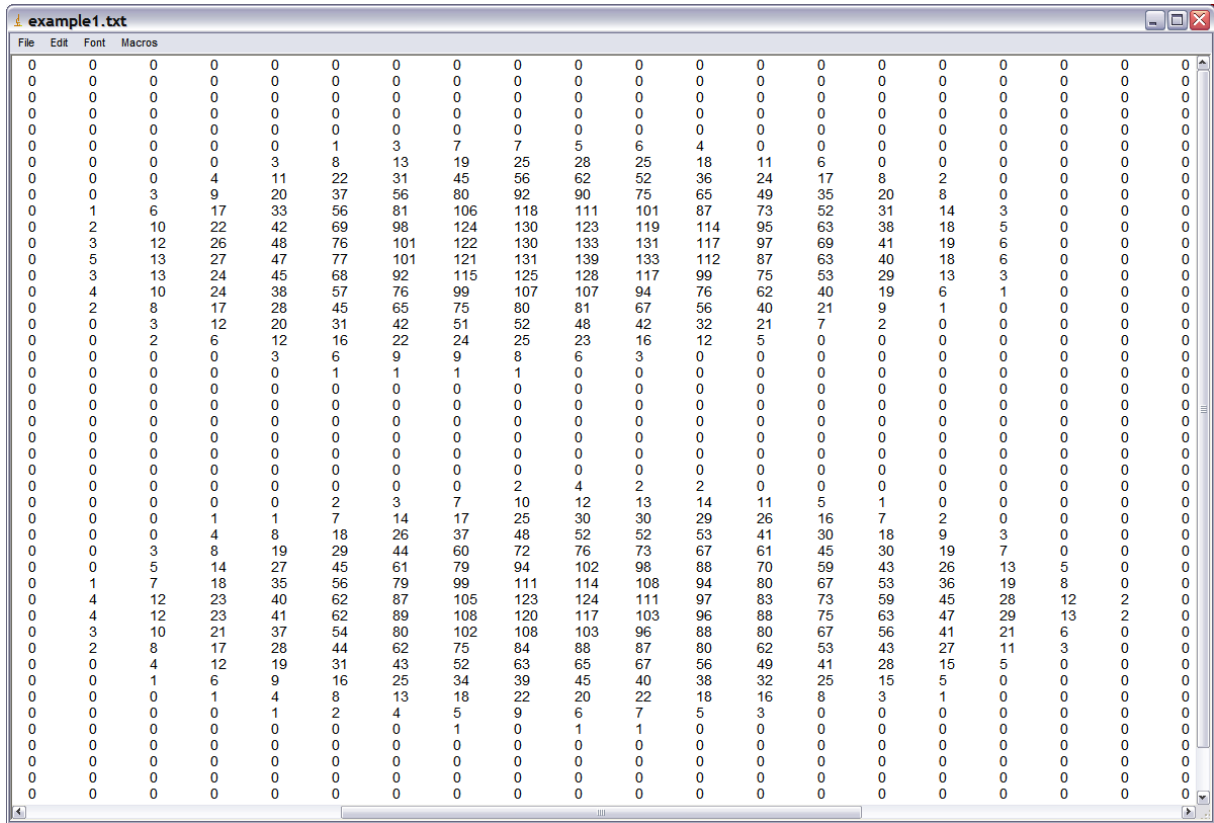


Figure 30: The image as a matrix of intensity values.

5.2 Brightness and Contrast adjustment

Open the image *bc-adjust.tif*. The image is dark and the contrast is very low. Open the Brightness and Contrast Adjuster from the menu *Image>Adjust>Brightness/Contrast* or press *Shift+C* or run the B/C-Adjuster from the Lookup Tables tool set.

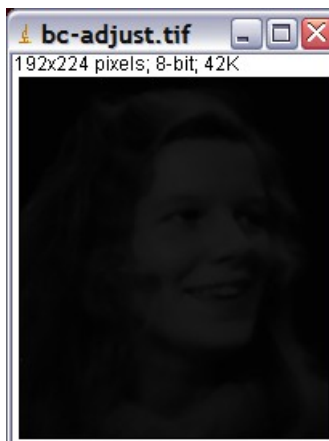


Figure 31: The intensities are too low to distinguish details in the image.

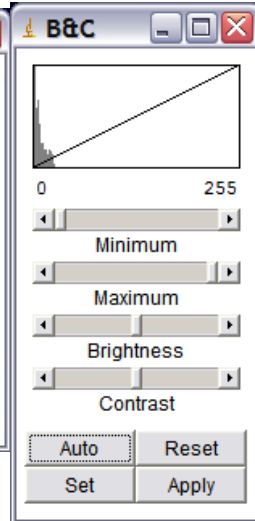


Figure 32: The brightness and contrast adjuster.



Figure 33: The image after the auto-adjustment.

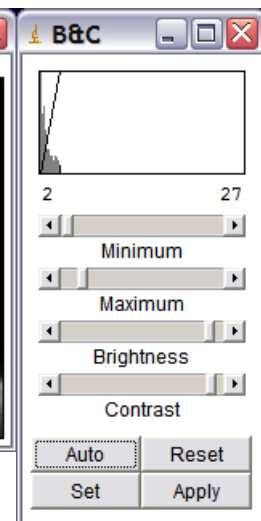


Figure 34: After the adjustment intensities below 3 will be displayed as black and intensities above 26 as the 255.

Press the *Auto* button to make an automatic adjustment. With *Reset* you can go back to the original display. Use the four sliders to adjust brightness and contrast manually. Remark that only the displayed intensities are changed, the pixel values remain unchanged.

Press the measure button on the toolbox or select the image and press *Ctrl+m*, then change the brightness/contrast and measure again. Compare the values *mean* and *IntDen*, which give the average and total intensity in the image. If you can't find *mean* or *IntDen* in the results table, open the dialog *Analyze>Set Measurements...* and check *Mean Gray Value* and *Integrated Density*.

	Area	Mean	Min	Max	IntDen
1	43008	9.296	2	31	399804
2	43008	9.296	2	31	399804

Figure 35: The ImageJ results table.

How does the adjustment work? When the image is displayed two points are stored together with the image. The point $(min, 0)$ and the point $(max, maxIntensity)$. All intensity values below min are displayed with the intensity zero, all intensity values above max will be displayed with the intensity value $maxIntensity$. Values between min and max are mapped by the line equation that is defined by the two points $(min,0)$ and $(max, maxIntensity)$.

In the beginning when we opened the example image, min was 0 and max was 255 and the $maxIntensity$ is 255. Since this defines a line with the gradient 1, each intensity value is mapped to itself. After we applied the auto adjustment, min becomes 2 and max 27, so pixels that have the intensity 27 are displayed with the value 255, pixels that have the intensity 20 with the value 184, and so on.

Now look at the image in the upper part of the B&C window. What you see there is the histogram of the image. On the x-axis are the intensity values from 0 to 255. On the y-axis you find the count of pixels in the image that have the intensity x . We'll come back to the histogram later when we talk about thresholding. For the brightness and contrast adjustment it shows you how many pixels you set to the max intensity and to zero. On the left you find the *min* display value and on the right the *max* display value. The line that is drawn in the histogram shows the mapping of the values between *min* and *max*.

If you move the brightness slider to the right the whole line will move to the left and lower intensity values will be mapped to higher display values, making the display brighter.

If you move the contrast slider to the right the gradient of the line increases so the intensity values closer together will be mapped to display values with a bigger distance.

You can use the set button to enter the min and max value directly. If you press apply, the pixel intensity values in the image will be changed to the values displayed.

The auto adjustment will look at the histogram of the image and adjust the min and max values in a way that a small percentage of the pixels become zero and *maxIntensity*.

Sometimes this doesn't yield the desired results, especially when the object is very small compared to the background. In this case you can make a selection on the image and the auto adjuster will use the histogram of the selection to compute the *min* and *max* values that are then applied to the whole image.

Use the rectangular selection tool from the toolbox to make a selection on the image and press the auto button. Try this for different regions in the image. Another way to adjust the display is the *Window&Level adjuster*. In this case you choose a middle value (the level) and a range around the middle value (the window). If you choose for example *level* 15 and *window* 24 you get a *min* value of 3 and a *max* of 27.

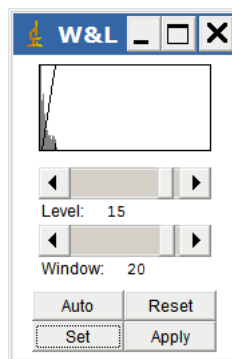


Figure 36: The window/level tool.

5.3 Non linear display-adjustments

In the brightness and contrast adjustments in the last chapter, intensity values are mapped to display values by a linear function, meaning small intensities are changed by the same factor as big values. In cases where very high and very low intensities should become visible a non-linear display adjustment is needed, so that small values can become bigger without saturating already big values.

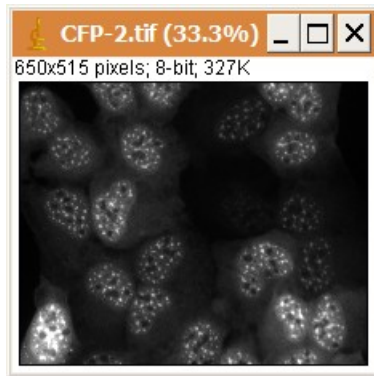


Figure 37: The low intensities are invisible in the original image.

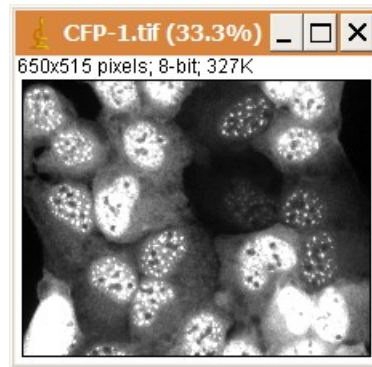


Figure 38: A linear display adjustment makes low intensities visible but saturates high intensities.

5.3.1 Gamma correction

Instead of a linear function we will map intensities to display values using a function of the form $V_{out} = V_{in}^\gamma$.

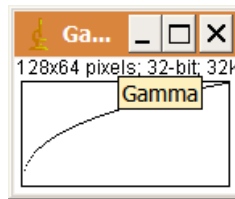


Figure 39: gamma = 0.4

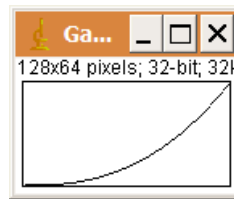


Figure 40: gamma = 2.5

Open the image *cells.tif* or *cells2.tif*. Use *Process>Math>Gamma* to adjust the image intensities using a gamma-function. You can select the *Preview* checkbox and try different values. When you click *ok* the gamma function is applied to the image, i.e. the pixel values are modified accordingly.

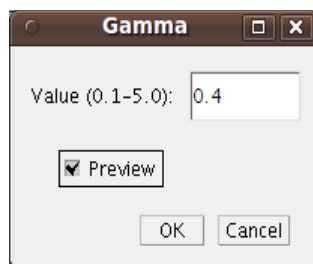


Figure 41: The dialog for applying a gamma-correction.

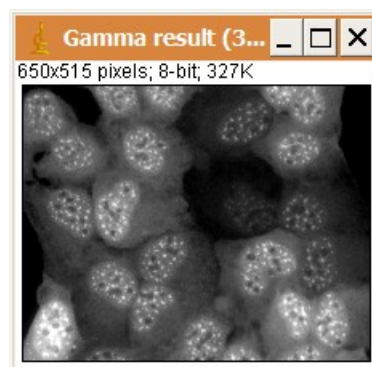


Figure 42: The gamma-correction has been applied to the image.

The exact function used by this command is:

$$f(i) = e^{\log\left(\frac{i}{255}\right) \cdot \gamma} \cdot 255$$



Image processing and analysis with ImageJ

This command in the menu *Math* is there to modify the pixel values of the image not to make a display adjustment.

To adjust the display without affecting the pixel values the macro “GammaCorrectionTool” can be used. ImageJ provides a convenient way to run macros and plugins from a toolset. We will install the tool on a new toolset, so that it can be run by pressing a tool button. Look at the rightmost button on the ImageJ launcher window. Clicking on it allows to change the current tool-set.

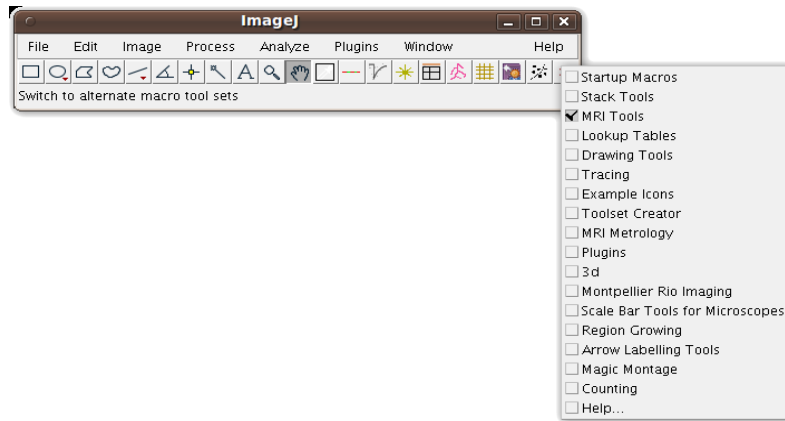


Figure 43: The button >> on the ImageJ launcher displays the list of available toolsets

First we need to get the macro from the ImageJ website. Use *Help>Dev. Resources...* to open the developer resources page on the ImageJ website. Click on *Macro Tools*. Scroll down to *GammaCorrectionTool.txt* and click on the link. You should now see the macro code in your web-browser (Don't worry you don't need to understand the code, at least not right now). Select the macro text (*ctrl+a*) and copy it into the system clipboard (*ctrl+c*). Now go back to ImageJ. From the toolset list (click on the rightmost button to get it) select *Toolset Creator*.

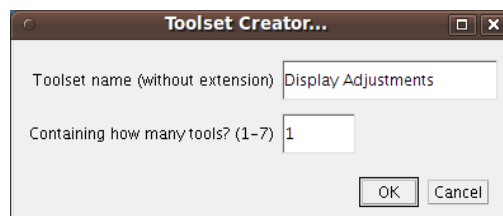


Figure 44: Creating a new toolset.

Enter *Display Adjustments* as the name of the new toolset and enter 1 as the number of tools. The *Toolset Creator* is normally there to create toolsets from the plugins in the plugins menu. We will just use it to create a new toolset and we will overwrite the created content. Normally you would now enter information for each tool on the toolset. You can ignore this and press ok. Close the log window. We will now open the code of the newly created toolset. Hold down the shift-key, display the toolset-list and select *Display Adjustments*. The code of the toolset is now displayed in a text window. Replace the content with the content of the macro. To do so click in the text window, press *ctrl+a* to select all and *ctrl+v* to paste the code of the macro that we copied into the clipboard before. Save the changed file using the *File* menu from the text window (or *ctrl+s*).

To update the display, reselect the *Display Adjustments* toolset again. You should see a button with the Greek letter gamma. Open the image *cells.tif* or *cells2.tif* again or use *File>Revert* to

reload it. Now select the gamma-correction tool by pressing the button with the letter gamma and click in the image. The gamma function is displayed on the image and you can change the value of gamma by moving the mouse. When you have found a good value use ctrl+shift+a to get rid of the function display in the image. Be careful to not click into the image again, before you selected another tool. In contrast to the command above this macro only changes the display, not the pixel values.

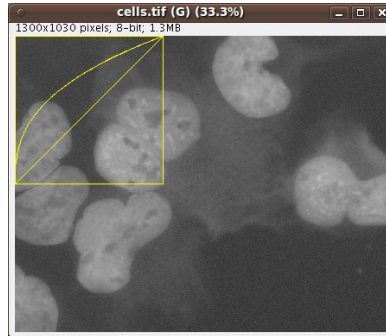


Figure 45: The gamma function is displayed on the image.

You might have noticed that using the same gamma value with the `Processing>Math>Gamma` command and with the `GammaCorrectionTool` macro yields different results. This is because the later uses a different gamma function:

$$f(i) = \left(\frac{i}{255}\right)^{\frac{1}{\gamma}} \cdot 255$$

5.3.2 Enhance Contrast

The enhance contrast tool, that can be found under `Process>Enhance Contrast`, has three different functions. When neither `Normalize` nor `Equalize Histogram` is selected, the display is adjusted in a way, that the given number of pixels becomes saturated. In this case the intensity values in the image are not changed. When `Normalize` is selected, a contrast stretching or normalization is done. The third function is the histogram equalization.

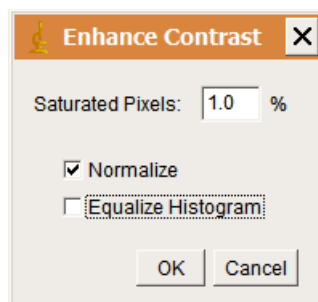


Figure 46: The enhance contrast dialog.

5.3.2.1 Normalization or contrast stretching

The normalization stretches the range of intensity values, so that the given percentage of pixels becomes saturated. After a normalization the min and max values for the given image

type will be present in the image. If for example the minimum and maximum value in an image are 10 and 20 the values can be mapped to use the whole range between 0 and 255 by:

$$i' = (i - \text{image_min} / (\text{image_max} - \text{image_min})) * \text{range_max}$$

$$i'(10) = (10 - 10) / (20 - 10) * 255 = 0$$

$$i'(20) = (20 - 10) / (20 - 10) * 255 = 255$$

$$i'(15) = (15 - 10) / (20 - 10) * 255 = 127.5$$

However this would be very sensible to outliers. A better way is to determine *image_min'* and *image_max'* so that a given percentage of pixels is below *image_min'* and above *image_max'*.

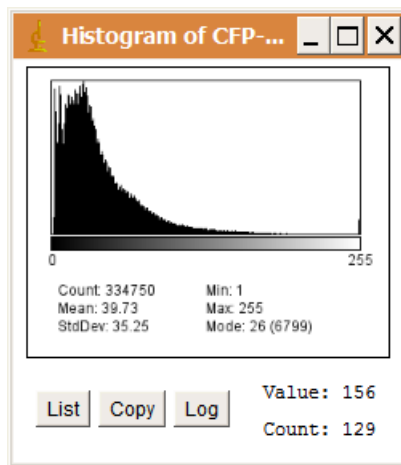


Figure 47: Histogram before normalization.

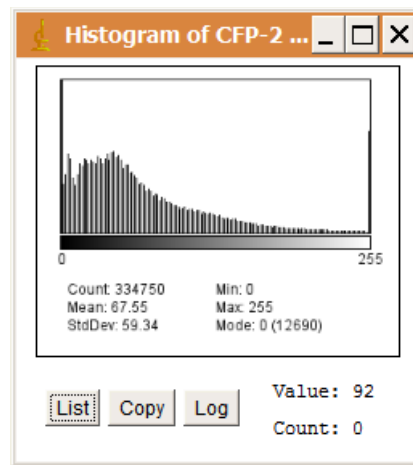


Figure 48: Histogram after normalization.

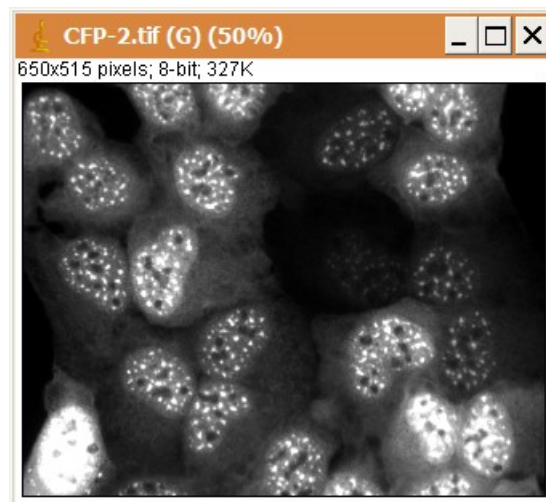


Figure 49: The image after histogram normalization.

Using the enhance contrast tool with the normalize option has the same effect as using it without the normalize option and then pressing apply on the brightness and contrast adjuster.

5.3.2.2 Histogram Equalization

In the histogram equalization each intensity value i is replaced with the the sum of the histogram values for all intensities up to i (including $h(i)$ itself). The result is then normalized to the *min* and *max* intensities of the image type. If you check equalize histogram in the enhance contrast dialog, the other input in the dialog is ignored. If you hold down *alt*, the classical histogram equalization will be executed. Otherwise a modified version, using the square root of the histogram values will be used. If a selection exists on the image, the calculation will be based on the histogram of the selection only.

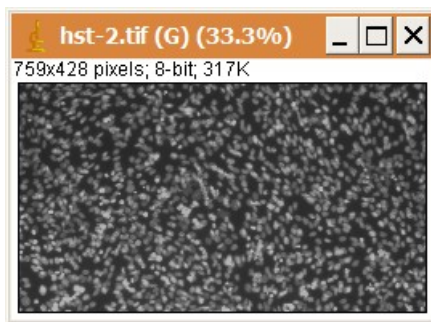


Figure 50: The input image.

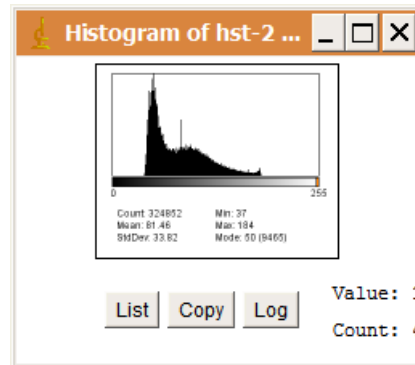


Figure 51: The histogram of the input image.

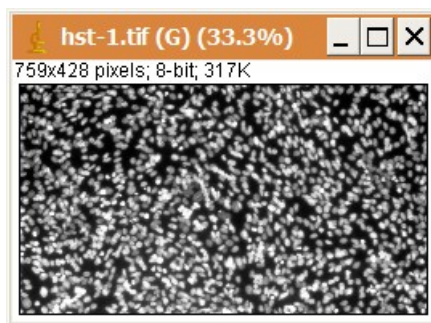


Figure 52: The image after histogram equalization in the square root version.

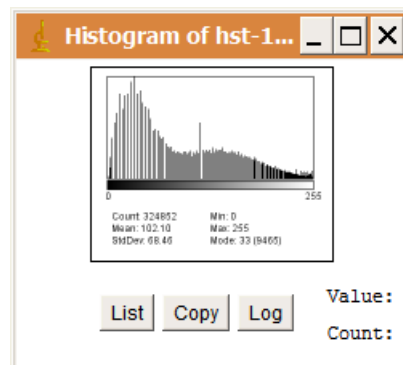


Figure 53: The histogram of the image after histogram equalization in the square root version.

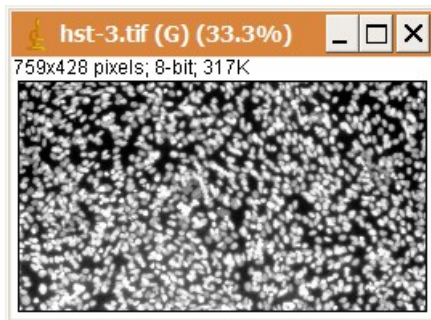


Figure 54: The image after histogram equalization in the standard version (alt key down).

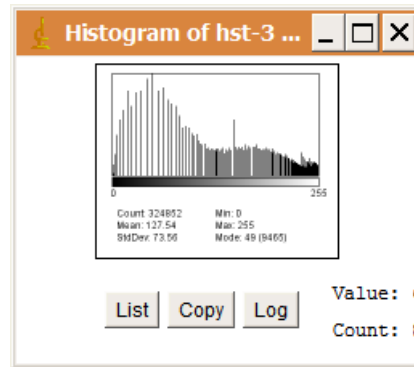


Figure 55: The histogram of the image after histogram equalization in the standard version (alt key down).

Histogram equalization augments the local contrast and is most useful when objects and background contain high and low intensities, as for example in brightfield microscopy. Compare the results of a linear adjustment a gamma adjustment and the histogram equalization on the image *cells2.tif*.

Try the histogram equalization on the image *roots.tif*.

5.4 Changing the palettes

We will now change the display of our image by mapping the intensity values to colors. This is done by the use of lookup tables. A lookup table is a table that maps the 255 intensity values to 255 arbitrary colors. Since colors are expressed by the proportion of the three basic colors red, green and blue, an entry that would map the intensity 255 to white would for example be:

255 -> 255, 255, 255

To map the intensity 0 to red the look-up-table entry would be:

0 -> 255, 0, 0

Select the *Toolbar Menus* tool-set. Click on the *LUT* button and open the *Rainbow RGB* lut.

You see an image that shows the mapping of the intensity values to the colors. Each intensity value is represented by a column with the width of 1 pixel, painted in the corresponding color.

Can you tell in which color the intensity value 200 is displayed?

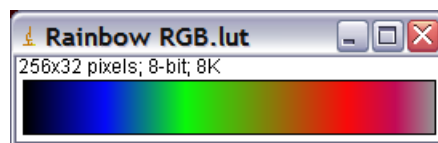


Figure 56: A graphical representation of the lookup-table.

Go to the menu *image>color>show lut*. Above the color table you see how each color of the lut is mixed from the red, green and blue components.

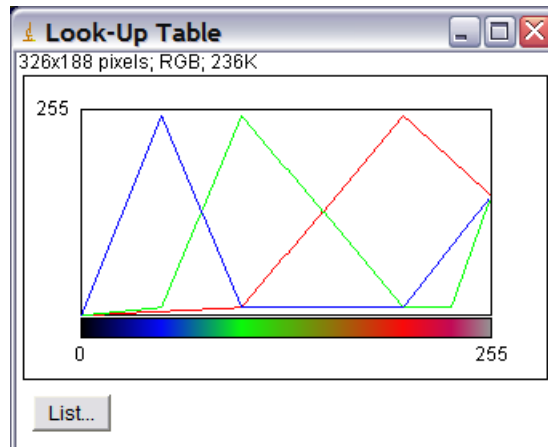


Figure 57: The red, green and blue components for each index of the lookup-table.

If you click on the list button, you get the lookup-table in numerical form. What are the RGB-components of the intensity 100?

Index	Red	Green	Blue
98	10	239	20
99	10	244	15
100	10	249	10
101	12	247	10

Figure 58: Part of the lookup-table.

Open the image *cells.tif* and click on *Rainbow RGB* lut again. When images are opened the lookup-table will be applied to the active image. Try different lookup tables.

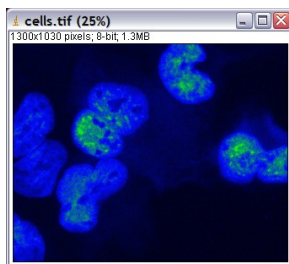


Figure 59: The rainbow RGB lookup-table applied to an image.

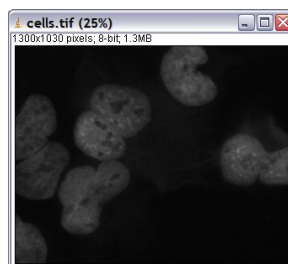


Figure 60: The grey lookup-table applied to an image.

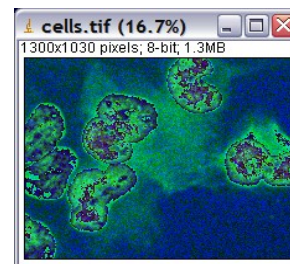


Figure 61: The 3-3-2 RGB lookup table applied to an image.

The lookup table *HiLo* is useful to adjust the display of our image. 0 will be displayed in blue, 255 in red and values in between will be displayed in gray. Can you describe how this lookup table looks like?

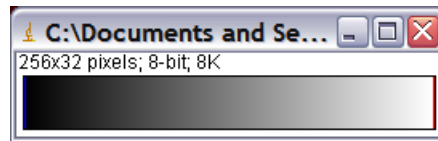


Figure 62: The hilo lookup-table.

Apply it to the image and adjust the brightness and contrast in a way that most of the background is zero and only a small portion of pixels becomes saturated.

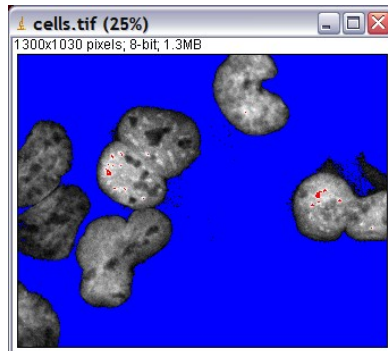


Figure 63: Use the hilo lut to adjust brightness and contrast.

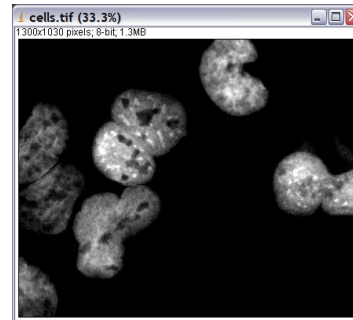


Figure 64: The brightness and contrast adjustment has been applied and the lut has been changed back to grays.

Go to *Image>Color>Edit LUT...* or press the *Edit LUT* button from the *Lookup Tables* tool-set. Change the lookup table in a way that 0 will be displayed in green, 255 in yellow and values between 100 and 120 in different shades of red, becoming lighter with higher intensity. Save the lookup table under a new name, apply it to the image and adjust the *B&C* again.

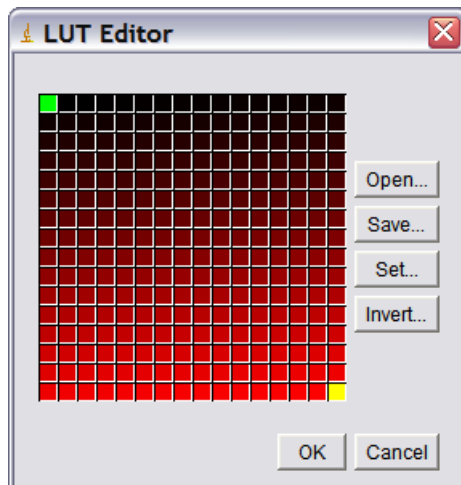


Figure 65: The lut editor.

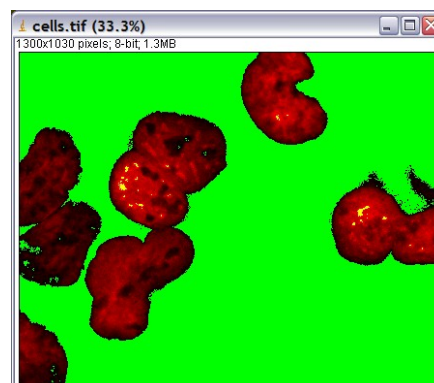


Figure 66: The newly created lut applied to an image.

Another convenient way to access all lookup tables is to tear off the *Image>Lookup Tables* menu by using *Plugins>Utilities>Control Panel*.

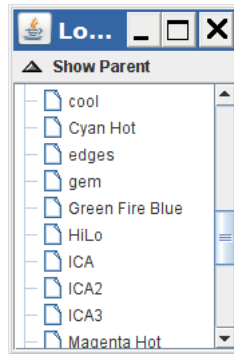


Figure 67: Menu can be kept on the screen using the Control Panel.

5.5 Overlay of multiple channels

A common task in fluorescent microscopy is to create a combined image from the different channels. The task consists in creating an overlay of the different channels, in adjusting the display of each channel and eventually in transferring the settings from one overlay to another, to allow a visual comparison. In ImageJ this can be accomplished using so called hyperstacks. Hyperstacks allow to work with multidimensional images. The different dimensions are the x,y and z axis, the time and the channel (representing the color or wavelength).

Open the images *dapi 3.tif* and *rhod 3.tif* from the folder *11 – overlay*.

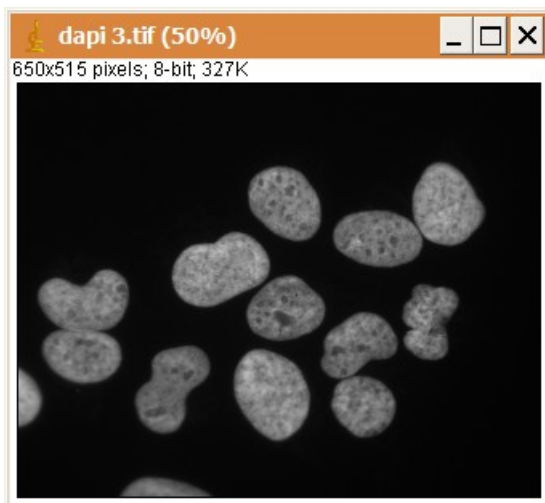


Figure 68: The first channel contains the dapi staining.

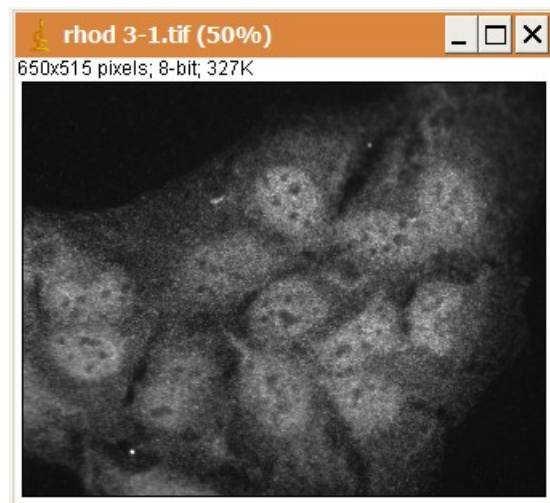


Figure 69: The second channel contains the rhodamine staining.

Run the *Merge Channels...* command from the menu *Image>Color*.

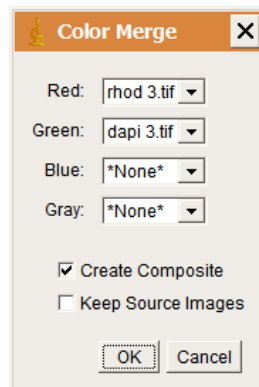


Figure 70: Dialog to create hyperstacks, i.e. multidimensional images.

Select *Create Composite* and press *ok*. You now see an overlay of the two channels. Open the Channels tool from the menu *Image>HyperStacks* and the Brightness and Contrast Adjuster by pressing *shift+c* on the image. With the slider on the bottom of the stack window you select the channel to manipulate. If you move it to the right the histogram in the Brightness and Contrast tool becomes green indicating that the green channel is selected. If you move the slider back to the left the histogram becomes red again. You can change the colors by applying lookup tables from the *LUT menu*. Adjust the display of the two channels using the *B&C tool*. Remark that the display adjustment of one channel remains when you change to the other channel. You don't need to press the apply button.

You can save a hyperstack and reload it without losing the display adjustment. Save it in tif format using *File>Save* or *File>Save As*. Close the hyperstack window and reload the saved file using *File>Open*.

If you want to create a snapshot of the overlay click on the *More* button in the Channels window (*Image>Hyperstacks>Channels Tool...*) and select *Convert to RGB*.

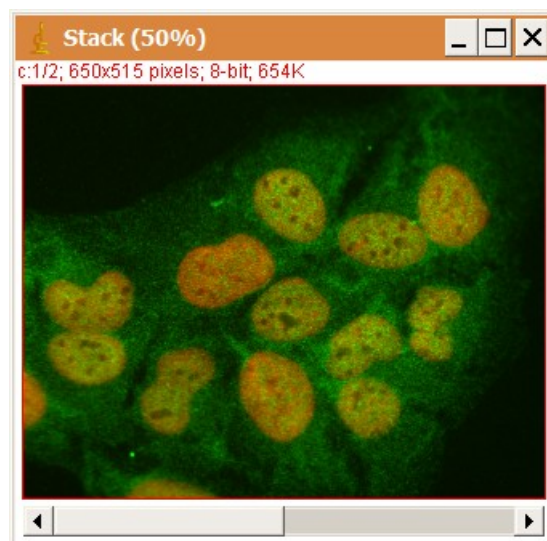


Figure 71: The overlay of two channels.



Figure 72: The channel dialog allows to show and hide selected channels.

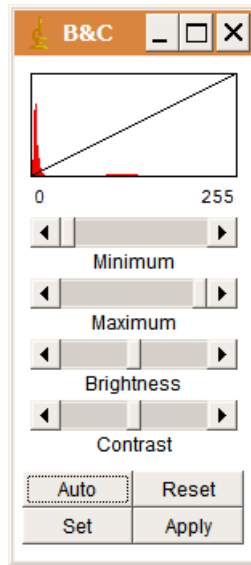


Figure 73: The B&C tool works on the selected channel and displays the histogram in the color of the selected channel.

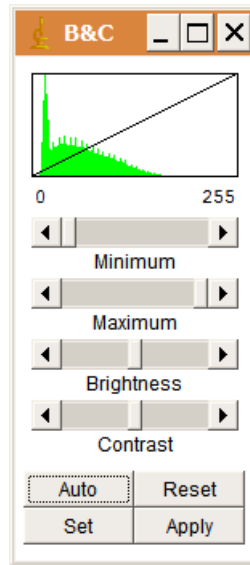


Figure 74: Changing the selected channel automatically changes the channel on which the B&C tool works.

Create a second hyperstack from the images *dapi 5* and *rhod 5*. Imagine that you want to compare the intensities in the two images. To be able to do this you need to set the same display adjustments for both images. Select the red channel in both images. Adjust the display for the first image (for example by pressing reset followed by auto), then press the button set. Select *Propagate to all other 2 channel images* and press *ok*. The display of the red channel in the second image has been adjusted in the same way as the one in the first image now. Select the green channel and repeat the procedure.

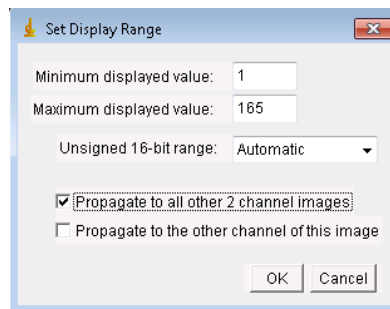


Figure 75: The display settings can be propagated to all open images (same channel).



Figure 76: The display settings have been optimized for the first image.

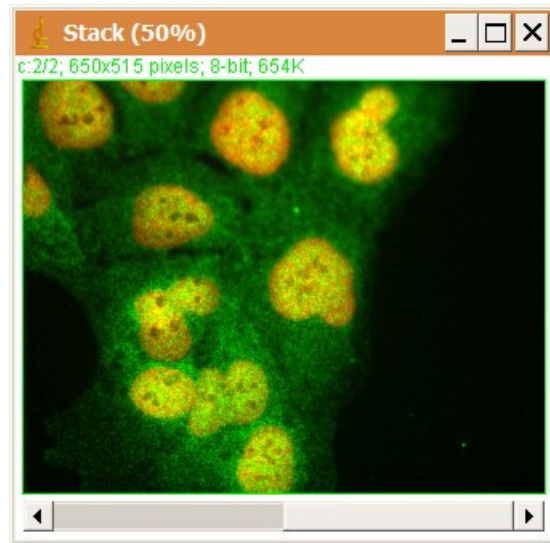


Figure 77: The display settings have been transferred to the second image.

Create an overlay from the three images *b2RFP_gemDeltaC2_blue.tif*, *b2RFP_gemDeltaC2_green.tif* and *b2RFP_gemDeltaC2_red.tif* and adjust the display.

5.5.1 Aligning channels

You can use the composite display to align two channels. Open the images *dapi 4.tif* and *Rhod 4.tif* and create a hyperstack. As you can see the two channels have been shifted against each other. Use *ctrl+a* to select the image with the first channel selected, then press *ctrl+x* to cut the channel and *ctrl+v* to paste it again. You can now move the image of the first channel around. Correct the alignment then select a rectangle around the not empty area of the image and call *Image>Crop* to get rid of the border.

5.5.2 Overlay of volume images

We will now display an image with 54 z-slices and 3 channels. Open the images *Dapi.stk*, *Gtub.stk* and *Actine.stk*. Create the Composite image using *Image>Color>Merge Channels...* You now have a window with two sliders, one to select the channel and one to select the slice in the stack.



Figure 78: A hyperstack of a volume image with three channels.

To create a composite with more than seven channels you can use the command *Image>Hyperstacks>Stack to Hyperstack*.

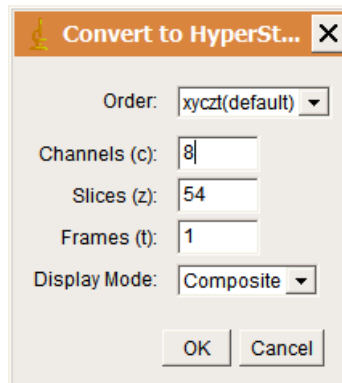


Figure 79: With the Stack to Hyperstack command hyperstacks with more than 7 channels can be created.

5.6 Noise suppression

Open the image *plant-noise.tif*. The image contains a high level of noise. Zoom into the image. The background should be homogeneous, but it contains a random distribution of intensities. The same random distribution is added to the signal.

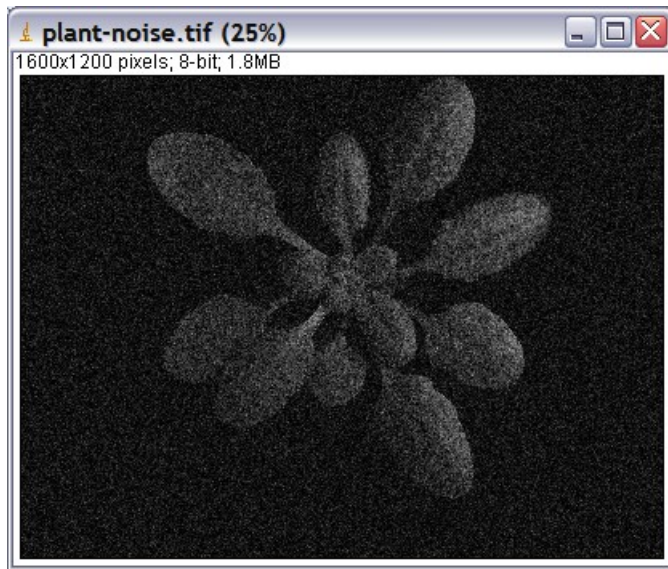


Figure 80: An image with a high level of noise.

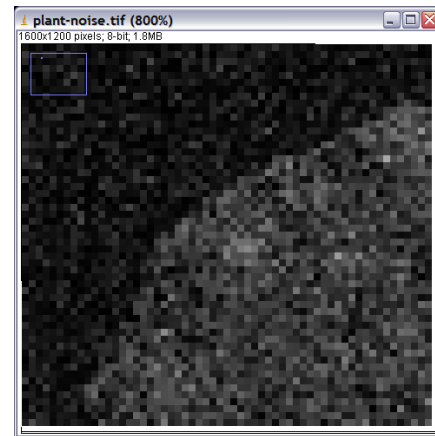


Figure 81: Pixel values in background and object are changed in a random manner.

There are different possible sources of noise in an image. One kind of noise stems for example from the digital camera. In the camera incoming photons are transformed into an electrical charge by a charge coupled device or CCD. However some electrons are created within the CCD randomly. Another source of noise is the random emission of photons from your specimen, which is a quantum physical phenomenon.

The noise can pose a problem for the analysis of the image, for example in the separation of the objects from the background. If we have a light object on a dark background we could for example look for the lowest intensity value in the object and separate the object from the background by searching all pixels that have an intensity value higher than this threshold value. However the noise has changed the intensities in the background and in the object, so that there are very high intensities within the background and very low intensities in the object. We might have to pre-process our image to reduce the disturbing effect of noise.

5.6.1 Convolution filter

5.6.1.1 Mean filter

An evident way to smooth our image and to reduce the impact of the noise is to replace each pixel in the image by an average of the intensities in its neighborhood. This is called a mean filter in ImageJ. You can find it under *Process>Filters>Mean*. You have to enter the radius of the neighborhood in which the average for each pixel is computed. If the radius is 1 the neighborhood has a size of 3x3 pixels. Run the filter with different values for the radius and compare the results by zooming into the images. To run a mean filter with the radius 1 you can use *Process>Smooth* or *Shift+S*.

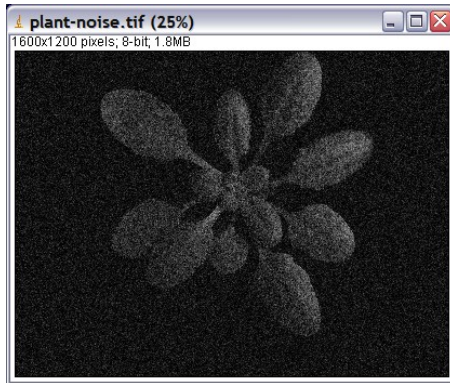


Figure 82: An image containing noise.

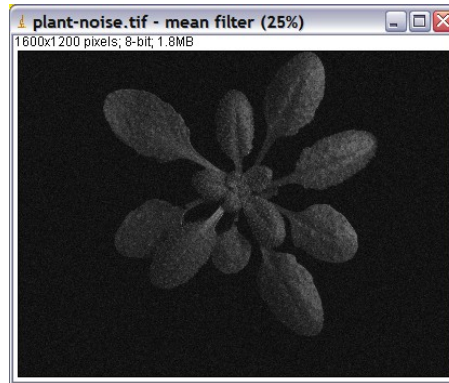


Figure 83: The image filtered with a mean filter of radius 1.

Open the image rectangle. You see a filled rectangle with intensity values 255 on a black background. What will be the intensity values of the edge and corner pixels after applying a mean filter with radius 1?

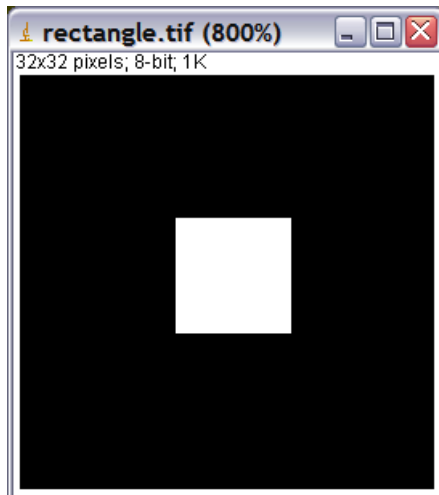


Figure 84: A white rectangle.

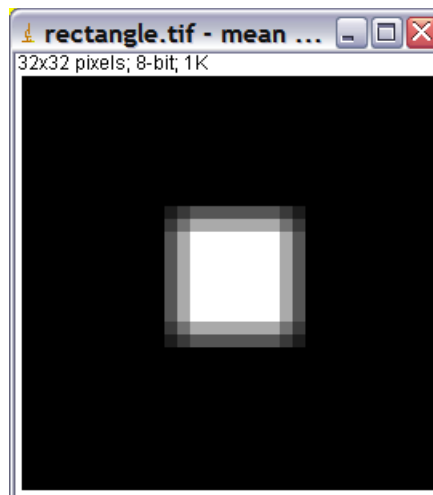


Figure 85: The white rectangle after applying a mean filter of radius 1.

Compare the histograms of the original plant image and the plant image after a mean filter has been applied. Use a large radius, for example 15. Press *h* on the image or use *Analyze>Histogram* to display the histograms of the images. Use the brightness and contrast adjuster to make the background as dark as possible and the plant as light as possible.

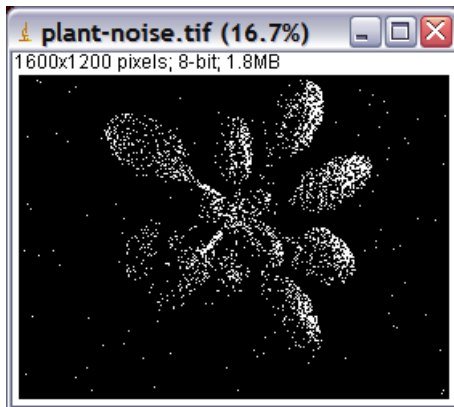


Figure 86: Separation of object and background is impossible in the noisy image.

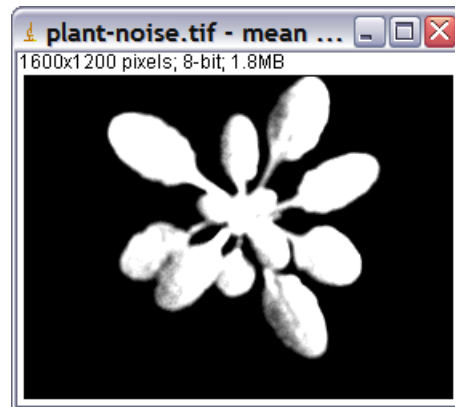
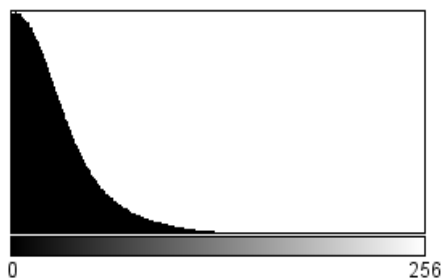


Figure 87: The separation becomes possible after application of a mean filter.



Count: 1920000 Min: 0
 Mean: 27.252 Max: 226
 StdDev: 23.572 Mode: 2 (48226)

Figure 88: The histogram of the noisy image.



Count: 1920000 Min: 15
 Mean: 27.250 Max: 98
 StdDev: 14.454 Mode: 20 (701069)

Figure 89: The histogram after a mean filter of radius 15 has been applied.

A mean filter can be applied in the following way:

- For radius r create a matrix of size $N = 2r+1 \times 2r+1$ filled with values $1/N$. The matrix is called the kernel of the filter.
- Move the center of the kernel across the image. For each position:
 - Multiply each matrix element with the corresponding intensity value and calculate the sum of the results.
 - In the result image replace the intensity of the current pixel with the calculated result.

A generalization of this technique is called convolution. A convolution is an operation that calculates the overlap of two functions. In the general case the kernel and the image can have infinite size.

Instead of simply calculating the average, we can calculate a weighted average by using different values in the matrix. Convolution filters can not only be used for smoothing but also for other purposes, for example to enhance edges.

To apply a convolution filter go to *Process > Filters > Convolve* and enter the matrix in the dialog. To create a mean filter of radius 1 put in a 3×3 matrix, in which all elements are 1. Normally we would have to use $1/9$, but we can select the “normalize kernel” option and the computer will do this for us.

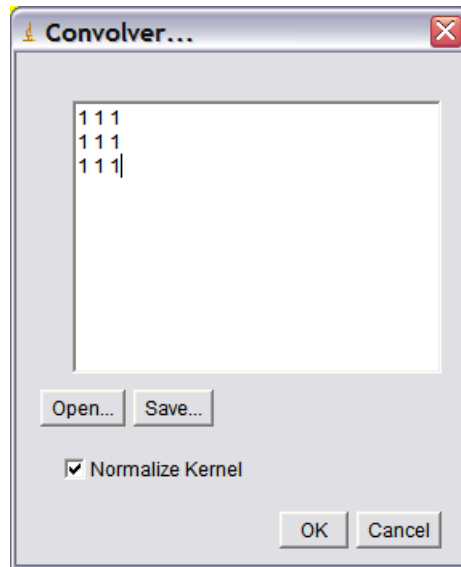


Figure 90: Applying a mean filter by doing a convolution.

Apply the convolution filter. Compare the result with the result from the mean filter. To test whether the results are absolutely identical you can subtract one result image from the other. If the images were identical the result image must be all black. Use *Process>ImageCalculator* to subtract one image from the other. Apply the hilo lut to the result to see if intensities other than 0 are present.

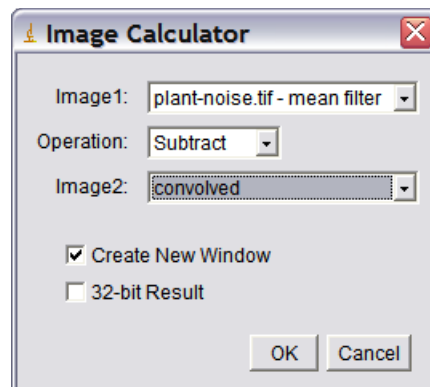


Figure 91: Subtracting one image from another image.

The smoothing reduces the deranging effect of noise but in the same time blurs the object, so that less details are visible.

5.6.1.2 Gaussian blur filter

A Gaussian blur filter is similar to the mean filter, but instead of using uniform weights in the kernel, the weights are the values of a normal distribution, also called Gaussian distribution. A normal distribution is used to describe random processes. Throw a coin 100 times and count the times the head is up. Repeat this 1000 times and draw a histogram of the outcomes. Most of the time you'll get values around 50, resulting in high bars around 50 in the histogram. The case that in 100 throws only 1 head turns up will not occur very often. The

same will be the case for 100 heads in 100 throws. If you connect the top points of the bars you get a curve that looks like a normal distribution and if instead of throwing the coin 1000 times you throw it an infinite number of times you'll get a normal distribution. A two dimensional normal distribution has the form of a bell.

$$\varphi_{\mu,\sigma^2}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Figure 92: A gaussian function.

In older versions of ImageJ the Gaussian Blur filter had the radius as a parameter. In newer versions (from 1.38q) the parameter is sigma, the standard deviation of the distribution. To get the same results the values in the old version have to be multiplied by 2.5. Open the file *Gaussian.txt*. Here you see the values of a normal distribution. How does it look like as an image? Open the file as an image by using *File>Import>Text Image*. Apply the ice lut. Now go to *Analyze>Surface Plot* or use the *Interactive 3D surface plot* plugin (Plugins>3D) to get an impression of the 3D form. Because of the discrete nature, what you see looks less smooth then the real form. Besides the real curve never drops down to zero, but values will be very low outside the central area so that we can use a truncated Gaussian as an approximation.

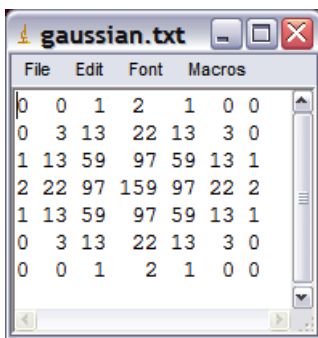


Figure 93: Convolution kernel for a (truncated) gaussian blur filter.

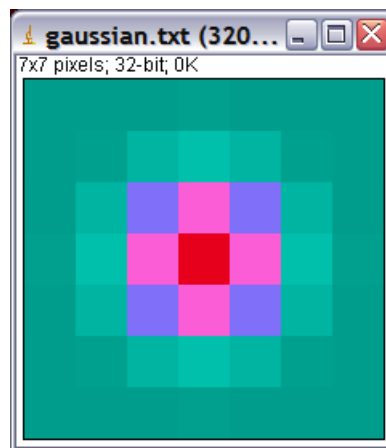


Figure 94: The kernel as image.

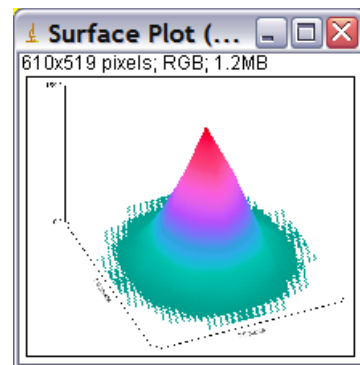


Figure 95: The kernel as surface plot.

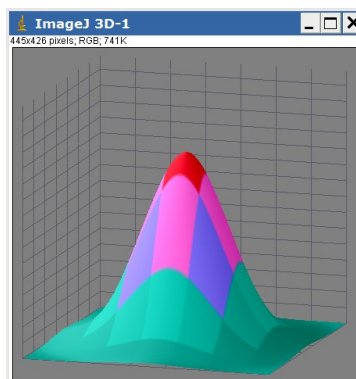


Figure 96: The kernel as a surface plot created with the «interactive 3D surface plot» plugin.

Apply the Gaussian blur filter with sigma 1.2 from *Process>Filters>Gaussian blur* to the rectangle image and compare the result with the result of the mean filter with radius 3.

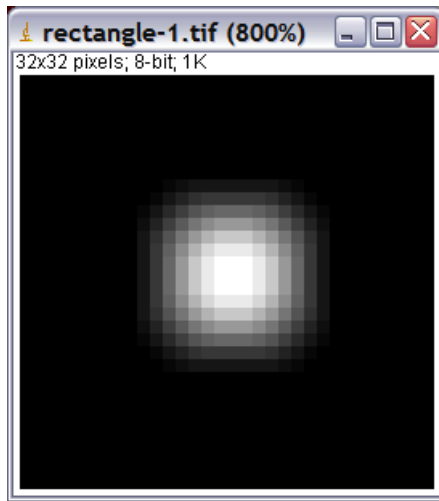


Figure 97: Mean filter with radius 3 applied to the rectangle image.

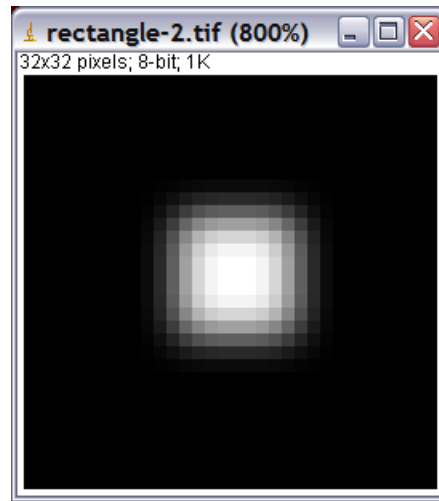


Figure 98: Gaussian blur filter with sigma 1.2 applied to the rectangle image.

Although both images are blurred, the result from the Gaussian filter looks more like the original form. The Gaussian filter while removing noise as well, keeps edges better. Apply a Gaussian blur filter with a sigma of 6 and a Mean filter of radius 15 to the image roots and compare the results.

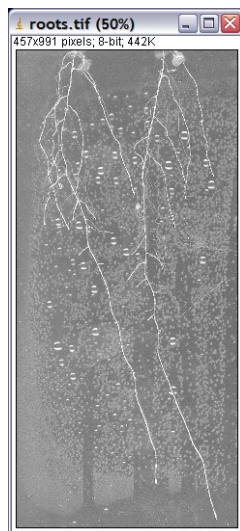


Figure 99: The roots image.



Figure 100: A mean filter of radius 15 has been applied to the image.



Figure 101: A gaussian blur filter of radius 6 has been applied to the image.

Open the plant-noise image. Open the convolution tool and load the Gaussian.txt into it. Run the convolution filter on the image.

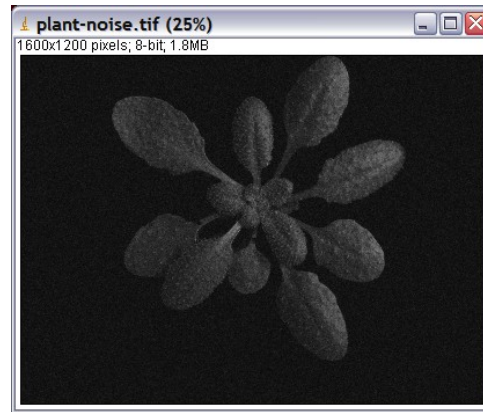


Figure 102: The plant image after the convolution with the gaussian kernel has been applied.

5.6.1.3 Edge enhancing filter

Create a convolution filter of radius 1. Put 1 1 1 in the first row, -1 -1 -1 in the last row, and 0 0 0 in the middle row. Apply it to the rectangle image, after having converted it to 32-bit. Apply another filter with the values above in the columns instead of the rows to the original 32-bit image. Square the result images, add them and apply square-root to the result. This is called a Prewitt filter. To see the effect of the filter, create an overlay of the original and the filtered image, by using *Image>Color>Merge Channels*. Put the filtered image in the red channel, the original image in the green channel and choose none for the blue channel.

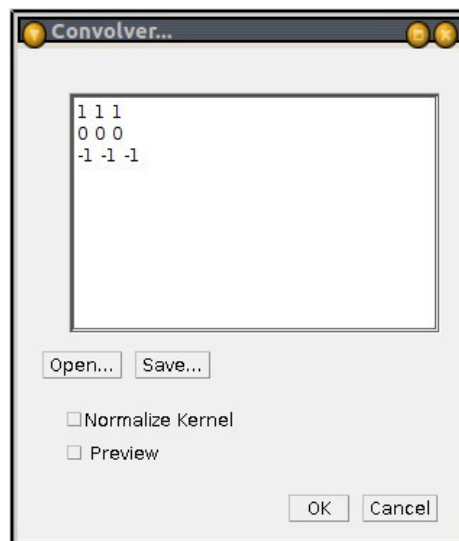


Figure 103: The convolution kernel for an edge enhancing filter.

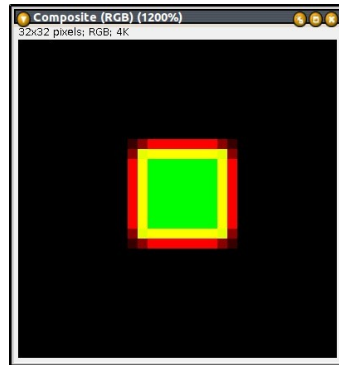


Figure 104: Channel-merge of the original image and the result after application of the filter.

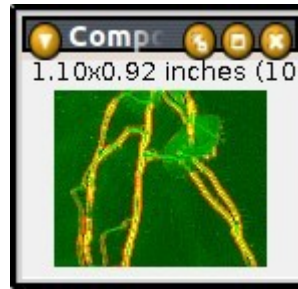


Figure 105: Detail of the channel-merge of the roots image and the result from the convolution that detects contours.

Apply the Prewitt filter to the roots image after having converted it to 32-bit. Using the same

procedure with the kernels $\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$ and $\begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$ gives the Sobel-filter. In

ImageJ the command `Process>Finde Edges` implements the Sobel-filter.

5.6.2 Rank Filter

Rank filters are another class of filters. Again we look at a neighborhood of radius r for each pixel. But this time the intensity values are sorted and the pixel is replaced with the value in a specific position.

If we use the middle position we get a median filter. If we use the first position we get a minimum filter and using the last position we get a maximum filter.

Open the image `plant-sp-noise.tif`. Zoom in to the image. You see that in this case the noise consists of the addition of white and black pixels to the image. This kind of noise is called salt-and-pepper noise.

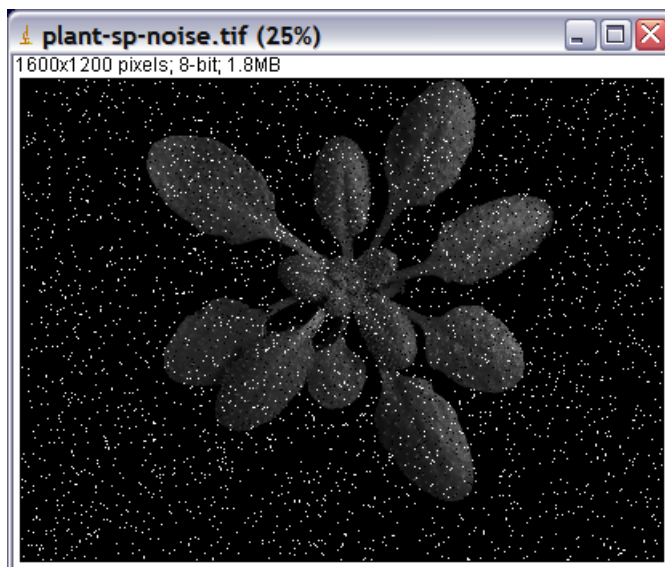


Figure 106: The plant image with salt-and-pepper noise.

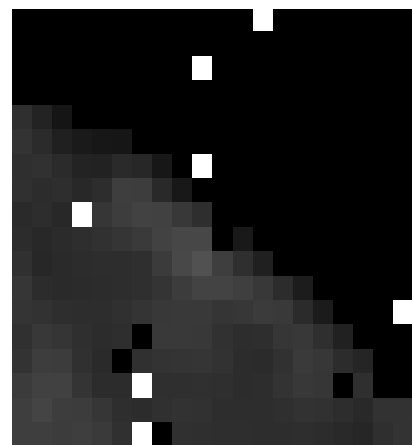


Figure 107: Salt-and-pepper noise consists of the addition of white and black pixels to the image.

Apply a median filter of radius 3 from *Process>Filters>Median* to the image. Compare the result with the result from the mean filter. The median filter is more effective in removing salt-and-pepper noise.

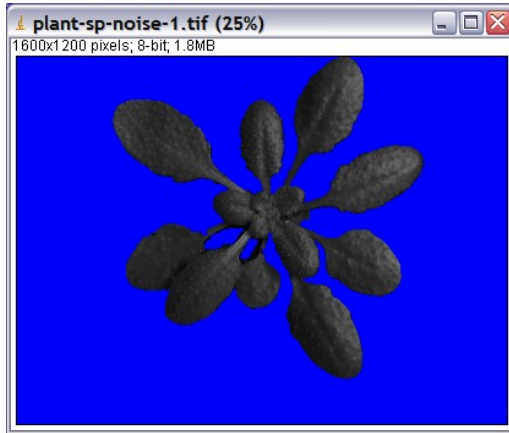


Figure 108: A median filter effectively removes salt-and-pepper noise.

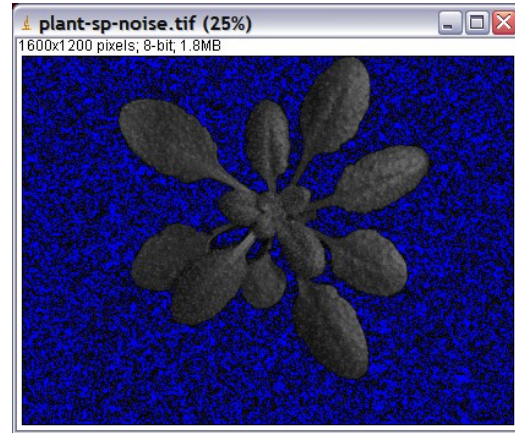


Figure 109: A mean filter is not so effective on salt-and-pepper noise.

You can run a median filter with radius 1 using *Process>Noise>Despeckle*.

In *Process>Noise>Remove Outliers*, you find a selective median filter, that replaces a pixel by the median in the neighborhood if the value of the pixel differs more than a threshold value from the median. Use it to remove the dark and light components of the salt and pepper noise separately.

5.6.3 Filtering in the frequency domain

In its normal representation an image consists of an intensity value for each spatial coordinate. We call this the spatial domain. The same information can be represented in other ways. Some operations are easier to apply with a different representation.

One important representation is to transform the image into the (spatial) frequency domain. Here each value represents the frequency of change of the intensity value. Large structures have a low frequency, small repeating structures have a high frequency. The Fourier transform transforms the image from the spatial to the frequency domain. The transform is based on the fact that each signal can be presented as a sum of harmonic (sinoid) functions of different phase and amplitude. The result is an image of complex numbers that can be divided in a real part, the power spectrum and an imaginary part, the phase information. From the result the original image can be reconstructed without any loss by applying the inverse Fourier transform. Usually only the power spectrum is shown, since it contains most of the interesting information. However to reconstruct the original image, both, the power spectrum and the phase image are needed.

There exists an efficient way to compute a Fourier Transform. It is called the Fast Fourier Transform or FFT. We can use the Fourier Transform to filter our image in the frequency domain.

One reason why the frequency domain is interesting is because we will be able to recognize frequencies that appear as clusters visually. We can use this to filter out frequencies that correspond to objects or patterns of different size in the image.

Low frequencies will be displayed in the middle of the power spectrum and the higher the frequencies are, the bigger is their distance to the center. If we filter out the low frequencies in

the middle, we apply a high pass filter. This can be used as an edge enhancing filter since the edges information is in the high frequencies.

If we filter out the high frequencies, we apply a low pass filter. This has a smoothing effect on the image, since the fine details have a high frequency.

Another reason why the Fourier Transform is important is because we can calculate a convolution more efficiently in the Fourier Domain. The Convolution Theorem states that a convolution in the spatial domain is equivalent to a multiplication in the frequency domain. Open the image *Pierre.tif*. Apply the Fourier transform from *Process>FFT>FFT*.



Figure 110: An image.

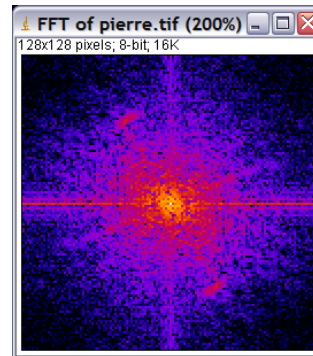


Figure 111: The power spectrum of the image.

You can use the lut tool to make visible different clusters of frequencies. Do you see the two clusters in the upper left and lower right quadrant. Draw a selection around the first. Then add a selection around the second by pressing shift when you start the selection. You can release the shift key once you started the second selection. Now run the fill command from *Edit>Fill* or press *CTRL+f*.¹ We have set the intensities in the selection to zero and in this way suppressed the corresponding frequencies. Apply the inverse Fourier transform from *Process>FFT>Inverse FFT*. At first look the image doesn't seem to have changed. Zoom in on the image and on the result image behind the ear. You see that the lines of the window blinds have disappeared in the filtered image, since we suppressed the corresponding frequencies.

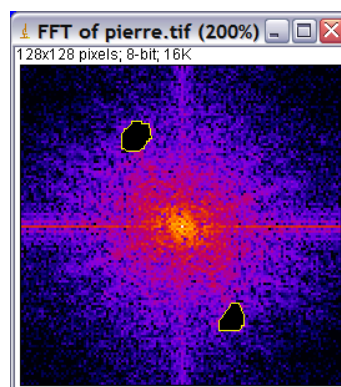


Figure 112: Filling parts of the power spectrum image with the background color filters out the corresponding frequencies.

¹ Fill actually uses the current foreground color. Double click on the color picker tool and set the current foreground color to black.



Figure 113: Detail of the original image.

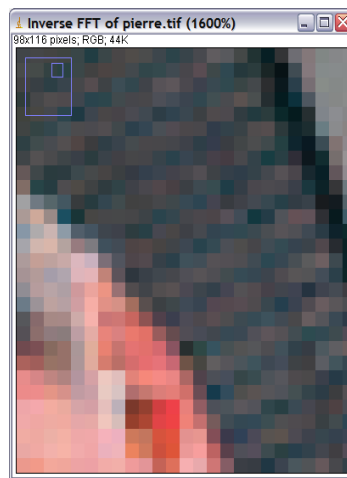


Figure 114: Detail of the filtered image.

Subtract the filtered image from the input image to see what exactly has been filtered out. Instead of building the difference you can as well go back to the power spectrum and fill the inverse of the mask.

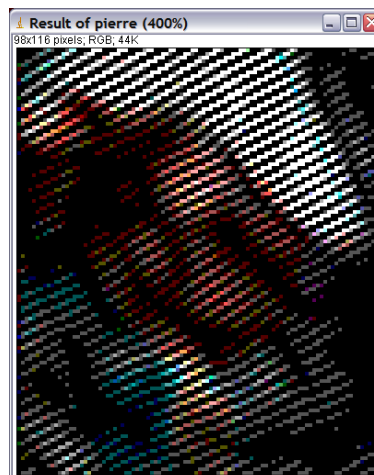


Figure 115: The difference of the original image and the filtered image.

Use *Process>FFT>Redisplay Powerspectrum* to redisplay the original power spectrum. Try to smooth the plant-noise image. Which frequencies do you have to filter out?

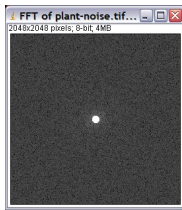


Figure 116: Using clear instead of fill lets the corresponding frequencies pass.

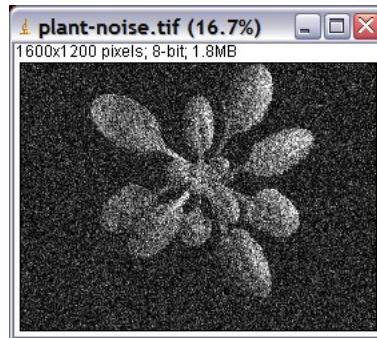


Figure 117: The original image.

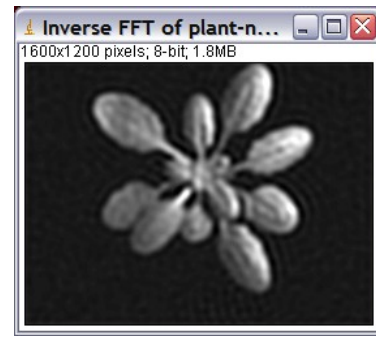


Figure 118: The image after filtering in the frequency domain.

Try to enhance the edges in the roots image. Always build the difference of the original image and the filtered image to see what exactly has been filtered out.

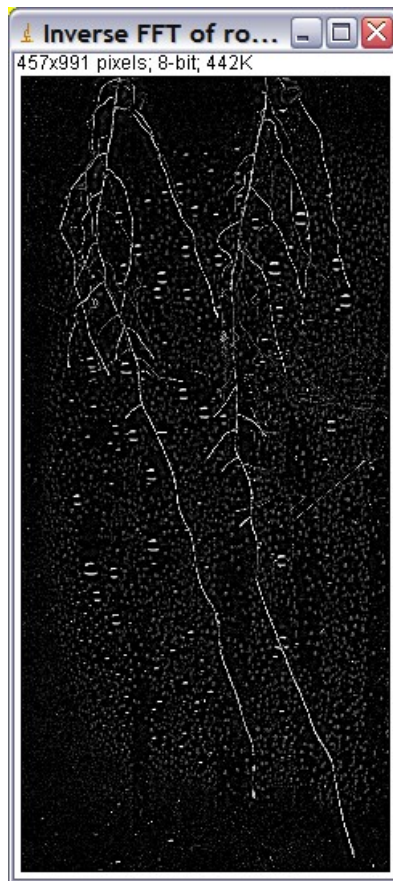


Figure 119: The roots image after a high pass filter has been applied.

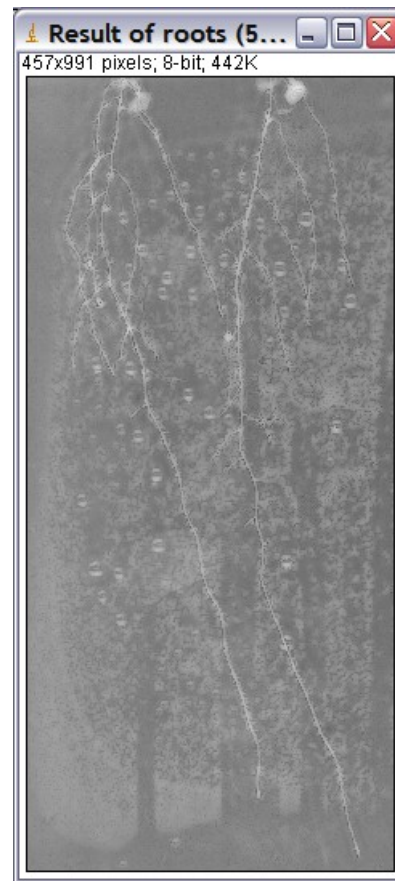


Figure 120: The difference of the original and the filtered image.

If we set everything, except for a ring, to zero in the power spectrum, we will filter out all frequencies higher than a certain value and all frequencies lower than a certain value. This is called a band-pass filter, since the only frequencies that pass are those within the ring. Try it for example with the image *pierre.tif*. Draw a circular selection and then remove a second circle from within the selection by pressing the alt key while making the second

selection. Use the I from the toolbox to invert the selection and apply fill from the context menu.

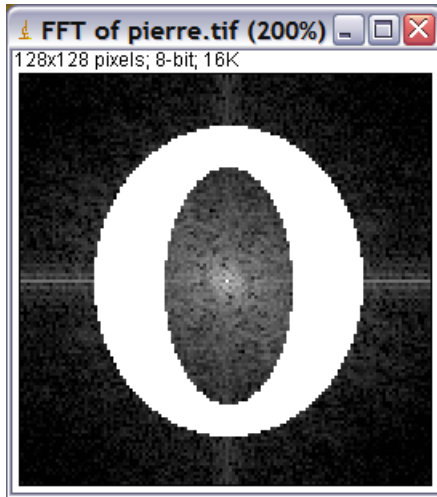


Figure 121: Defining a bandpass filter.



Figure 122: The bandpass filter applied to an image.

You can apply a bandpass filter using the command *Process>FFT>Bandpass Filter...*

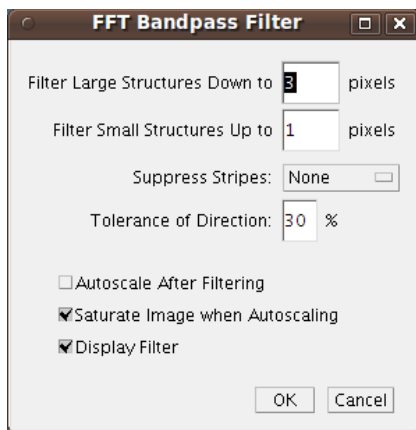


Figure 123: The FFT bandpass filter dialog.

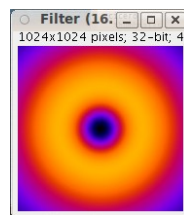


Figure 124: A bandpass filter.

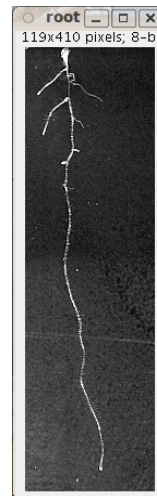


Figure 125: The input image with a gradient in the background.

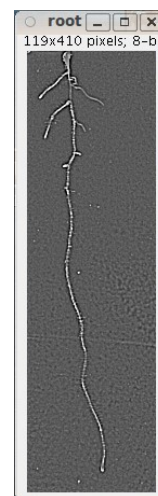


Figure 126: The result of the bandpass filter.

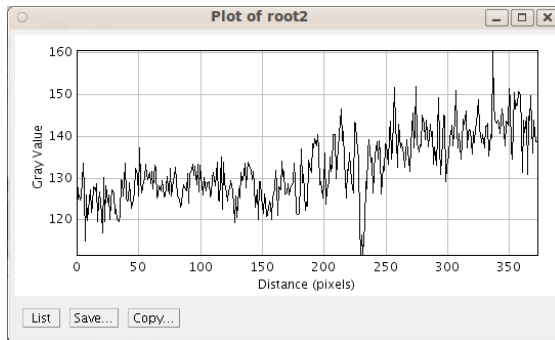


Figure 127: The profile plot shows the gradient in the background.

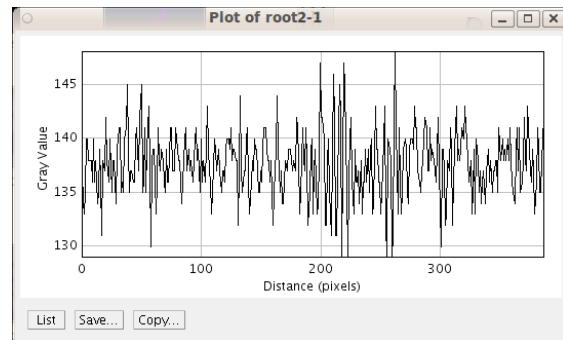


Illustration 128: The profile plot shows that the gradient has disappeared after the application of the bandpass filter.

5.7 Background subtraction

Open the image *nuclei-rhod.tif*. Apply the *hilo lut*. As you see no pixels become blue, meaning that the background is not zero. Since this image is from fluorescent microscopy, normally the background should be black, in reality however there is some intensity in the background.

In this example the background is low and, except for the noise, homogeneous. A high background can make it harder to clearly distinguish objects. Furthermore if we want to compare intensities between images with different background levels we have to compensate for the background.

Use the B&C tool to clearly distinguish a background region. Draw a line selection on the background and create a profile plot using *Analyze>Plot profile* or select the image and press *k*. The profile plot shows the intensity values along the line selection. You see that the values vary somewhere between 1 and 9 due to noise and that the average is somewhere around 3. Press the measure button on the toolbox or select the image and press *ctrl+m* to get the exact mean value.

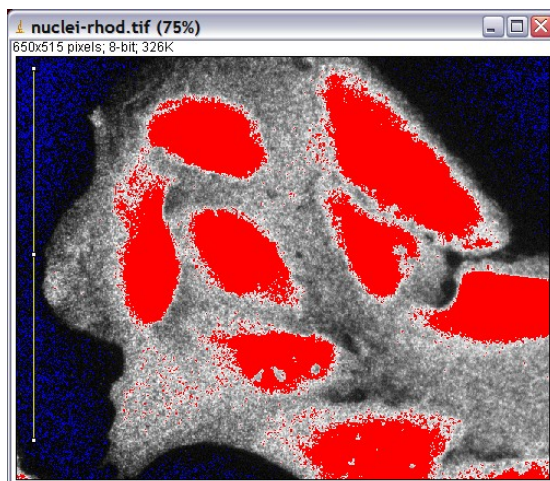


Figure 129: The background should be black, but contains some intensity in reality.

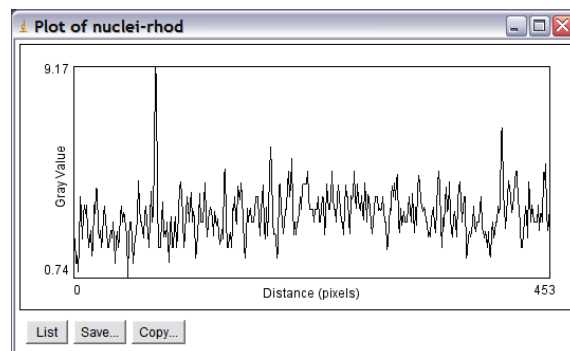


Figure 130: The line plot displays the intensities along the selected line.

Go to *Process > Math > subtract* and subtract the mean background value from the image. Use the *R* button from the toolbox to get back the line selection and create the profile plot again. This time the values go down to zero.

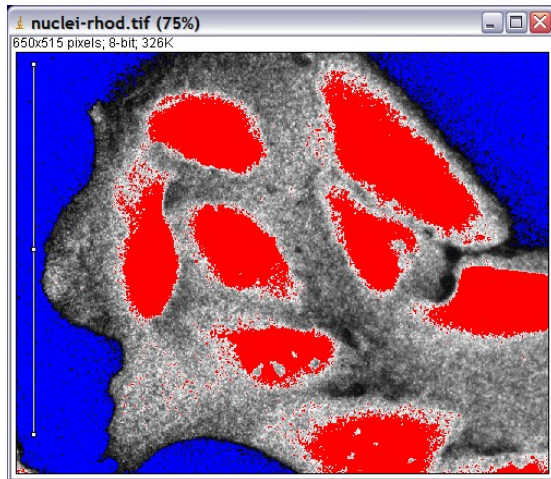


Figure 131: The image after the value 3 has been subtracted from each pixel value..

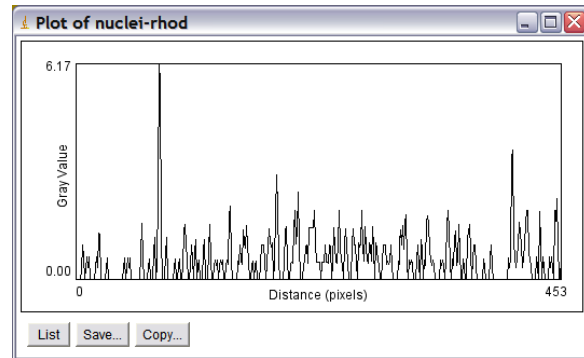


Figure 132: The line plot goes down to zero now.

5.7.1 Inhomogeneous background

Open the image *root2.tif*. Draw a vertical line on the background again and create the profile plot. As you see this time the background baseline is not constant. The image is darker in the top area and lighter in the bottom area. Measure the mean and subtract it from the image. Create the profile plot again. You see that, although the background value is lower now, the gradient is still present.

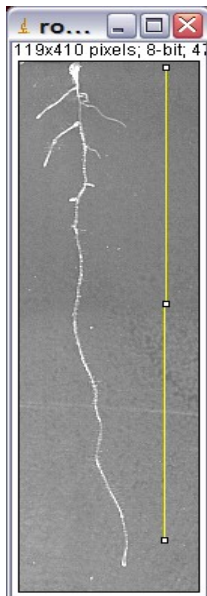


Figure 133: An image with an inhomogeneous background.

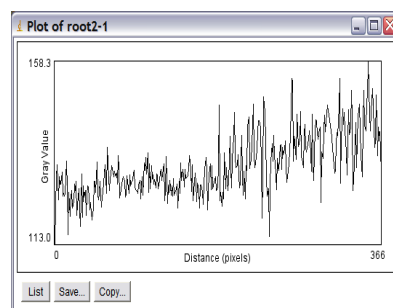


Figure 134: The line plot reveals a gradient from the top to the bottom of the image.

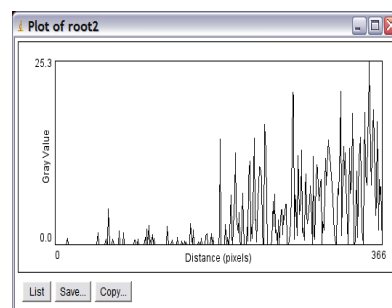


Figure 135: After subtraction of the average background intensity the gradient is still present.

Apply the operation *Process>Subtract Background* to the image and create the profile plot. The gradient has disappeared. The command uses a rolling ball algorithm that, roughly described works in the following way: Imagine the intensity values as a height and the image as a 3d landscape. A ball with the given radius rolling over the backside of the 3d landscape creates the background image that is then subtracted from the original image. In this way the local information is taken into account.

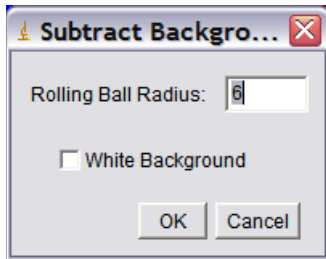


Figure 136: The options dialog of the subtract background command.

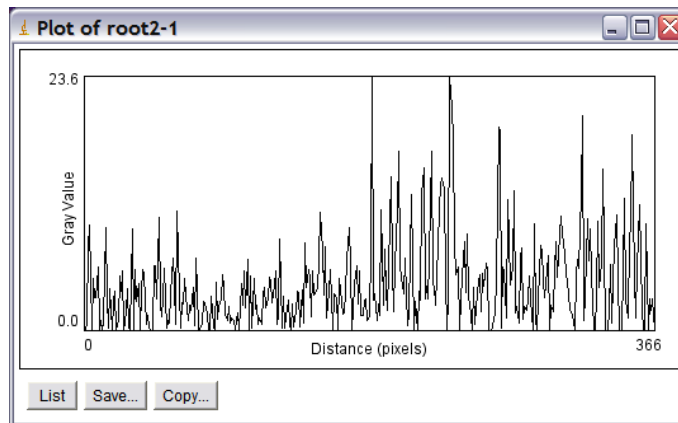


Figure 137: The subtract background command removes the gradient using a rolling ball algorithm.

One very elegant way to remove background is to make an image of the empty scene along with the image of the specimen. One can then divide the specimen image by the background image and multiply the result with the mean intensity in the background image. This is often possible in microscopy and the technique is called flat field correction. If it is not possible to take the “empty” image we can simulate the process by applying a very large filter that will remove the object from the image. We then have to divide the result from the original image and multiply with the average intensity of the filtered image. You should use 32bit for the result of the image-division operation. You can always reconvert the final result back into an 8bit image if you want. Look at the profile plot from the result. Again the gradient has disappeared. The Pseudo Flat Field correction applies a mean filter to the image. You can do the same thing using other filters. Try a Gaussian blur filter for example.

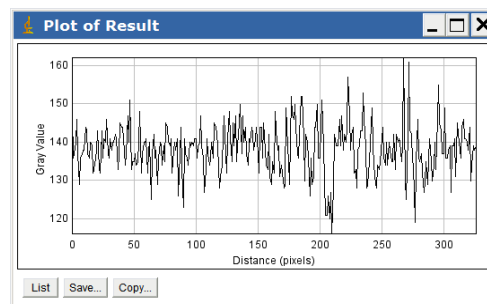
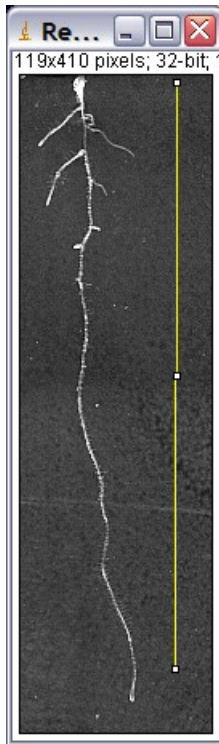


Figure 141: Line plot after pseudo flatfield correction. The gradient has disappeared.

5.8 Increasing the apparent sharpness

Open the image *hst2.tif* from the folder “*cells in structures*”. Zoom in 2 or 3 times. The image appears somewhat blurred or unsharp. Duplicate the image with the duplicate button from the tool box and apply a Gaussian blur with sigma 0.8. Multiply the result with the factor 0.6 and subtract the result from the original image.

This technique is called unsharp masking. You can use *Process>Filters>Unsharp Mask* to do it in one step.

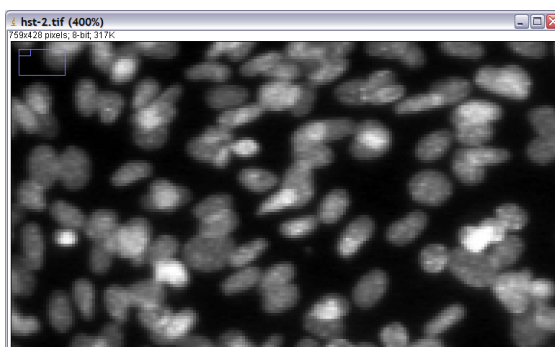


Figure 142: The image appears somewhat unsharp or blurred.

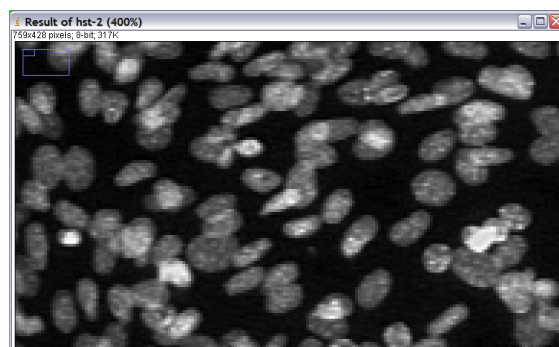


Figure 143: After applying the unsharp mask filter the image appears less blurred.

6. Segmentation

Segmentation is the process of separating the objects from the background and from one another. The simplest way of segmentation is based on the difference in the intensities of objects and background. One looks for a minimum and maximum intensity value, so that all pixels belonging to the objects one is interested in have intensities between min and max and all other pixels have intensity values below min or above max.

The result of a segmentation is often represented as a binary mask. A binary mask is an image that contains only two values. The first value signifies that the pixel belongs to the background and the second signifies that a pixel belongs to the object. In ImageJ the values 0 and 255 are used. Once we created a mask, we can combine it with the original image, for example to suppress everything but the objects we are interested in.

Another way of presenting the result of a segmentation is a selection. The selection represents the borders of the objects we are interested in. A selection itself is not an image. ImageJ stores the selection together with the image. A selection is called a region of interest or ROI in ImageJ. We can turn a mask into a selection and vice versa and use the representation the most practical for a given task. A selection can be created from a mask with the help of the magic wand or tracing tool. Given a selection, a mask can be created by using fill on the mask, then inverting the selection and using clear. The command *Edit>Selection>Create mask* can be used to create a mask from a selection in one step. The command *Edit>Selection>Create Selection* can be used to create a selection from a mask or from an image on which the lower and upper threshold are set in one step.

6.1 Manual threshold selection

The threshold value(s) can be either chosen manually or automatically by an algorithm. The most simple automatic thresholding algorithms try to find a threshold that divides the histogram in a meaningful way.

Open the image *plant.tif*. Open the threshold adjuster from *Image>Adjust>Threshold* or press *shift+t* on the image. You can choose between three modes in the threshold tool. The first displays selected pixel in red and pixels not selected with their original intensities. The second displays the mask that will be the result after applying the threshold, i.e. background in white and objects in black. In the third mode, selected pixels are displayed with their intensities, pixels below the lower threshold are displayed in blue and pixels above the upper threshold are displayed in green.

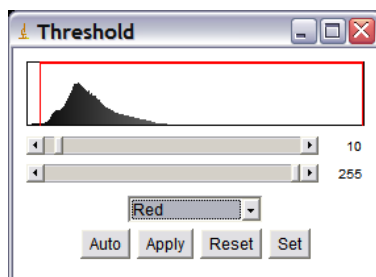


Figure 144: The threshold adjuster.

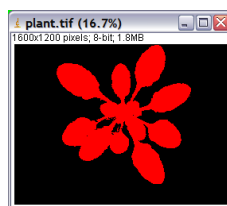


Figure 145: Object red, background black.

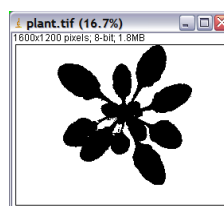


Figure 146: Object black, background white.

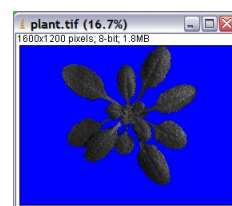


Figure 147: Object graylevels, background blue (below first threshold) and green (above second threshold).

Adjust the threshold to select the plant. The wand tool works together with the threshold tool. So if you want to select the plant, there is no need to create the mask. Use the wand tool to

select the plant. Use the alt key to keep out the two holes between the leaves. You can add to a selection by using *shift* and subtract from a selection by using *alt*. Alternatively you can use *Edit>Selection>Create Selection* after you changed the lower and upper threshold with the threshold adjuster. In contrast to the *create selection* command, using the wand tool allows to select one separated object in an image containing multiple objects. If you press the measure button on the toolbox or the *m* key on the image, you get a result table with measurements of the selection.

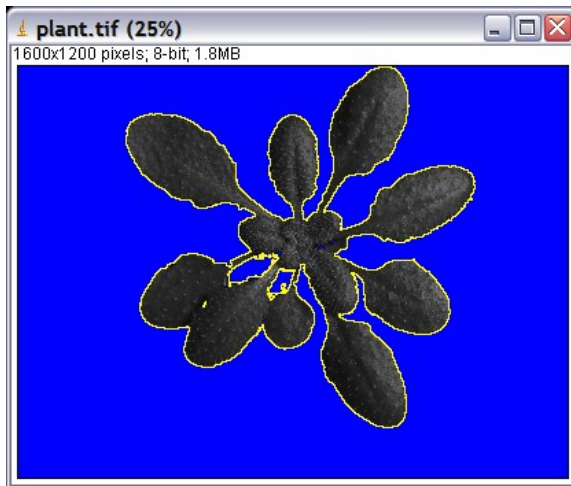
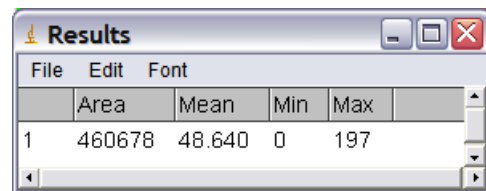


Figure 148: Selection created with the threshold adjuster and the wand tool.



File	Area	Mean	Min	Max
1	460678	48.640	0	197

Figure 149: Results table with the measurements from the selection.

Remove the selection by pressing *shift+a*. Run the Particle Analyzer to measure all objects with intensities between the selected thresholds. Use the *Apply* button if you want to create a mask.

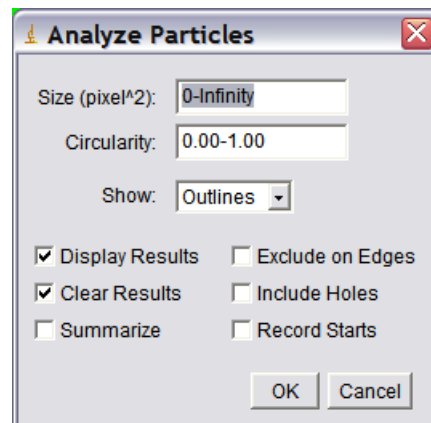


Figure 150: The particle analyzer.

You can filter out objects by their size and circularity and you can create a mask image or an outline image containing the contours of the measured objects. In the later image the objects will be numbered, as well. You can choose to exclude objects touching the image borders and to include or not include holes in the objects in the mask.

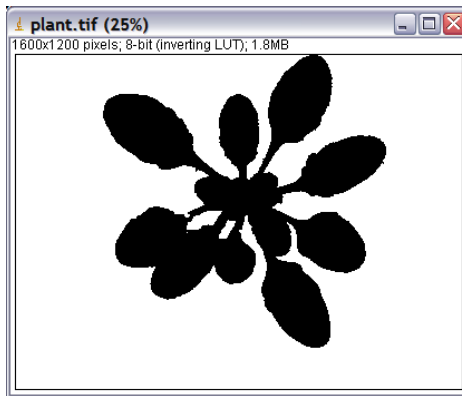


Figure 151: A mask for the plant image.

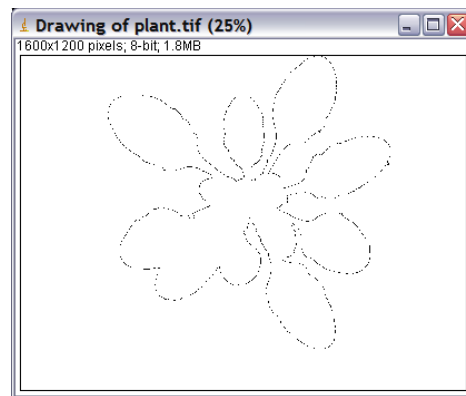


Figure 152: An outline of the plant created with the particle analyzer.

You can configure the values that will be measured by the Particle Analyzer under *Analyze>Set Measurements*. Here you can as well configure a redirect image. In this way you can use a mask but the intensities will be measured in the redirect image, for example the original image the mask was created from.

Use the Particle Analyzer to measure the plant with the intensity values from the original image and display the outline result.

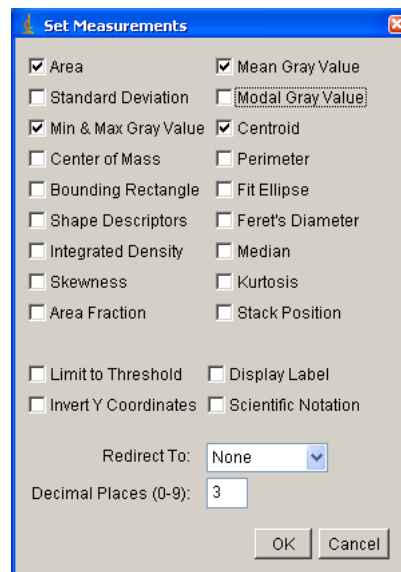


Figure 153: The set measurements dialog. The settings are used by the measure command and by the particle analyzer.

Another way to use a mask to measure in another image, is by using the wand tool on the mask and then transferring the mask to the other image by activating the image and pressing *shift+e* on the image. You can then use the measure command (*m*) to measure the content of the selection.

Open the *plant-noise.tif* image and try to segment it with the help of the threshold adjuster. You will see that the noise prevents you from making a good segmentation. Try one of the noise suppression techniques we discussed and try the segmentation again.

6.2 Measurements and the ROI-Manager

Open the image *A4 dapi1.tif*. Set a threshold that separates the nuclei from the background. Run the *Particle Analyzer*. Select the *Add to Manager* option. For each particle detected by the Particle Analyzer, a roi is added to the ROI-Manager. The *Show All* button shows or hides all rois in the manager on the active image.

If you click on one roi in the list, it will be set to the active image. Clicking on a label of a roi in the image, selects the roi in the list. Now deselect all rois and press the measure button.

You get a results table with the measurements of all rois. If you select a number of rois using shift, only the selected rois will be measured. In general the commands in the roi manager are either applied to the selected rois or to all rois if there is no selection.

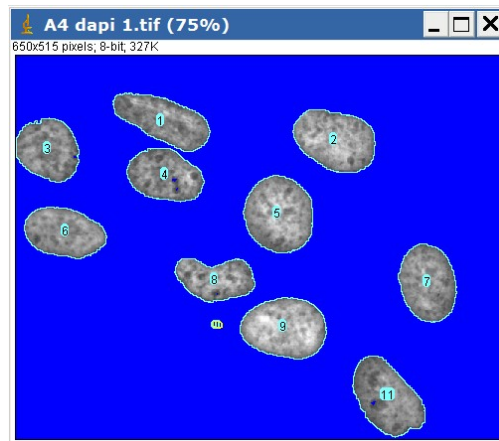


Figure 154: The roi manager displays the rois of all particles.

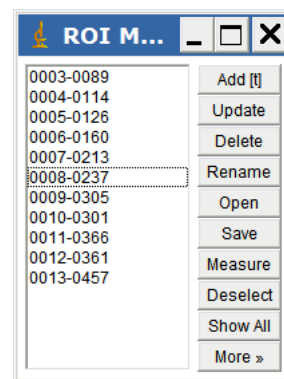


Figure 155: The rois of the particles have been added to the roi manager.

	Area	Mean	StdDev	Mode	Min	Max	X	Y	XM	YM	Perim.
1	6117	126.136	25.681	116	68	222	198.181	86.632	199.346	86.766	352.818
2	7060	149.329	31.552	157	71	239	428.761	113.516	428.095	113.451	330.534
3	5502	126.578	24.370	116	66	191	43.222	126.345	42.816	126.213	294.635
4	5494	125.617	27.222	135	62	235	199.802	161.013	200.499	161.116	292.149
5	7532	145.791	32.992	150	71	246	353.098	213.308	353.442	212.643	327.948
6	5661	135.269	24.036	145	71	198	64.646	237.529	65.243	237.119	300.735
7	6192	131.989	25.203	132	71	211	552.580	305.328	551.915	304.752	305.806
8	4604	136.906	31.713	130	63	224	267.910	300.179	267.436	299.965	299.806
9	6869	122.956	26.279	126	59	210	497.642	459.132	499.617	459.511	340.274
10	7324	166.883	36.564	167	71	255	360.952	365.877	360.801	365.956	331.463

Figure 156: The measurements of the rois in the roi manager.

Remove particle 10 and 9. Use the wand tool to select particle 10 again and add it again to the roi manager.

Save all rois, then close and re-open the roi-manager and open the saved rois again.

Compare the measured values of different particles. What is the meaning of the measured values?

Area:

The surface of the roi. It is measured in square-pixel if no spacial scale is set.

Mean:

The average gray value within the roi.



Image processing and analysis with ImageJ

StdDev:

The standard deviation of the mean gray value within the roi.

Mode:

The most frequently occurring gray value within the roi. This corresponds to the highest peak in the histogram of the roi. If you display the histogram the mode will be shown together with the number of occurrences.

Min:

The minimal gray value within the roi.

Max:

The maximum gray value within the roi.

X:

The x-coordinate of the centroid. This is the average of the x-coordinates of the pixels in the roi.

Y:

The y-coordinate of the centroid. This is the average of the y-coordinates of the pixels in the roi.

XM:

The x-coordinate of the center of mass. This is the brightness weighted average of the x-coordinates of the pixels in the roi.

YM:

The y-coordinate of the center of mass. This is the brightness weighted average of the y-coordinates of the pixels in the roi.

Perim.:

The length of the outside boundary of the roi.

BX:

The x-coordinate of the upper-left corner of the bounding box of the roi. The bounding box is the smallest rectangle, that entirely contains the roi.

BY:

The y-coordinate of the upper-left corner of the bounding box of the roi.

Width:

The width of the bounding box of the roi.

Height:

The height of the bounding box of the roi.

Major:

The length of the major axis of the best fitting ellipse. The ellipse has the same area, orientation and centroid as the original selection.

Minor:

The length of the minor axis of the best fitting ellipse.

Angle:

The angle of the major axis of the best fitting ellipse against the x-axis of the image.

Circularity:

$4\pi(\text{area}/\text{perimeter}^2)$. A value of 1 indicates a perfect circle. As the value approaches 0, it indicates an increasingly elongated polygon. Values may not be valid for very small particles. The perimeter of a circle is $p = 2\pi r$ and the area is $a = \pi r^2$.

Feret:

The longest distance between two points on the boundary of the roi. You can use the macro *Draw Ferets Diameter* to draw the feret diameter of a selection into the image. Use *Plugins>Macros>Install* and select the macro from the macros folder in the ImageJ base-folder. Make a selection in the image and call the macro from the *Plugins>Macros* menu.

IntDen:

The integrated density is the sum of the gray-values of all pixels within the roi. If the image has a calibration for the gray-values, the calibrated values are used.



Image processing and analysis with ImageJ

Median:

The median gray value of the pixels within the roi, i.e. The gray value that lies in the middle or the value between the two values in the middle, when all gray-values are sorted by their numerical value.

Skewness:

A measure of the asymmetry of the distribution of the gray values around the mean within the roi.

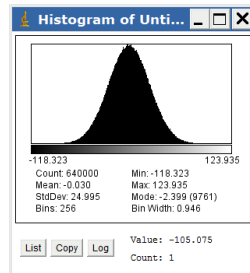


Figure 157: The measured skewness is 0.001

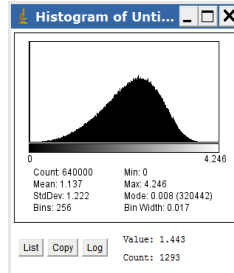


Figure 158: The measured skewness is 0.345

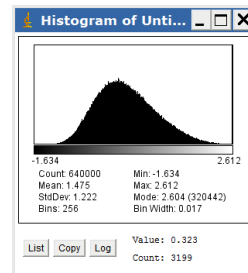


Figure 159: The measured skewness is -0.345

Kurtosis:

A measure of the "peakedness" of the distribution of the gray values around the mean within the roi. Higher kurtosis means more of the variance is due to infrequent extreme deviations, as opposed to frequent modestly-sized deviations.

%Area:

The percentage of the surface that has an intensity value above the min-threshold or above 0 if no min-threshold is set. Loosely speaking this is the percentage of pixels within the object that do not belong to holes.

RawIntDen:

The integrated density is the sum of the gray-values of all pixels within the roi. The raw pixel values are used even if a calibration of gray-values is set for the image.

Slice:

The number of the slice of the stack on which the particle has been detected.

FeretX:

FeretX and FeretY give the starting point for the Feret diameter. Together with the FeretAngle and the length of the Feret diameter they allow to draw the Feret diameter.

FeretY:

See FeretX

FeretAngle:

The angle of the Feret diameter with the x-axis of the image (0-180).

MinFeret:

This is a measure of the particle's width. It is called the minimum caliper diameter as well. It is defined as the shortest distance between two parallel planes touching the particle on opposite sites, for any orientation of the particle.

AR:

The aspect ratio of the particle. This is the length of the major axis divided by the length of the minor axis of the fitted ellipse.

Round:

The roundness of the particle, defined as: $4 \cdot \text{area} / \pi \cdot (\text{major axis})^2$. The roundness is 1 for a circle and approaches 0 for very elongated objects. The major axis of a circle is $p = 2r$ and the area is $a = \pi r^2$.

Solidity:

The area of the particle divided by the area of the convex hull of the particle. You can see the convex hull that is used by creating it from a selection using *Edit>Selection>Convex Hull*.

6.3 Computed thresholds

Open the *plant.tif* image again. We will now have a look at the automatic threshold operations. Open the *threshold-adjuster* from *Image>Adjust>Threshold (shift+t)*. You can select different threshold methods on the left list-button. Try the *Default* method, the *MaxEntropy-threshold*, the *Mean-threshold* and the *Otsu-threshold*. Which one gives the best result? Try them on the *root2.tif* image. Which one gives the best result for this image?

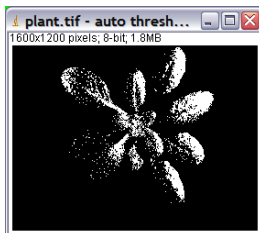


Figure 160: Result of the auto-threshold.

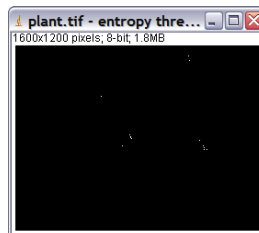


Figure 161: Result of the entropy-threshold.

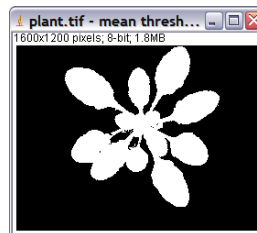


Figure 162: Result of the mean-threshold.

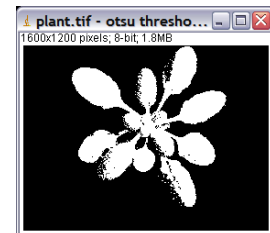


Figure 163: Result of the Otsu-threshold.

The auto-threshold (Default) separates the histogram in a way that the threshold value equals half of the sum, of the average background and the average object intensity. In other words it is trying to find a threshold in a way that half of the average intensity belongs to background and half of it to objects.

The max-entropy threshold searches a threshold value for which the inter class entropy is maximal.

The Otsu-threshold uses a threshold value for which the inter class variance is maximal. The mean threshold uses the mean intensity of the image as threshold value.

6.4 Segmentation of more than 2 object classes

We can have more than two classes for background and objects. We could for example have background, dark objects and light objects.

Install the ij-plugins-toolkit from http://sourceforge.net/projects/ij-plugins/files/ij-plugins_toolkit/. Restart ImageJ or call *Help>Refresh Menus*. Go to *Plugins>Segmentation>k-means-clustering* and apply it to the plant image, using 4 classes. The result is an image with the intensities 1, 2, 3 and 4. Pixels with a similar intensity are marked with the same number. We could now multiply the image with 60 and then use the threshold adjuster to create a mask from each of the numbers. Two binary masks can be combined by using *Process>Image Calculator>And*.

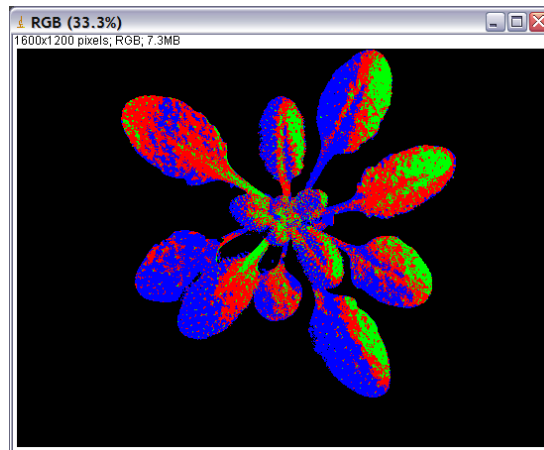


Figure 164: Image segmented into four classes: 1. background - black, 2. dark areas - blue, 3. brighter areas - red, 4. brightest areas - green.

The k-means clustering works for color images as well. Try it on the image *flamingo.png* and on one of the images from the plant robot folder. Use a bigger number of classes (about 15) on the later.

The k-means-clustering interprets the intensity values in n-colors as coordinates in an n-dimensional space. Each cluster is represented by its centroid. Centroids are initially randomly set and optimized afterwards. Pixels are grouped by their proximity to a cluster's centroid.

6.5 Watershed segmentation

Open the image *two-circles.tif*. Apply a threshold. Now there are two objects touching each other and we want them to be separated. This can be done using a binary watershed. Apply it from *Process > Binary > Watershed*.

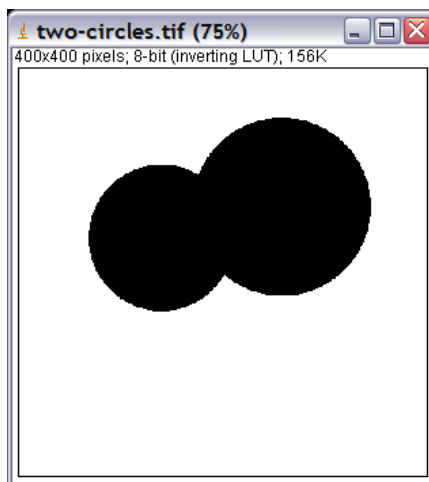


Figure 165: The thresholded image contains two objects touching each other.

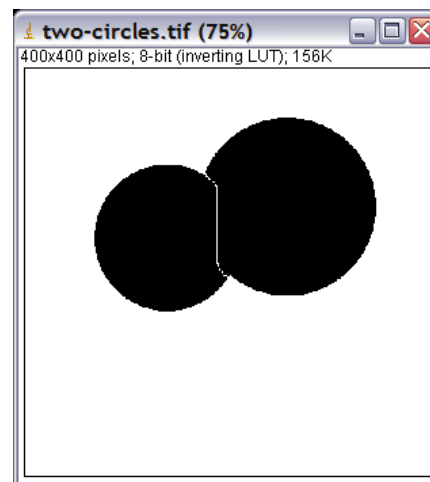


Figure 166: The binary watershed algorithm separated the two objects.

Starting from the binary image a distance map is computed. This gives a grayscale image where the intensities represent the distance from the border for each object. You can create the distance map from the mask image with *Process > Binary > Distance Map*. Imagine the

intensity values as height above the ground. This way the image becomes a landscape with mountains in places of high intensities and valleys in places of low intensities. Apply *Analyze>Surface Plot* or use *Interactive 3D surface plot* plugin (*Plugins>3d*) to see the image this way. The watershed slowly fills the image with water starting from the level 0 and going up one level in each step. In the beginning some separated basins will be created. Whenever the rising water unites two basins a dam is build and remembered.

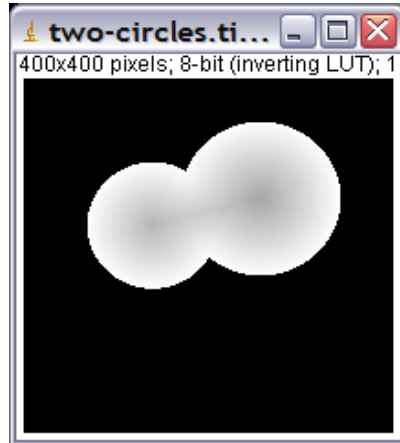


Figure 167: The distance map, pixels within the object are darker, the further away from the border they are.

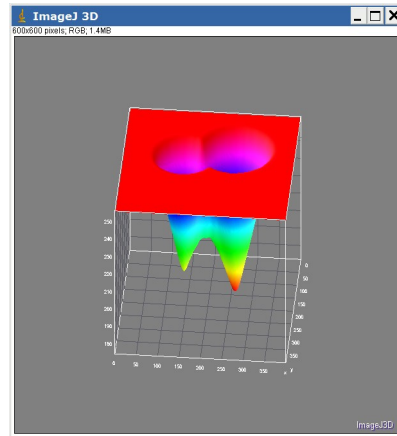


Figure 168: The distance map as a surface plot.

The watershed can be used with a grayscale image directly. To try this we need to install the watershed plugin from <http://imagej.nih.gov/ij/plugins/watershed.html>. Drag the *.jar* file onto the ImageJ launcher window and save it into the *plugins* folder. It will add the *Watershed Algorithm* command to the menu *Plugins>Filters*. Now let's try to segment the image *compartments-1.jpg* with the help of the watershed. Open the image. Before starting the plugin from *Plugins>Filters>Watershed Algorithm*, use a smoothing filter with a high value for the smoothing and invert the result image so that the basins are dark, using *Edit>Invert* or *shift+I* on the image.

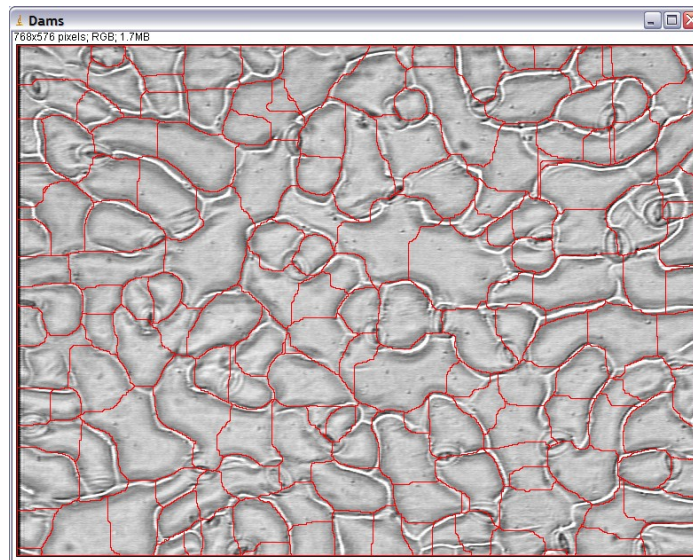


Figure 169: The red lines represent the dams found by the watershed. The image is slightly oversegmented.

Can you get a better result by applying an edge enhancing filter? Try *Process>Find Edges* or *shift+f* on the image before using the watershed.

7. Image types and formats

7.1 Spatial resolution

The optical resolution is defined as the shortest distance between two points on a specimen that can still be distinguished by the observer or camera system. For a microscope the resolution depends on the wavelength and the numerical aperture of the objective. The resolution is the wavelength divided by 2 times the numerical aperture. Smaller wavelength and higher numerical aperture yield higher resolution.

The pixel size in an image taken with a microscope and a ccd camera is the product of the size of a single ccd element multiplied by the binning used and divided by the overall magnification factor.

The more pixels an image has, the higher can be the resolution, though the number of pixels alone doesn't tell anything about the actual resolution. The more pixels an image has, the larger will be the file size in the same time. By how much the file size grows with the number of pixels, depends on the bit-depth of the image. Eight-bit images store intensity values between 0 and 255. Each pixel is stored in 8 bits or one byte. If the image has 400x400 pixels the size will be $400 \times 400 = 160000$ bytes or about 156kb (1kb = 1024b). If we double the size of the image to 800x800 pixels the file size will be four times as large, i.e. 625kb. If we take four times the pixels in x and y direction the file size will be 16 times as large.

7.2 Image types

In ImageJ there are the following basic image types available:

- 8 bit grayscale images (values from 0 to 255)
- 16 bit grayscale images (values from 0 to 65535)
- 32 bit floating point number images (uses signed floating point numbers)

- 8-bit color (values from 0 to 255), an 8 bit image with a lookup table
- RGB Color (3 channels of values from 0 to 255, bit depth 24)

What happens if we convert an 8-bit grayscale image into a 16-bit grayscale image? Try it! Open the image *cells2.tif* and look at the histogram (press *h* on the image). Then convert it to 16bit using *Image>Type>16 bit*. Display the histogram again. As you see the intensity values have not been changed.

7.2.1 Conversion from 16 to 8 bit

Open the image *PAX_Noc30.tif* now. Display the histogram. This is a 16 bit image containing values from 1000 to 35678. What will happen when we convert it to 8bit? Use *Image>Type>8 bit*. The values have been scaled down from the min/max values 1000 and 35678 to 0 and 255.

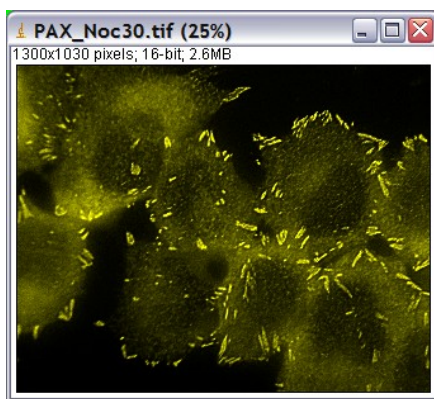


Figure 170: The 16-bit images contains intensity values from 1000 to 35678.

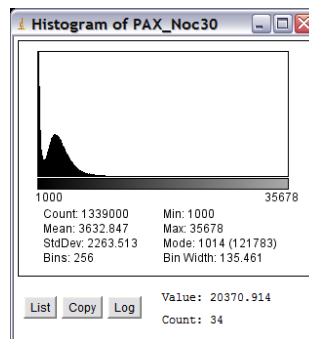


Figure 171: The histogram of the 16-bit image.

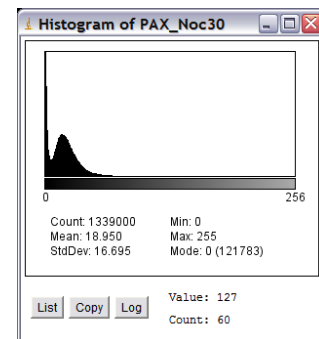


Figure 172: The histogram after conversion to 8-bit.

In *Edit>Options>Conversions* you can turn the scaling off. However then all values above 255 will simply be truncated when converting to 8 bit. The result image will be all white, since the smallest value in the original image is 1000.

Compare the histograms of the results of

- Converting the 16 bit image to 8 bit
- Use *Process>Enhance Contrast>Normalize* on the 16 bit image before converting to 8 bit.
- Convert the 16 bit image to 8 bit and do the normalization

7.2.2 RGB Images

Open an image from the plant robot folder. For color images there is a profile tool as well. Draw a line across the plant and the earth and call *Plugins>Graphics>RGB Profile Plot*.



Figure 173: An RGB-color image.

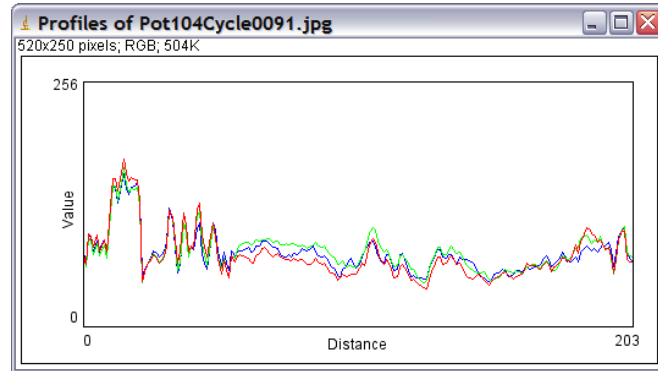


Figure 174: The color profile along the line.

An RGB image can be split into its three components with *Image>Color>Split Channels* and two or three 8-bit grayscale images can be combined into an RGB color image using *Image>Color>Merge Channels...* (make sure *Create Composite* is not selected). Besides this, an RGB image can be converted into a RGB stack or into a HSB stack using the type menu. HSB stands for hue, saturation and brightness.

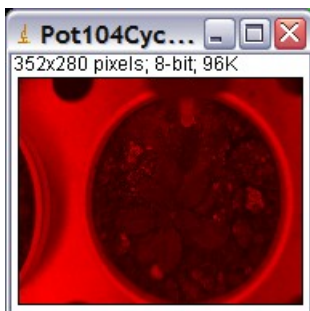


Figure 175: Red component of the image.

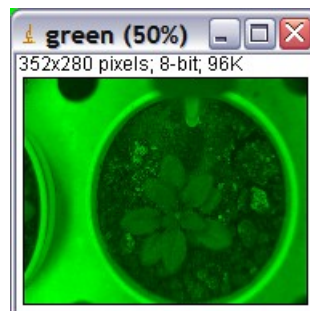


Figure 176: Green component of the image.

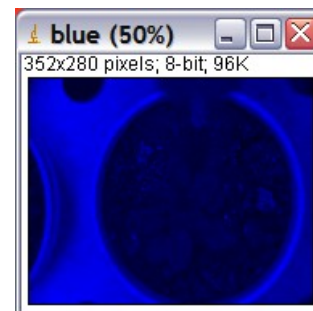


Figure 177: Blue component of the image.

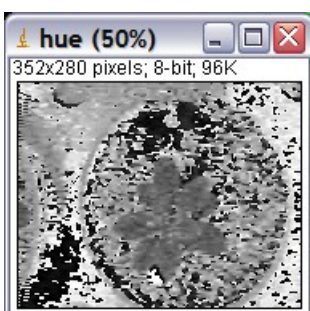


Figure 178: Hue component of the image.

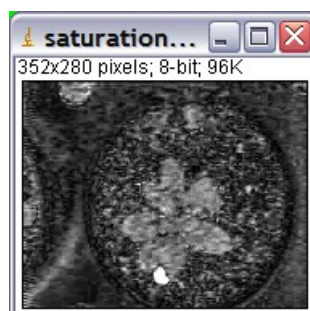


Figure 179: Saturation component of the image.

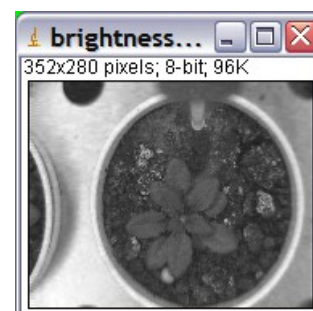


Figure 180: Brightness component of the image.

An RGB image can be converted into an indexed color image (*Image>8 bit color*). You can choose the number of colors that will be used and a lookup table will be created accordingly. Create two images of same size and draw a white rectangle on a dark ground into the two images in a way that the two rectangles are overlapping when the images are combined. Apply rgb merge and put the images into the red and into the green channels. What will be the color of the overlapping region?

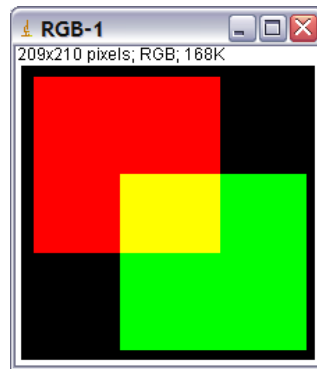


Figure 181: Green and red mix to yellow in an additive color model.

7.2.3 File Formats

Open an image and save it as jpg, using *File>SaveAs>Jpeg*. Open the jpg file, and calculate the difference between the jpg version and the original image. What has happened? If you look at the file sizes you'll see that the jpg version is smaller than the tiff version. Jpg is a compressed format. There are lossless and not lossless compressions. Jpg uses the first scheme. Some information is sacrificed in order to achieve a smaller image in which the difference is not too much visible for human beings. This is similar to mp3 audio compression.

8. Image Analysis

For our purpose we can define the term image analysis in the following way:

- *Image analysis is the extraction of meaningful information from digital images by means of digital image processing techniques.*

8.1 Counting objects

8.1.1 Counting objects manually

Open the image *nuclei.tif*. How many nuclei are there in the image? The cell counter plugin, written by Kurt De Vos helps to count objects manually. Open the cell counter from *Plugins>Particle Analysis>Cell Counter*. You first have to press the initialize button to activate the cell counter on the current image. You can count different classes of objects in the same image. Select a counter type and click on a nucleus in the image. The nucleus will be marked with a number corresponding to the selected class and the count of the class will be incremented by one. If you select delete mode a click deletes the nearest marker of the selected class and decrements the counter of that class.

Once you marked all objects you can use results to get a table with the count for the different classes. You can save the markers independent from the image and load them later on again, either on the same, or on another image. Export image creates an image where the markers are drawn into the image. You can save this image to show which cells you counted, however in contrast to saving the markers, you can't continue the counting later on with the exported image.

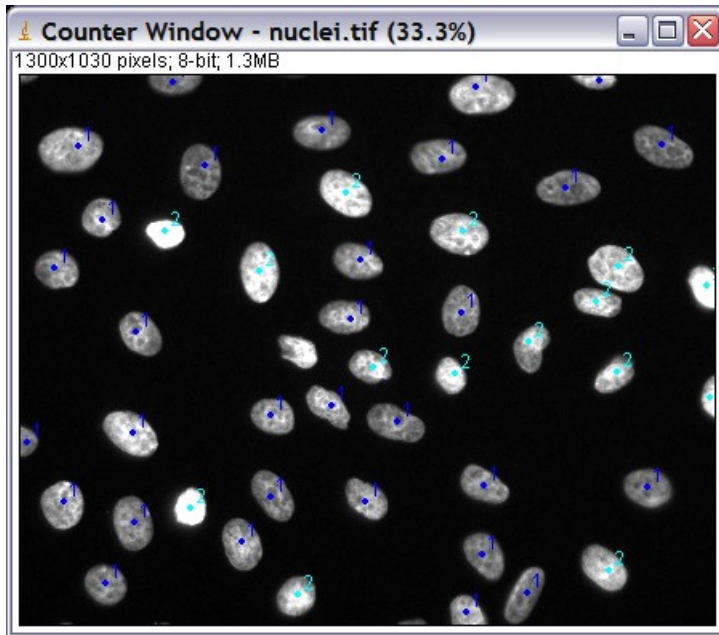


Figure 182: Nuclei marked with the cell counter tool.

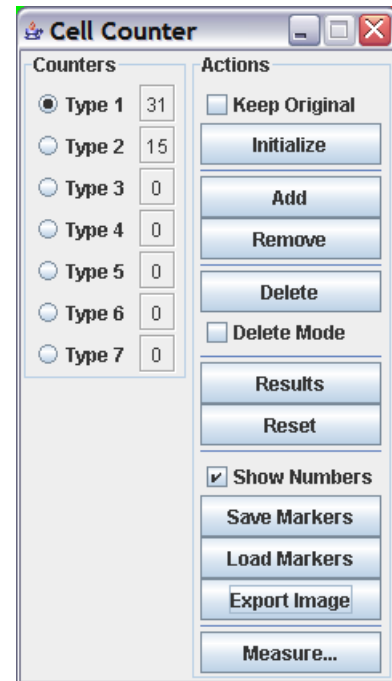


Figure 183: The cell counter tool.

8.1.2 Counting and measuring separated objects automatically

Use the threshold adjuster to create a mask from the nuclei image. Then go to Analyze>Set Measurements and make sure that centroids is selected. Open the Particle Analyzer from Analyze>Analyze Particles or from the toolbox.

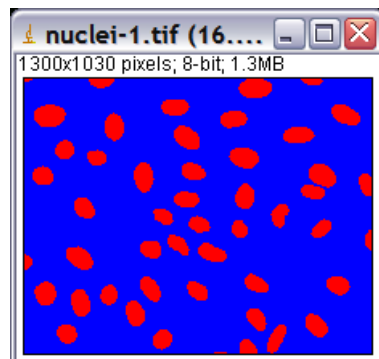


Figure 184: The mask created with the threshold adjuster.

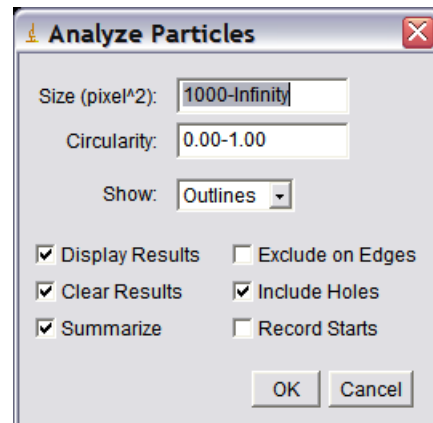


Figure 185: The particle analyzer.

You can exclude particles by their size and circularity. Put in a minimum size of 1000. Select *outlines* in the show field and select *Display Result*, *Summarize* and *Include Holes*. As a result you get a results table with the measurements for each particle, a results table with the summary and an image of the numbered outlines of the particles taken into account. You find the total number of objects in the summary results table.

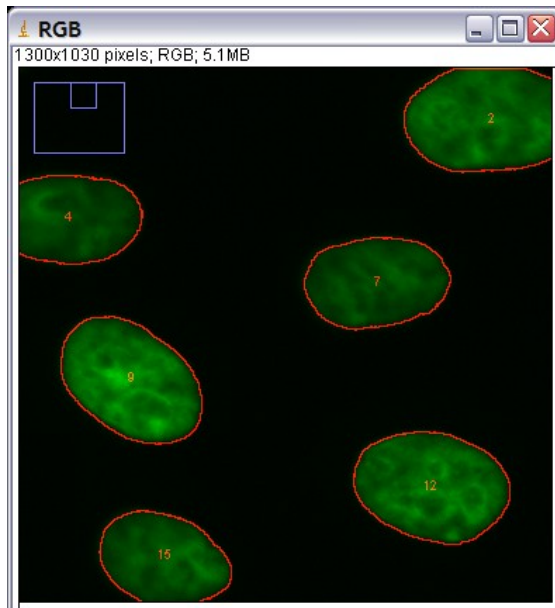


Figure 186: Detail of the overlay of the original image and the outlines of the particles taken into account by the particle analyzer.

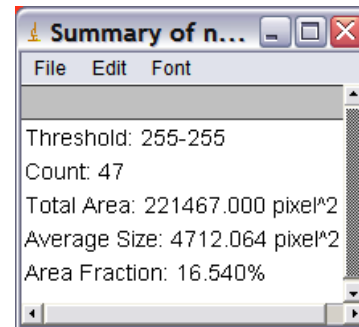


Figure 187: The summary of the measurements includes the total count.

As an alternative to using the outlines to mark which particles have been taken into account, you can use the *Analyze>Label* command to stick a number to each object. This is only possible if you configured the particle analyzer to measure the centroids of particles.

8.1.3 Counting and measuring objects touching each other automatically

8.1.3.1 Statistical approach

It can be difficult to separate objects touching each other. One approach to count them is not to try to separate them, but to measure the total area and to divide the result by the average area of a particle. Open the image *si3_hoechst_1.tif* from the count cells folder. Convert the image to 8 bit and apply a threshold. To estimate the average size of a particle use the wand tool to select singular particles and measure their area. You can use *Analyze>distribution* to get the average of a column of the results table. Then use the particle analyzer to measure the total area of all particles. In case you don't need to exclude small particles that don't represent cells, you can use *Edit>Selection>Create Selection* and measure the selected area.

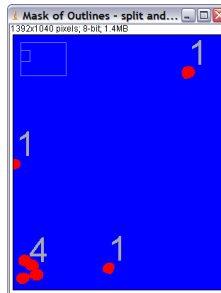


Figure 188: Particles bigger than the average particle size are counted multiple times.

8.1.3.2 Local Threshold

Sometimes unsharp masking can help to separate particles. Try the local threshold plugin from *Plugins>Segmentation>Local Threshold* on the image *hst-0.tif* in the cells in structures folder. The plugin applies a median filter and subtracts the result from the original image. The result image is a 32 bit image. It then lets the user choose a threshold value.

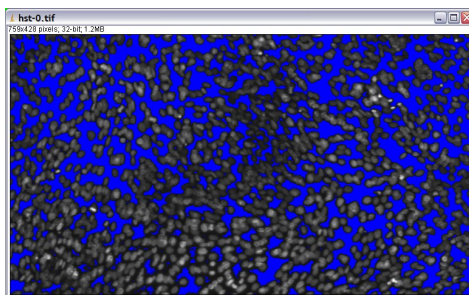


Figure 189: The original image with a threshold.

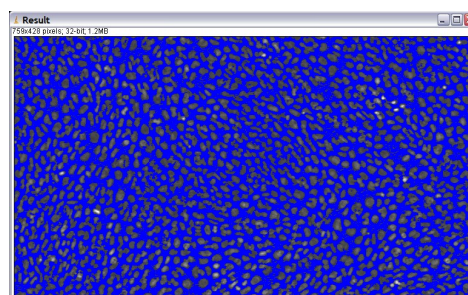


Figure 190: The image after application of the local threshold plugin.

8.1.3.3 Watershed

We can use the greyscale watershed plugin to separate the cells in the *hst-0.tif* image. Invert the image, so that the nuclei are dark and the space around them is bright. Use a small radius for the smoothing filter. Select the show the watershed dams only option. Run the watershed. Invert the watershed image and subtract it from the original *hst-0.tif* image. The borders between the cells are now set to zero and you can use the threshold adjuster to get a mask for the separated cells.

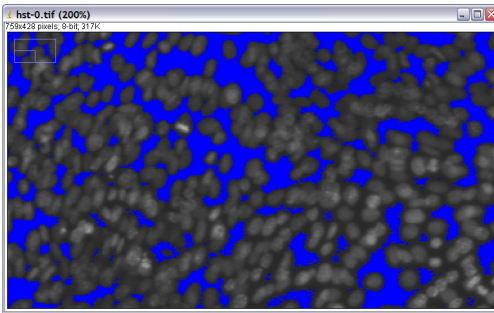


Figure 191: The original image.

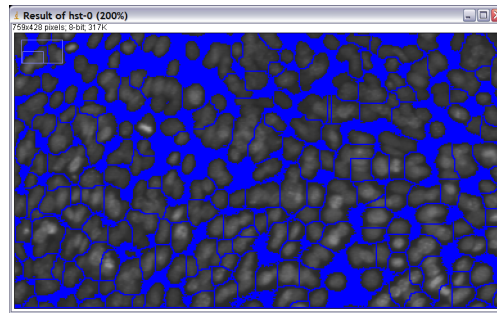


Figure 192: Cells separated by the watershed plugin.

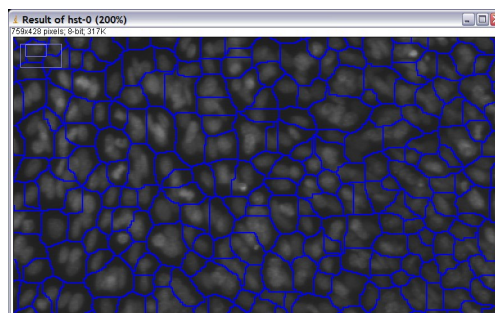


Figure 193: Image with watershed dams.

8.1.3.4 Other methods

Other methods use for example first or second order gradients of the image or mathematical morphology.

8.2 Comparing intensities

Open the image *A4 Rhod 1.tif*. We want to compare the total intensity in the nuclei with the total intensity in the cytoplasm. The image *A4 dapi 1.tif* helps us to find the nuclei. Apply the *hilo* lut to the rhod image and subtract a baseline value, so that the empty background becomes zero. Threshold the dapi image. If there are holes in the mask of the nuclei use *Process>Binary>Fill Holes* to close them. Create a selection from the mask and if necessary modify the selection, so that it only contains the nuclei. Transfer the selection from the mask image to the rhod image using *Edit>Selection>restore* or *CTRL+Shift+e*. Configure the measurements to include the total intensity. And measure the intensity in the selection (*ctrl+m*). Inverse the selection (*Edit>Selection>Make Inverse*) and measure again.

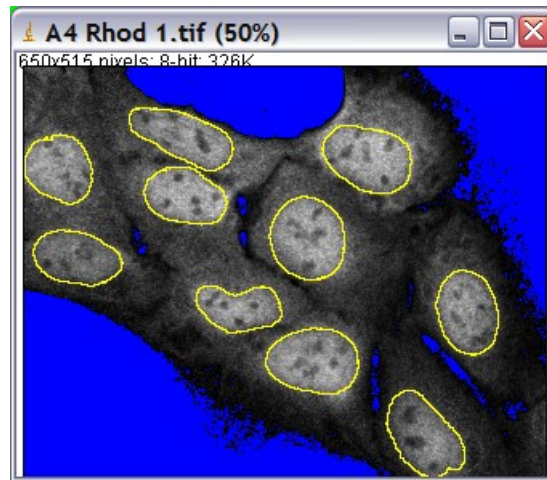


Figure 194: Measuring the intensity in the nuclei and in the cytoplasm.

8.3 Classifying objects

Sometimes you might not only want to separate objects from the background and from each other but also to distinguish between different kinds of objects in the image. One could for example be interested in distinguishing between normal and apoptotic cells. To do so you need to find a set of features that separates the different classes of objects you are looking for. Load the image ovals and triangles and try to separate the ovals from the triangles. In this case the two classes can be distinguished by the feature roundness. Use the particle analyzer to separate the two classes of objects.

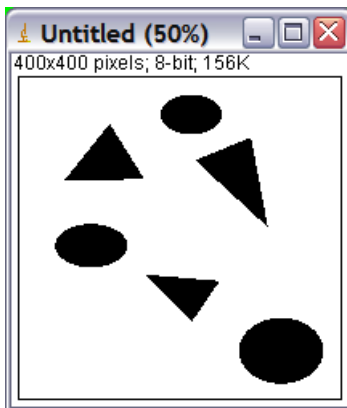


Figure 195: Different kinds of objects in one image.

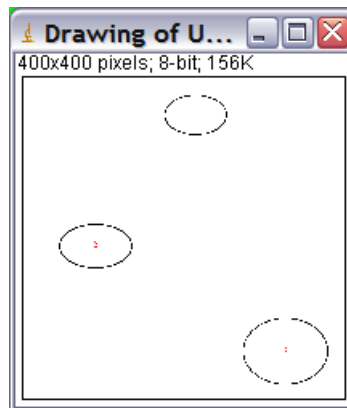


Figure 196: Ovals have a high roundness.

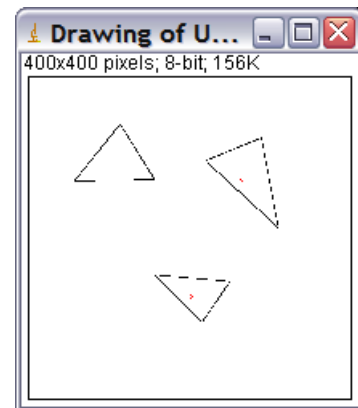


Figure 197: Triangles have a low roundness.

In general more than one feature might be necessary to classify objects. One useful set of features are the moments of objects and values derived from moments of objects. You can use the Moment Calculator plugin (*Plugins>Particle Analysis>Moment Calculator*) to compute them.

Image	Area	Mean	Min	Max	Cutoff	Factor	Mass	xC	yC	xxVar	yyVar	xyVar	xSkew	ySkew	xKurt	yKurt	Orient.	Elong.
[Thu Nov 02 13:07:22 CET 2006]																		
No spatial calibration [results in pixels].																		
No density calibration [uncalibrated results].																		
1 ovals and triangles.tif	2965	0	0	0	0	1	206040	214.500	46.500	948.569	382.025	0	0	0	-1.838	-1.835	0	6.6133221E10
2 ovals and triangles.tif	3824	0	0	0	0	1	264180	90	210	1296.621	464.767	0	0	0	-1.841	-1.838	0	1.82807249E11
3 ovals and triangles.tif	6786	0	0	0	0	1	465120	326.500	341	1763.904	1075.649	0	0	0	-1.840	-1.840	0	2.20324694E11
4 ovals and triangles.tif	3492	0	0	0	0	1	894540	103.756	83.364	1234.360	316.915	-10.672	0.155	0.698	-1.627	-0.309	-0.666	7.5334759E11
5 ovals and triangles.tif	4197	0	0	0	0	1	1471605	260.147	137.592	800.791	1290.682	-76.245	0.398	-0.392	-1.216	-1.249	-81.355	3.87395944E11
6 ovals and triangles.tif	2512	0	0	0	0	1	734910	200.281	280.926	1001.395	301.730	-18.412	0.239	-0.629	-1.515	-0.774	-1.506	3.60758149E11

Figure 198: Features based on moments of the circles (first 3 rows) and triangles (last three rows) .

Further features can for example be calculated with the Particle8 Plus or other morphological plugins from Gabriel Landini (*Plugins>morphology>Particles8 Plus*).

CountCorrect	AspRatio	Circ	Roundness	ArEquivD	PerEquivD	EquivElAr	Compactness	Solidity	Concavity	Convexity	Shape	RFractor	ModRatio	Sphericity	ArBBox
1.4126087	1.5573419	0.833441	0.6296095	60.49207	914.82263	2931.106	0.7934794	0.9869506	38.0	0.9562917	15.077697	0.8311676	0.6296199	0.6296199	3732.0
1.6183733	1.4487293	0.55073977	0.4380048	65.53348	1073.6593	5315.5747	0.6618193	0.9721862	96.5	0.9661557	22.817257	0.8616152	0.4608419	0.40049228	8768.0
1.7893784	1.7893128	0.49463472	0.3534843	71.92034	1293.1339	6422.9863	0.5945454	0.9721225	116.5	0.9552429	25.405355	0.8075261	0.36874095	0.30042994	8178.0
1.4933727	1.6480563	0.81488574	0.5941252	68.81263	1183.7944	3798.1855	0.77079517	0.9877822	46.0	0.9546544	15.421021	0.81514866	0.59377784	0.59377784	4836.0
1.525703	1.7634408	0.5260764	0.36003953	55.411346	767.6043	3798.1855	0.6000329	0.9743434	63.5	0.9735579	23.88697	0.80540156	0.36512804	0.31091702	4836.0
1.7082019	1.2717441	0.8750348	0.77439785	92.03037	2117.3975	6754.4243	0.8799988	0.9895864	70.0	0.9521979	14.360995	0.89577025	0.7745258	0.7745258	8600.0

Figure 199: Further features of the circles (rows 1, 4, 6) and triangles (rows 2, 3, 5).

9. Annotating images

9.1 Scale bar, time stamper and event stamper

Open the image *Pot.tif*. This opens a stack of image which represents a time series. You can cycle through the images with the slider at the bottom of the window. We want to add three annotations to the images: A scale bar indicating the length of a known distance, the time point from the beginning of the series and a marker for images in which the flower appears. To add a scale bar we first have to calibrate the image. We either need to know the pixel size or the length of an object in the image. Let's say that the diameter of the pot is 6cm. Make a line selection from one side of the pot to the opposite side. Then open *Analyze>Set Scale...* Enter the distance 6 and the unit cm and press ok. When the scale is set ImageJ will use for all measurements, i.e. results will now no longer be in pixel but in *cm*.

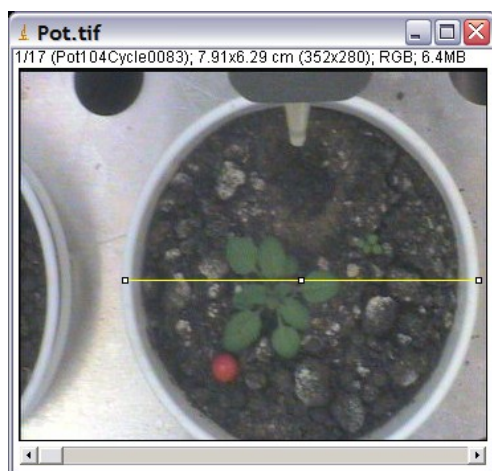


Figure 200: Selecting a known distance.

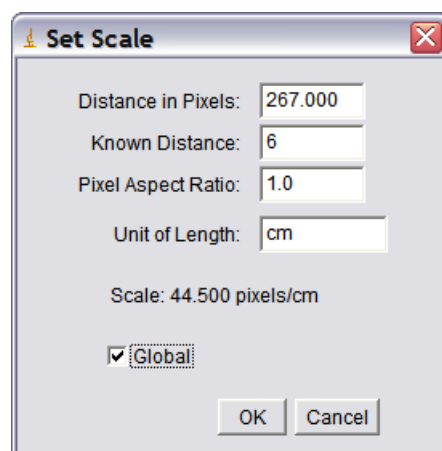


Figure 201: Setting the scale.

Go to *Analyze>Tools>Scale Bar* and put a black scale bar of 2cm length into the lower left corner. Select *Label all Slices* to put the scale bar into all images of the series.

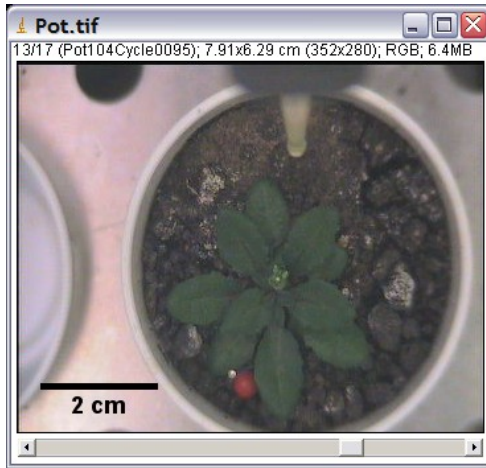


Figure 202: The black line segment has a length of two centimeter.

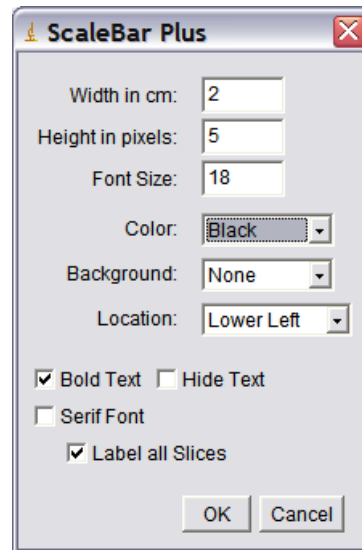


Figure 203: Adding a scale bar to each image of the series.

We will now add a text indicating the time when the image was made. Let's say the image have been made with an interval of 1 day between to images. Make a rectangular selection in the place where the stamp should appear. Double click the pipette icon and set the foreground color to the color you want the stamp to have. Then open the time stamper from *Plugins>Movie>Time Stamper*.

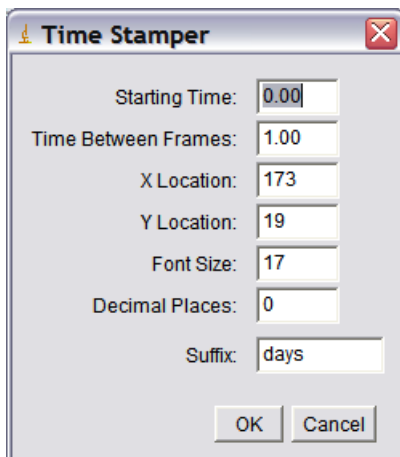


Figure 204: The time stamper dialog.

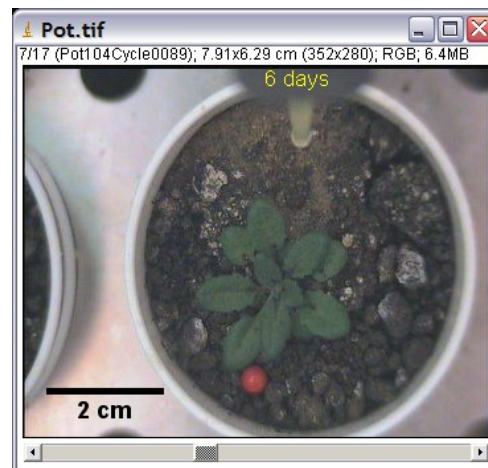


Figure 205: The time stamp has been added to each image of the series.

Finally add a label *blooming* on the slices 14 to 17. This is done in a similar way as adding the time stamp, only that the command *Plugins>Movie>Event Stamper* is used.

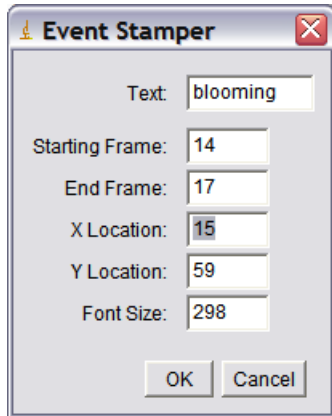


Figure 206: The event stamper dialog.

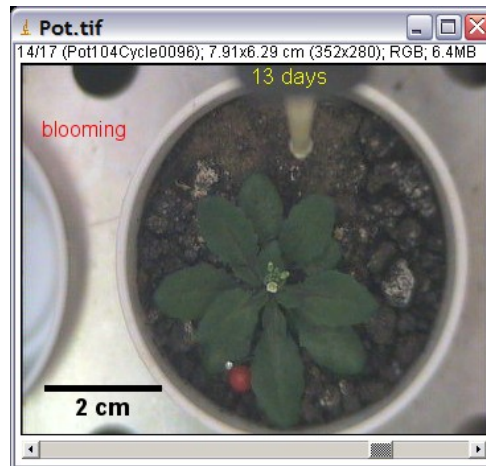


Figure 207: A text has been added to some images of the series.

9.2 Calibration bar

Using a lookup table on a greyscale image, we want to provide a bar that shows which colors represent which intensities, normalized to a range between 0 and 1, in the image.

Open the image *A4 Rhod 1.tif* and apply the lookup table *Rainbow RGB*. There are two ways to achieve our goal. One way is to make a selection in the image and run *Plugin>LUT>Add Ramp*. This will add a ramp with the colors of the lut from the bottom with the value 0 to the top with the value 255. It only works for 8 bit images.

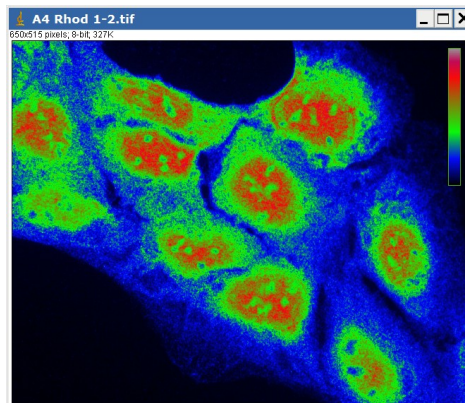


Figure 208: A ramp shows which color represents which intensity.

The other way is to first calibrate the image intensities using *Analyze>Calibrate...* A calibration bar can then be stamped into the image using *Analyze>Tools>Calibration Bar...* To calibrate the image you have to enter a number of pairs of uncalibrated and calibrated values. You can select the form of a curve that will be fitted to the entered data. For a calibrated image, intensity values will be displayed in the calibrated form, followed by the greylevel value in brackets. Measurements will be in calibrated form. Call *Analyze>Calibrate...* and enter the points 0, 0 and 255, 1. Select the function straight line. Then use *Analyze>Tools>Calibration Bar...* stamp the calibration bar into the image.

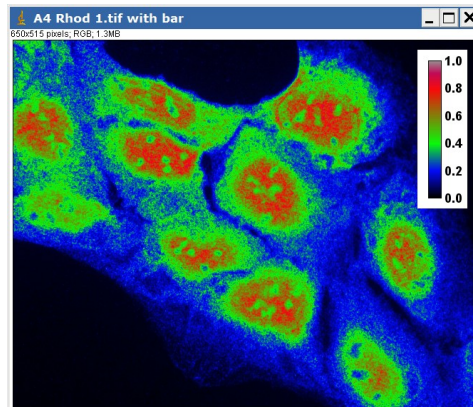


Illustration 209: An image with a calibration bar.

10. Working with multi-dimensional data

In general image data can have multiple dimensions. 3 spacial dimensions, the time dimension and many channels (for example different wavelengths).

10.1 Time series

Open the image *pot.tif*. You can play the image sequence using *Image>Stacks>Start Animation*. You can use the last button in the ImageJ launcher window to switch the current tool set to the stack tools. This will display buttons to:

- animate the stack (click again to stop),
- go to first and last slice,
- go one slice forwards / backwards
- add a slice after the current slice / delete current slice

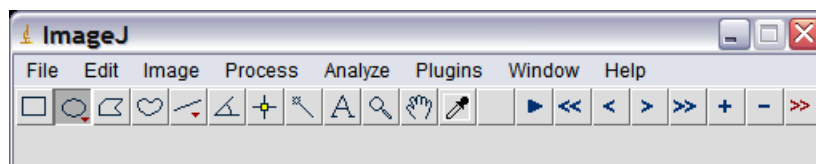


Figure 210: ImageJ with the stack tool set selected.

As you see the camera position is not exactly the same for each image which makes the plant somewhat jump around. Use *Plugins>Stack Shuffling>Align* slices for an automatic stack registration. Choose either rigid body or translation as transformation.

Make a rectangular selection, so that the black borders that have been created by the stack registration are outside of the selection and duplicate the stack (*ctrl+shift+d*).

You can export the series as a movie in one of the following formats:

- animated gif
- avi movie
- quicktime movie

Use *File>Save As* and one of these formats to export the series as a movie. If your image sequence is too big to fit into memory at once, you can still create a movie from it by using the virtual stack opener (*File>Import>Virtual Stack*).

10.2 Measuring velocities

Open the image moving-particles. You see two circles that move from the lower left to the upper left side of the image. We want to measure the lengths of the paths they traveled.

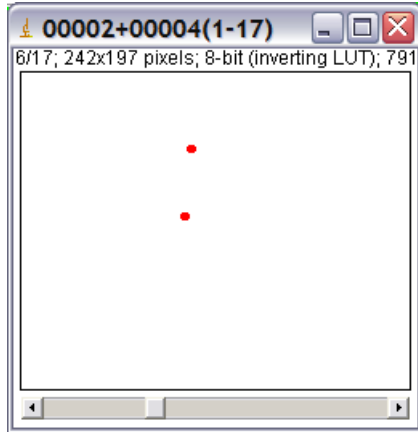


Figure 211: Two moving particles.

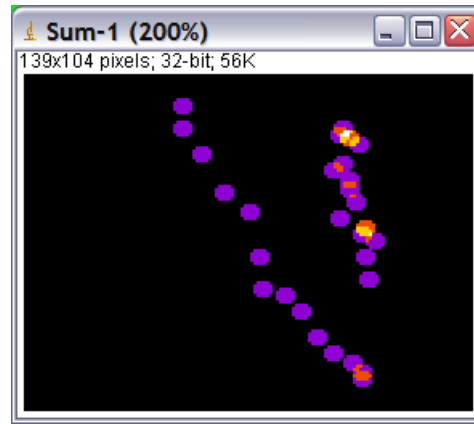


Figure 212: A projection of the time series.

In principal this could be done manually using for example the polygon selection tool. Go to the first slice set the first polygon point to the into the first particle, then go one slice ahead using the '>' key and set the next point and so on. When the path is finished press *ctrl+m* to measure its length.

Another possibility is to create a projection of the stack (*Image>Stack>Z Project...*) and use the polygon selection tool on it.

The *Mtrack2* plugin measures the track length automatically. Open it from *Plugins>Particle Analysis>Mtrack2*.

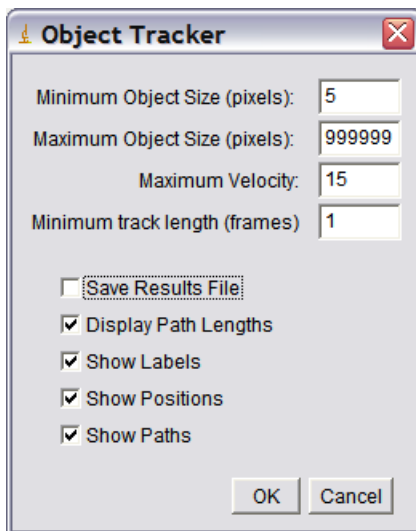


Figure 213: The MTrack2 plugin.

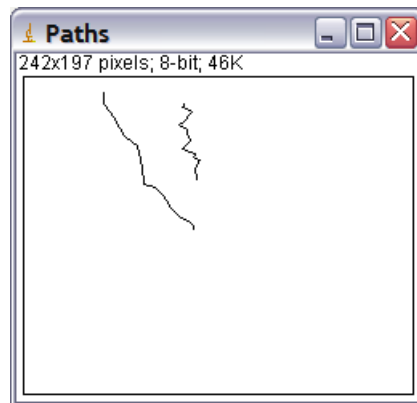
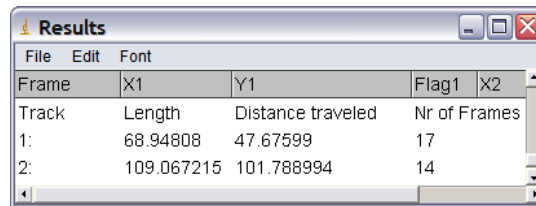


Figure 214: The paths found by the MTrack2 plugin.



Frame	X1	Y1	Flag1	X2
Track	Length	Distance traveled	Nr of Frames	
1:	68.94808	47.67599	17	
2:	109.067215	101.788994	14	

Figure 215: The plugin answers the distance travelled by the particles (length) and the linear distance from the start point to the end point of the track.

11. Biological applications

11.1 Colocalization analysis

11.1.1 Visualizing colocalization using overlays

In principle an overlay of the two channels, using two appropriate lookup-tables, like for example red and green or cyan and magenta, should give as a hint if two fluochromes are in the same place. In the red/green case places in which the red and green intensities are close to each other will appear yellow in the image. Another possibility is to use magenta, green and blue.

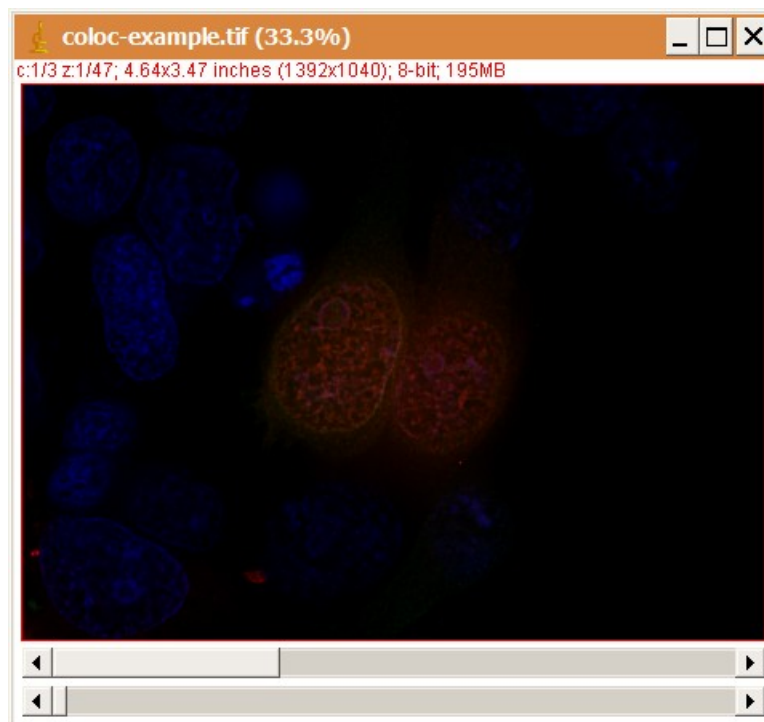


Figure 216: The overlay of the channels. The green channel has lower intensities and is invisible.

But there are some problems with this approach. Yellow spots will only become visible if the intensities in both channels are similar, both channels must have similar histograms. Histogram normalization can be used but may suggest false results since intensities don't represent the real quantities of molecules anymore.

Image processing and analysis with ImageJ

Open the images *coloc-Rhod*, *coloc-GFP* and *coloc-Hoechst* from the *coloc* folder. Create an overlay of the channels that shows the colocalization between *gfp* and *rhodamine*. Instead of an additive overlay of channels the difference can be used. Apply magenta and cyan lookup tables, convert the images to RGB and calculate the difference using the image calculator. The same result can be achieved using the Colour merge command from the menu *Plugins>Colour functions*. Check *Use difference operator* in the dialog.

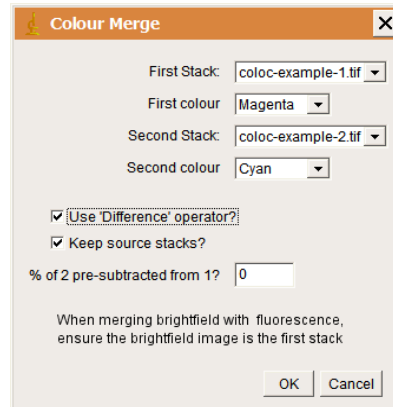


Figure 217: The color merge dialog.

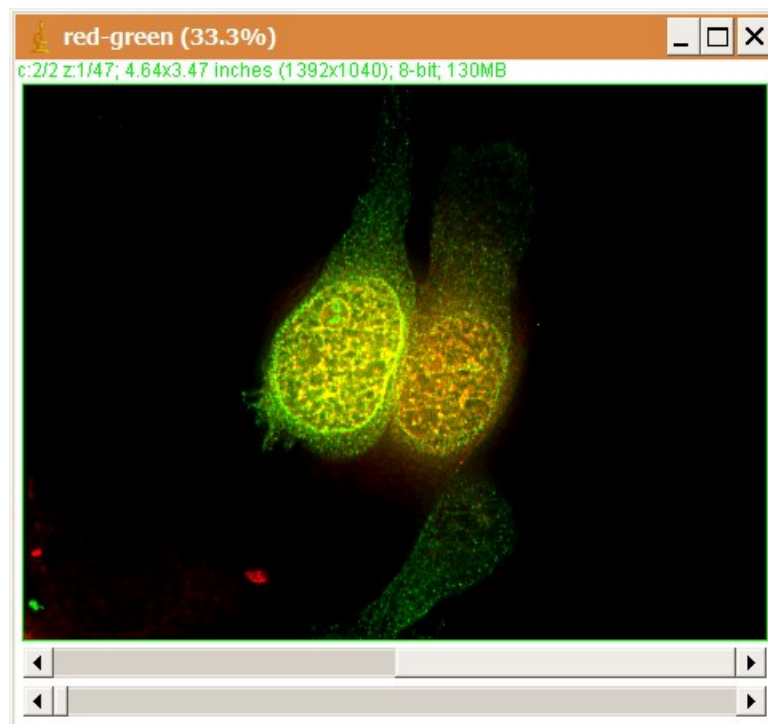


Figure 218: Overlay of the red and green channel after histogram normalization.

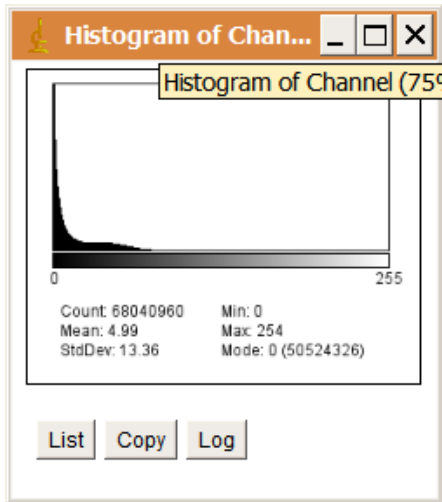


Figure 219: The histogram of the red channel.

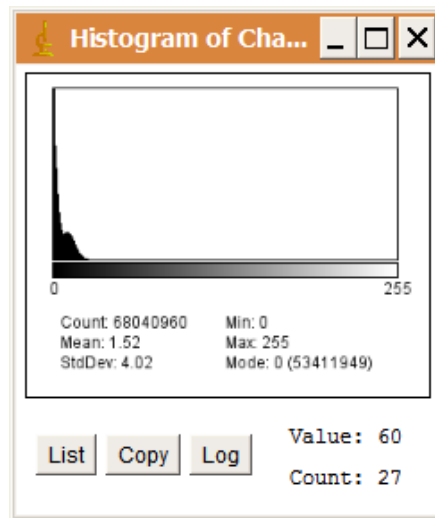


Figure 220: The histogram of the green channel.

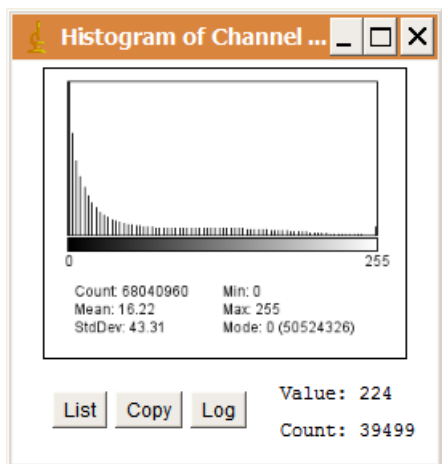


Figure 221: The histogram of the red channel after normalization.

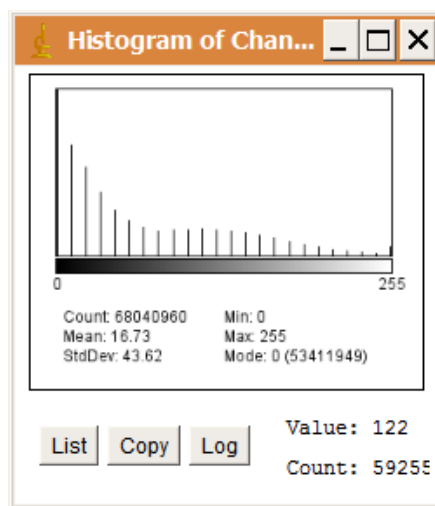


Figure 222: The histogram of the green channel after normalization.

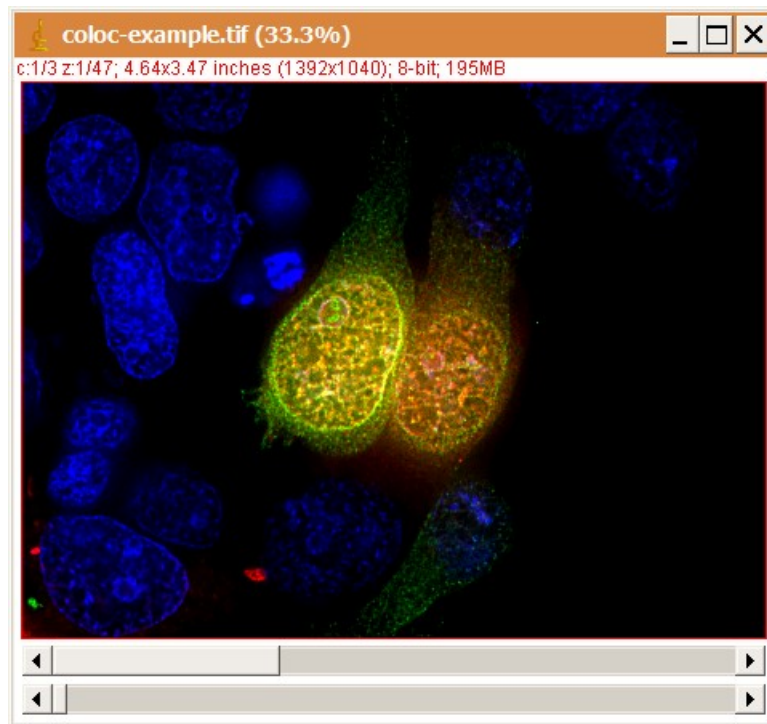


Figure 223: Original overlay with auto-display-adjustment.

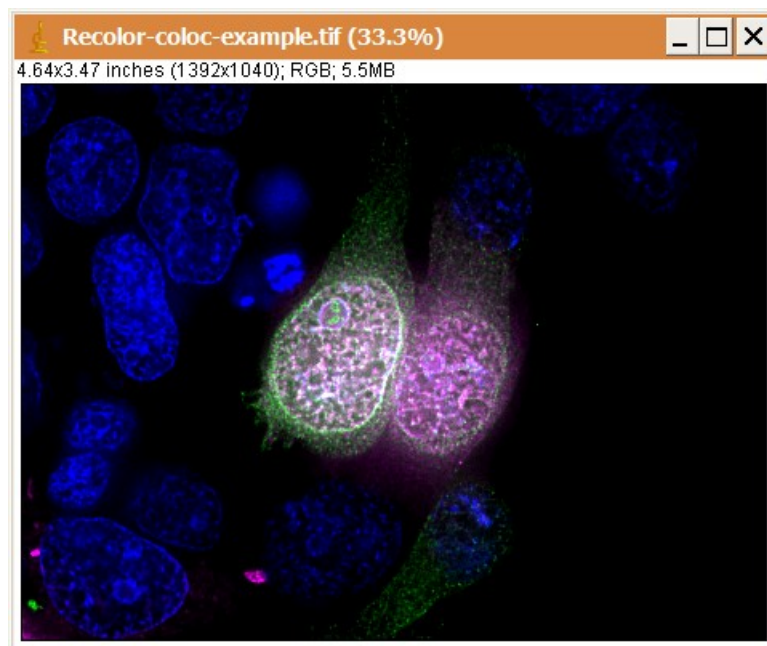


Figure 224: Original overlay with auto-display adjustment in Magenta/Green/Blue.

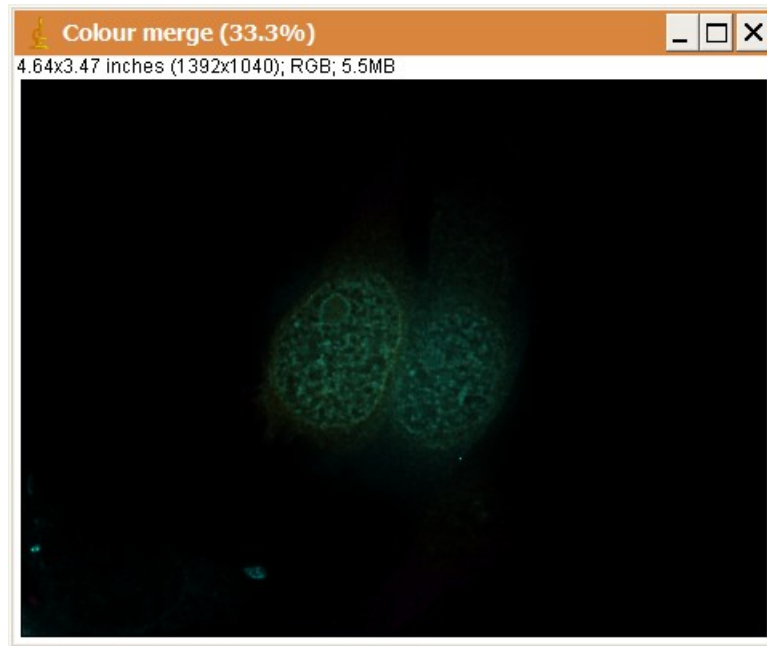


Figure 225: Color merge with difference of the original channels.

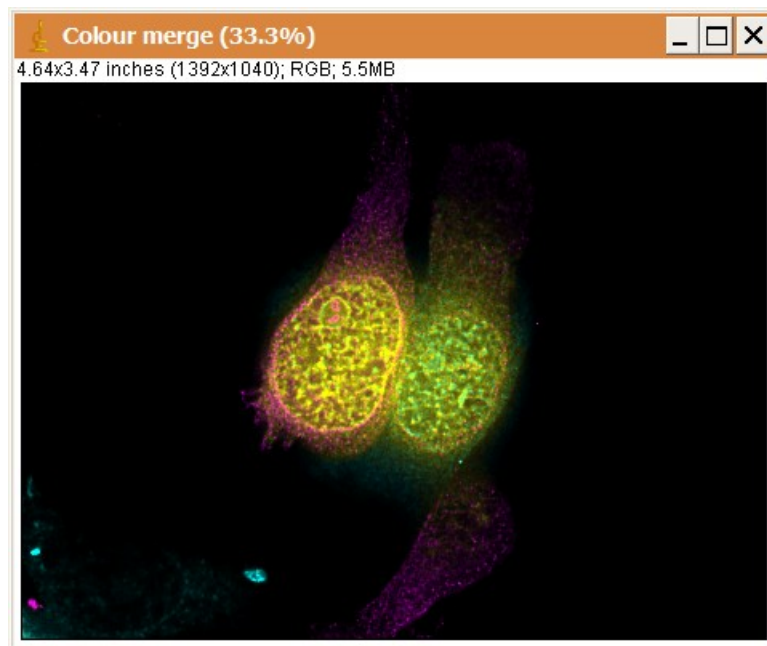


Figure 226: Color merge with difference after histogram normalization.

11.1.2 Quantification of colocalization

11.1.2.1 Intensity correlation coefficient based methods

In this approach a statistical analysis of the correlation of the intensity values in the two channels is performed. This is mostly done by using correlation coefficients that measuring

the strength of the linear relationship between two variables, the intensities in the two channels.

11.1.2.2 Pearson's coefficient and scatter plot

A fluorogram or scatter plot can help to judge if colocalization is present or not. In a scatter plot the x-axis represents the intensity values of the pixels in one channel and the y-axis the intensities in a second channel. The pixels of the two images will be compared one by one. If the intensity of pixel (0,0) is 10 in the first image and 15 in the second image, the point 10,15 will be marked in the scatter plot. Sometimes the frequency of the intensity pair is coded by the color or brightness in the scatter plot. A line will be fitted through the points in the scatter plot. A positive slope indicates a correlation, a zero slope indicates that the images are uncorrelated and a negative slope indicates a negative correlation (a mutual exclusion). The Pearson's Coefficient is a measure of the quality of the linear fit. It is obtained by dividing the covariance of the two variables by the product of their standard deviations. Mark that the thickness of the point cloud and the direction of the line have an influence on the PC, but not the slope of the line. The PC will be near 1 for a positive correlation, near zero in case of no correlation and near -1 for a negative correlation.

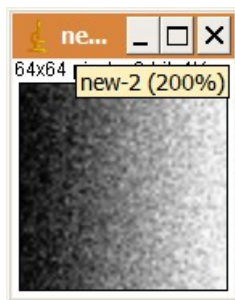


Figure 227: A noisy image.

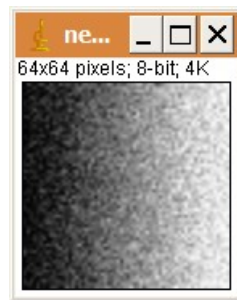


Figure 228: Another similar noisy image.

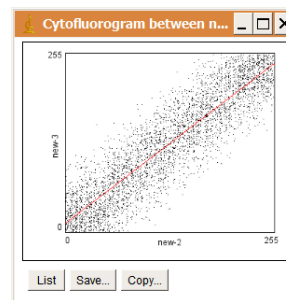


Figure 229: A scatter plot of correlated images

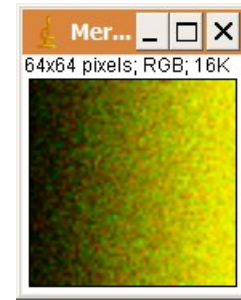


Figure 230: The color overlay of the input images.

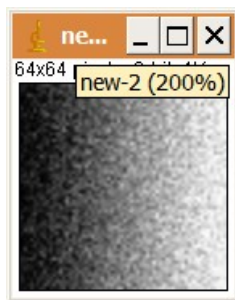


Figure 231: A noisy image.

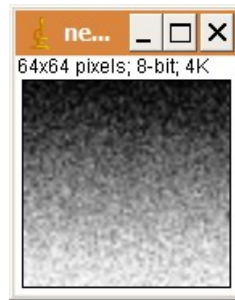


Figure 232: Another similar noisy image, rotated by 90°.

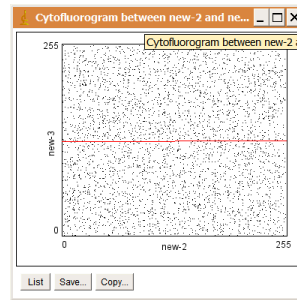


Figure 233: Scatter plot of uncorrelated images

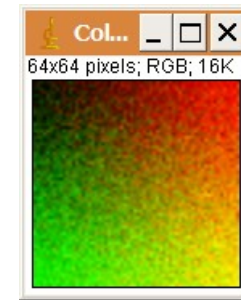


Figure 234: The color overlay of the input images.

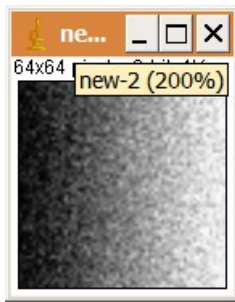


Figure 235: A noisy image.

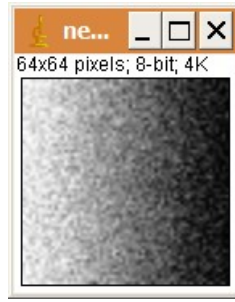


Figure 236: Another similar noisy image, rotated by 90°.

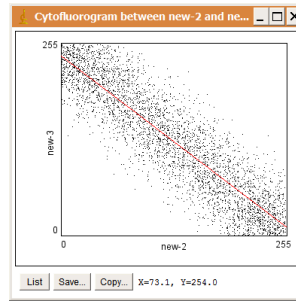


Figure 237: Scatter plot of anti-correlated images

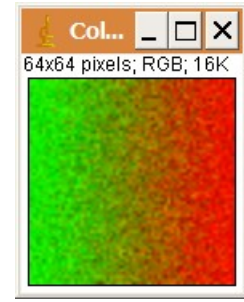


Figure 238: The color overlay of the input images.

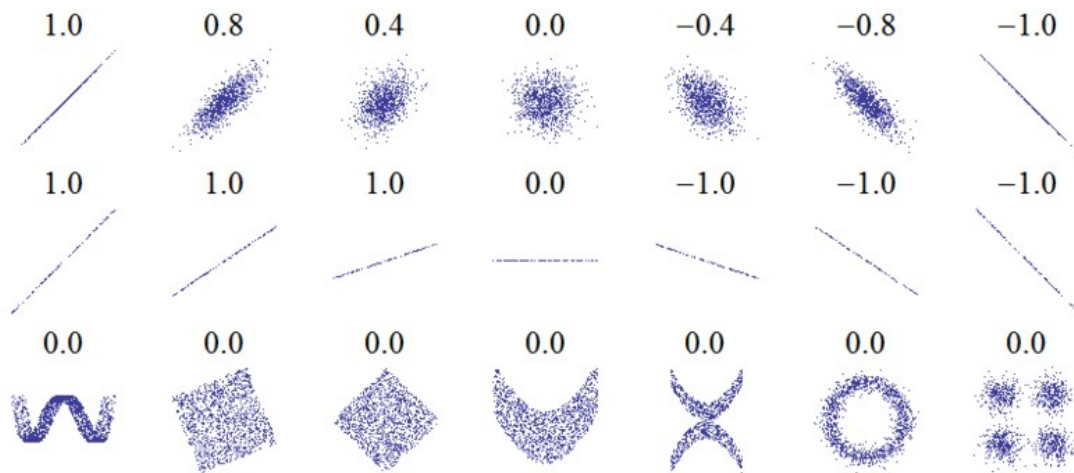


Figure 239: Pearson's coefficients for different scatter plots.

Open the images *coloc-GFP* and *coloc-Hoechst* from the folder *12 - coloc*. Use the *JACoP* plugin to calculate the scatter plot. Run the plugin from the *Plugins* menu, select the images in the dialog, select *cytofluogram* and *Add the zero line*. The scatter plot will be created and the fitted line will be shown. The Pearson's Coefficient will be written to the log window as *Correlation Coefficient*. If you want to calculate the Pearson's Coefficient without the scatter plot, check *Pearson's coefficient* instead of *cytofluogram* in the dialog.

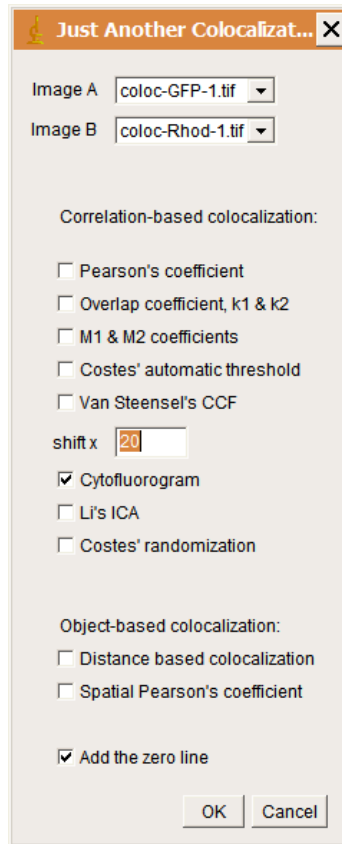


Figure 240: The dialog of the JACoP colocalization plugin.

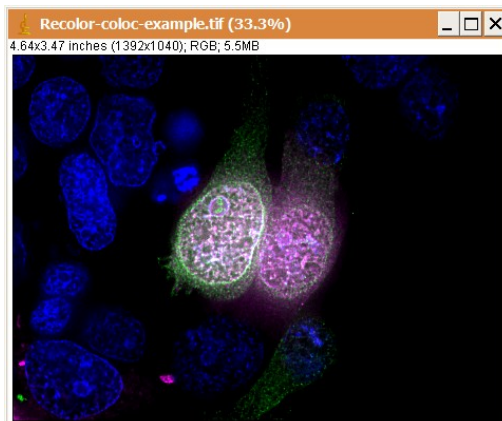


Figure 241: The overlay of the two input images.

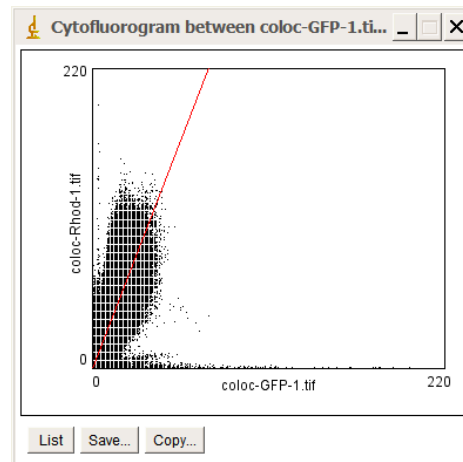


Figure 242: The scatter plot and the fitted line calculated by the JACoP plugin.

Scatter plot and Pearson's coefficient indicate colocalization, especially when it is complete. However they rarely discriminate differences between partial colocalization. Mid-range coefficients between -0.5 and 0.5 do not allow conclusions to be drawn. The result is dependent on noise and background or the threshold used to suppress the background.



Image processing and analysis with ImageJ

Bleedthrough of the fluorescents can lead to false results as well. Finally there could be a non linear relationship between the two signals, that could not be detected using this method.