

An Introduction to Image Analysis in Microscopy

CECAD Imaging Facility Course, v1.1

Nikolay Kladt

March 4, 2015

Imaging Facility
Cluster of Excellence – Cellular Stress Responses in
Aging-Associated Diseases
University of Cologne

Contents

1	Publishing original microscopy data	5
1.1	Pixels, spatial resolution and sampling frequency	7
1.1.1	Resampling Images	8
1.2	Bit-Depth & Computer Number Representations	12
1.2.1	Choice of Bitdepth	16
1.3	Image Dimensions	16
1.3.1	3D stacks	17

Module 1

Publishing original microscopy data

In this module, we will work through some necessary basics of image processing and explore common ways to visualize microscopy data. After this module, you should feel comfortable working with your images in Fiji and be able to use this knowledge to prepare your data for publication.

Digital images are ubiquitous and nearly everybody is used to taking pictures with a digital camera or smart-phone and also storing and processing these images. The same way that digital imaging has replaced analog cameras in our daily lives, light microscopes have transformed to digital imaging systems¹. Even the most simple light microscopes are usually equipped with a digital camera and a screen to display, take and store images - also allowing quick and simple image processing tasks without the need for additional software (e.g. brightness & contrast adjustments, white balancing). More complex light microscopes, such as confocal laser scanning microscopes (CLSM), already require sophisticated software interfaces to set up imaging parameters and visualize the results. Finally, there are also types of microscopes that depend on digital image processing (e.g. stochastic optical reconstruction microscope, STORM or single plane illumination microscopy, SPIM).

As a result of these advancements, modern digital microscopy offers more than just 'pretty pictures' - there are many imaging techniques that rely on computational analysis as well as methods for image enhancement, restoration and quantitative analysis. Modern microscopy cannot be understood without the basics of digital images and digital image processing. This knowledge is not only essential to make full use of the capabilities, but also to understand limitations – a very important prerequisite to maintain scientific validity.

¹In this course, we do not discuss the technologies that actually generate digital images in microscopes. If you want to learn about various microscopy techniques, we recommend looking at the online iBiology Microscopy Course (www.ibiology.org/ibioeducation/taking-courses/ibiology-microscopy-course.html, 25-02-2015)

Raster and vector images

Typically, most pictures you work with (scanned documents, photos taken with a digital camera, pictures found on the web) are raster graphics (see Fig. 1.1). This means that these images are made up of a grid of *pixels* (picture elements). Each pixel has a value that represents some property of the image (usually brightness). The range of this value is called *bit-depth* and determines possible values at each pixel. Sometimes, the term *bitmap image* is used for a raster image. For example, the image shown in figure 1.1 illustrates a raster image with 420 x 420 pixels and a bit-depth of 1 byte. Therefore, we have a total of 176400 pixels, multiplied with 1 byte taking up a total of 172 kilobytes disc space. As you can see, the number of pixels and the bit-depth affect the required disc space. Additionally, we cannot easily change the size of the image; enlarging causes the image to look 'pixelated' or we have to use interpolation techniques to estimate the value of added pixels. Decreasing the number of pixels in the image might result in a loss of image features. There are plenty of commercial and open-source programs to work with raster images; representative software packages are Adobe Photoshop (Adobe Inc., San Jose) and GIMP (GNU Image Manipulation Program, The GIMP team, <http://gimp.org>).

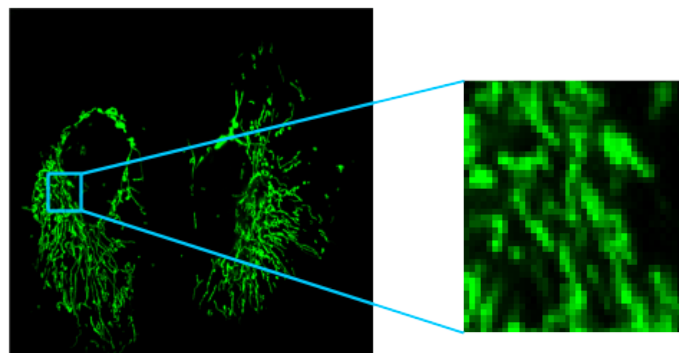


Figure 1.1: Fluorescent labeled mitochondria in HeLa cells. This image shows that raster graphics are made up of pixels that become clearly visible when zoomed.

In contrast, vector graphics are not made up of a bitmap. Instead, they are based on geometrical shapes such as lines, curves, polygons or other shapes that are based on mathematical expressions. Therefore, these images can be scaled without a loss in image quality and typically use less disc space than raster images. Especially when the scaling of lines, text or other image content might change, it makes sense to use vector graphics (e.g. figures for a manuscript, scientific poster). Similar to the editors for raster images, there are plenty of vector graphics programs; representative exam-

ples are Adobe Illustrator (Adobe Inc., San Jose) and Inkscape (The Inkscape team, <http://www.inkscape.org>).

Often, programs support the conversion between raster and vector graphics. These are called *rasterization* and *vectorization*. Usually, vector graphics programs allow you to include raster-graphics without conversion.

Let's say you're done for the day and want to submit another great manuscript to Nature (Nature Publishing Group). Nature has detailed requirements on how your figures should be prepared for print (http://www.nature.com/nature/authors/gta/3c_Final_artwork.pdf, as of 08.10.2014). According to their guide, they prefer vector graphics for text and line-art. We will revisit the Nature figure requirements as we go along discussing properties of digital images!

Can you explain why journals prefer vector graphics for text and line-art?

1.1 Pixels, spatial resolution and sampling frequency

It is important to remember that our microscopes have an optical resolution² that defines the ability to resolve details of the specimen.

Limited by the optical resolution, the number of discrete pixels in a digital image then determines the spatial resolution. The number of pixels is usually called the sampling frequency (or sampling interval). This simply means that the spatial resolution depends on the number of pixels within given physical dimensions and that the maximum spatial resolution is limited by the optical resolution³. It is important to note that each pixel represents the average response of the optical system measured at a point within an area that is specified by the characteristics of our optical system.

In a typical confocal microscope (not camera based), we can choose the number of pixels with which we want to sample the acquired image. A change in the pixel number obviously changes the size of the pixel and therefore also the area of which the average response is obtained. Choosing the wrong sampling frequency is a typical pitfall in microscopy, especially when zoom controls are used. If we sample inadequately, details can be lost. Oversampling is usually not critical, we just increase the amount of data we have to handle and we have to know that we oversample. Fortunately, sampling theory tells us exactly which sampling interval (pixel size, pixel number) is required

²For this course, we assume that you have a basic understanding why microscopes have an optical resolution and on which parameters the optical resolution depends.

³This is only partially true – we can theoretically build a microscope where the detector is the limiting factor. In general, the detector is chosen to match the optical resolution; and all internal microscope components involved in generating the digital image have to be matched as well.

to optimally represent the biological specimen given our optical resolution (the famous *Nyquist theorem* states that we require a sampling interval \geq twice the highest spatial frequency in our specimen and the highest frequencies we can observe are physically limited by our optical resolution).

This raises the question why our microscopes even allow us to change the sampling frequency and not always simply operate at the diffraction-limited configuration. The reason is, that there are many experimental situations where it is desirable to intentionally lower our spatial resolution. Basically, we have to make a trade-off between the required resolution and imaging speed, amount of light collected at each pixel, photo-toxicity or other experimental parameters. The important thing is that we are aware of the fact that we under-sample for whatever reasons we have.

The limited number of discrete pixels in an image acquired with a microscope has serious consequences when you create figures for a manuscript. Publishers often require a certain number of pixels per cm (inch) to create high-quality figures and unfortunately, they leave it up to you to adjust your figures. However, they usually ignore the fact that you only have a limited number of pixels in your original data. One option you have is resampling your original data to change the number of pixels. Another option is to show pixels enlarged – leading to a pixelated view of your data.

1.1.1 Resampling Images

Resampling an image changes the original data; the number of pixels is either increased or decreased (enlarging, shrinking). In comparison, using the zoom tool (icon magnifying glass) just increases or decreases the size of the pixels on your screen without resampling. Similar to other programs, Fiji also allows you to change the size of the canvas (the space on which your image is drawn, see Figure 1.2. If you adjust the canvas size [**Image > Adjust > Canvas Size**], the image is either cut (pixels outside the canvas size are cut off) or a margin is added where added pixels usually have the background color (zero value).

As you saw above, we determine the number of pixels during image acquisition according to our needs and the limits of the microscope. However, when you prepare your data for a presentation, poster or manuscript you will usually have to scale your original data. Before we discuss best practices and look at an example, we will introduce important basic terminology.

Image resolution outside microscopy

In printed images and in document scanners, resolution is often given in *DPI (Dots Per Inch)*, the number of dots within a line that spans 1 inch (2.54 cm). Computer screen, tablet or phone resolution is typically given in *PPI (Pixels Per Inch)*, the number of

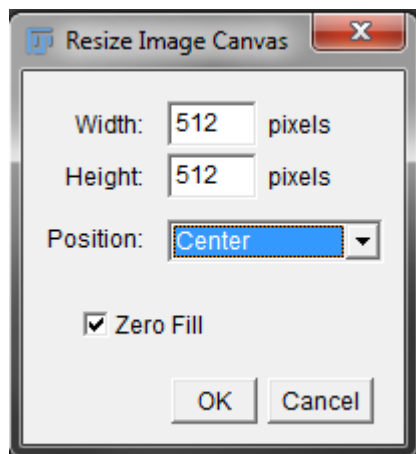


Figure 1.2: Adjust Canvas Size Dialog.

pixels within a line that spans 1 inch. Confusing, DPI is often used when PPI would be better, and even more confusing, one printer dot is usually not equal to one pixel!

As we learned in section 1.1, knowing the number of pixels is not sufficient to define the resolution – pixel size (sampling rate, distance of pixels, ..) is necessary as well. This is also true for consumer products such as digital cameras, smartphones or flatscreen TVs. Knowing that a Full HD TV is capable of displaying 1920 x 1080 pixels presents no information about the size of each pixel. An 18 mega-pixel camera that is able to generate photos with 5184 x 3456 pixels also has no resolution associated. For screens, we can calculate the PPI by dividing the number of pixels by the size of the screen. For example, a 40" Full HD TV would have 55.07 PPI, an iPhone 5 about 326 PPI (4", 1640 x 1136 pixels but two dots per pixel!), an Amazon Kindle Paperwhite 212 PPI (6", 1024 x 758).

High quality photo printing requires about 200-300 PPI. With the number of pixels of our digital camera, we can calculate the maximum size of a high-quality print. For example, an 18 mega-pixel camera print with at least 200 PPI:

18 mega-pixel = 5184 x 3456 pixels, divided by 200 PPI results in a print of about 65 x 43 cm (1 inch = 2.54 cm).

Publishers usually require that you submit your figures in 300-600 DPI (meaning PPI), depending on the content (e.g. see the requirements of Nature⁴ or PNAS⁵). This already determines the maximum size of your (sub)figure given available pixels.

⁴http://www.nature.com/nature/authors/gta/3c_Final_artwork.pdf, as of 08.10.2014

⁵<http://www.pnas.org/site/misc/digitalart.pdf>, as of 04/03/2015

Exercise 1.1: Resampling Example - No Interpolation

1. Open the file `resampling-test.tiff` in `/module1/data`. This is a 20x20 pixel black-and-white (binary) image. Use the magnification tool to zoom to the maximum magnification. You should now see a one pixel wide and a two pixel wide vertical white line and a 1px diagonal line.
2. Before we perform image manipulations, we duplicate the original image for convenience [`Image > Duplicate`].
3. Go to [`Image > Adjust > Size`], you should see a dialog as shown in figure 1.3.

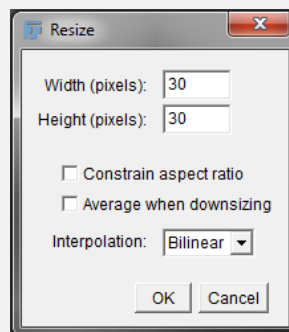


Figure 1.3: Adjust Size Dialog.

4. Perform a resize to 30 x 30 pixels (150% size), with no interpolation, and compare the result with the original figure. Use the **Line-Tool** (see fig. 1.4) to measure the width of both vertical lines. You should observe that one line was not scaled while the other was scaled to 150%.

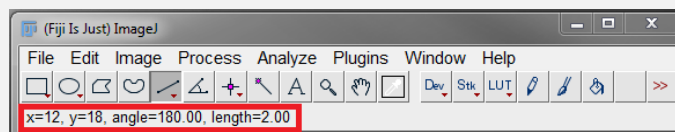


Figure 1.4: Line tool selection and values.

5. Try other values for the resizing and observe the results.

Exercise 1.2: Resampling Example - With Interpolation

1. Again, work on a duplicate of the resampling-test.tif image.
2. Adjust the size to 150% with interpolation set to 'Bilinear'. Use the **Point-Tool** and move the mouse over the image. On the bottom of the Fiji bar, you should see the mouse position in pixels and the value of the current pixel (Figure 1.5).

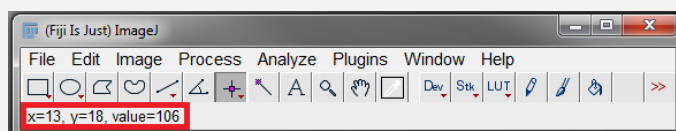


Figure 1.5: Pixel position and value.

3. While the interpolation helps to visually estimate the 150% re-sampling in the vertical and diagonal lines, you can see that the original data has been changed.
4. Try other values for the resizing and observe the results.

As you should have observed, the bilinear interpolation leads to gray values appearing in the previously black-and-white image. Fiji supports the bilinear and the bicubic interpolation. Both interpolation algorithms sample pixel values surrounding each pixel to calculate the pixel value at the given position⁶.

Let us go back to our main question of making figures for a manuscript that we want to publish. PNAS recommends a minimum resolution of 300 DPI for color images and further suggests not to use interpolation when changing the size of the image. What should we do with our microscope images? Following guidelines should help you decide what to do:

- Best practice: *no image resampling ever* (e.g., see recommendations of the Journal of Cell Biology). Usually, you have enough pixels (>1000 x 1000 pixels), allowing you to choose an image size within your figure that is sufficient. For example, a 1000 x 1000 pixel image at 300 DPI results in about 130 pixels/cm – you can choose a single column of 89 mm width (Nature requirements again) to display this image. If you then adjust the size of the image in your graphics software, you will only increase/decrease the size of each pixel (like the zoom tool) without resampling.
- Never resample more than once during image processing.

⁶For more information, Wikipedia has entries for bilinear and bicubic interpolations.

- Do the resampling at the end, after all other image manipulations and quantifications have been done.
- Report any resampling procedures, including the original image size, pixel dimensions and the interpolation method.
- Most journals allow supplementary original data - a good way to present the raw, unfiltered data. Also, there is increasing interest to provide platforms that allow you to publish raw data (e.g. Figshare⁷)
- Most important if you really have to resample: Convince yourself (and your lab-mates) that the original image and the re-sampled image convey the same information.

Fiji also has the option to directly interpolate the image to a given size and DPI (PPI!) using [Image > Adjust > Scale To DPI]. Using this function, you do not have to calculate the values yourself using the [Adjust Size] function (Fig. 1.6).

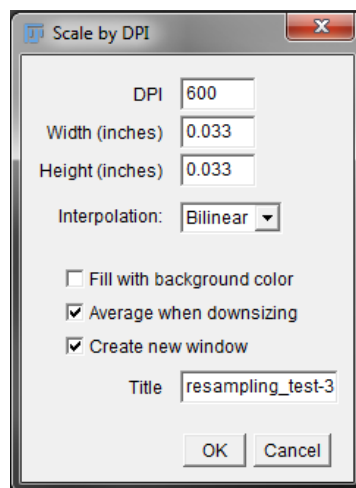


Figure 1.6: Scale to DPI dialog.

1.2 Bit-Depth & Computer Number Representations

We already introduced the term bit-depth that describes the range of values that can be represented at each pixel. When we refer to the possible values of each pixel, we use the terms brightness or intensity in the course context. It is very likely that you

⁷www.figshare.com, an online digital repository - free to upload and free to access; as of 13.10.2014

already encountered the words *bit*, *bits*, (kilo-, mega-, giga-) *bytes* somewhere: 32/64-bit computers, 8/16-bit microscopy images or 50 Mbit/s internet connection speed.

A *bit* is the elementary unit of binary numbers and its possible values are 0 and 1. Electronic devices code and store information with binary numbers because binary states (0 and 1) can easily be implemented in electronics⁸. If you have two bits, you can represent 4 different states (00, 01, 10, 11). In general, if you have n bits, you can represent 2^n different states. Therefore, 8 bits allow you to represent 256 (2^8) different brightness values and 16 bits already 65536 (2^{16}) different values. The higher the bit-depth of our image, the more gray-levels can be represented between black (0) and white (Maximum value). One *byte* consists of 8 bits (historic and convenience reasons). One kilobyte is a multiple of a byte, either 1000 bytes or 1024 bytes; this can be confusing and usually depends on the context. The same is true for further multiples, such as mega-, giga-, tera-, and petabytes.

Similar to choosing an appropriate pixel size (see section 1.1-samplingrate), we also have to make choices regarding the pixel intensity values during acquisition:

- *Bit-depth*: During image acquisition, we round the brightness values found in our samples into a certain number of levels that our chosen bit-depth can represent. For example, if we have 300 distinct brightness levels in our sample and we choose an 8-bit bit-depth, we cannot represent all 300 values (we only have 256 levels!). In this case, the 300 distinct values are rounded to the nearest level that can be represented. Pixels that have different intensities in our sample get rounded to the same values and the differences are lost. In practice, we do not worry about rounding errors during image acquisition – we usually only have to make the choice between a low bit-depth (8-bit) and a higher bit-depth (12/16-bit). This is discussed in section 1.2.1. However, rounding errors can present problems during image processing. This is discussed in later chapters.
- *Data saturation / clipping*: When you take images on a microscope, you usually set the laser output power, detector gain and detector offset for every image. The reason is that whatever bit-depth we chose, we want the brightest pixels just around the maximum value that can be represented by our bit-depth range and the darkest pixels just around zero. If we use the wrong settings and pixels are saturated/clipped, we lose information about the brightest and darkest parts of our sample. This can prevent further analysis of our data! Clipping can also occur during image processing as you will see in a few minutes.

The most common image bit-depths that you can generate with microscopes and that are supported in Fiji are:

⁸While this statement is true, it leaves out any details. However, there are plenty of resources on the web if you are really interested why and how bits are used.

- 8-bit images that can display 256 gray levels with whole numbers (Integers).
- 16-bit images that can display 65536 different gray levels with whole numbers (Integers).

In addition, there is a 32-bit image format that allows you to display 2^{32} gray levels with real numbers. However, this format is a little bit more tricky to use as many functions in Fiji do not handle this bit-depth properly.

When you prepare your figures, bit-depth is usually not a big issue. However, you have to be careful when you export/import images using various file formats – some common file formats only support 8-bit bit-depth and you have to make sure that you convert appropriately.

Exercise 1.3: Bit-Depth Conversion

1. Open the image beads.tif from mod1-publishing/data using [File > Open]. Duplicate the image with [Image > Duplicate...]. Choose the line selection tool and draw a line through one of the bright spheres in the image (Fig. 1.7).

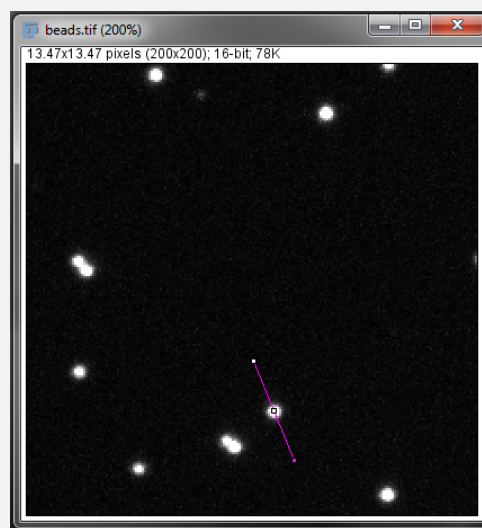


Figure 1.7: A line ROI through one of the beads.

2. In the next step, we will look at the intensity (brightness) distribution along this line. For this, do [Analyze > Plot Profile] (Fig. 1.8). You should observe that the gray value (y-axis) ranges from 0 to 65535 and that the curve looks cut at the upper end - this means that we

have *saturated* pixels, i.e. we cannot resolve any differences between these saturated pixels although the shape of the curve would suggest intensity changes. The [Plot Profile] function allows you to list (show), save and copy the values. If you click on [Live], you can change the line ROI and the plotted profile will update. Try the update by drawing a line somewhere on the background and then again through a bead. Turn the live mode off again by another click on the button.

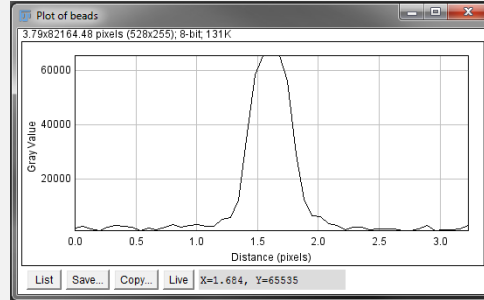


Figure 1.8: Intensity profile of the line.

- Now, we convert the 16bit image to 8bit. First, we make sure that we scale during conversion by [Edit > Option > Conversion]. Then, we use [Image > Type > 8-bit] to convert the image. Make sure that the line ROI is still there and perform the plot profile function again. A second window pops up. Compare the plot profiles of the 8-bit and the 16-bit images. The conversion modified the brightness value (y-value). While the profiles are similar, the scaling is different. The image is scaled according to following equation:

$$I_{8bit} = \frac{I_{16bit}(x, y) - \min(I_{16bit}(x, y))}{\max(I_{16bit}(x, y)) - \min(I_{16bit}(x, y))} * 2^8$$

with:

$I_{16bit}(x, y)$: 16bit image

$I_{8bit}(x, y)$: 8bit image

$\max(I_{16bit}(x, y))$: maximum value of 16bit image

$\min(I_{16bit}(x, y))$: minimum value of 16bit image

- Convert the image back to 16-bit and check the intensity values again. In this case, the intensity values are not increased.
- The conversion actually looks at the data as it is displayed. Adjust the brightness and contrast to an extreme value using [Image > Adjust >

Brightness/Contrast...] (contrast slider to the right edge). Convert the image to 8-bit again. Look at the profile of a bead. You should see that the image only consists of 2 intensities: 0 and 255. As you saw, it is important to reset the brightness and contrast display before converting the image.

1.2.1 Choice of Bitdepth

The choice whether to use a low or high bit-depth depends on the application. Of course you could always use the maximum bit-depth available (to be on the safe side). However, this increases the amount of data you acquire. For example, going from 8-bit to 16-bit doubles the required harddisk space. In addition, image processing might be much slower as well! A typical application where a lower bit-depth is sufficient is where you have a very bright signal (signal-noise ration very good) and you just want to identify cells (vesicle, nuclei, ...) for counting or shape analysis. On the other hand, if you want to analyze bright regions as well as darker regions, or need precise intensity comparisons (e.g. for protein density measurements), a higher bit-depth is recommended.

1.3 Image Dimensions

Up to now, we just looked at 2-dimensional images, where each pixel p can be identified by its spatial coordinates, its position along both axes $p(x,y)$. In microscopy, we often deal with more dimensions, common scenarios are:

- **3D z-stacks:** A single acquisition of a 3D volume. The additional dimension is the depth of the section. Each pixel p can be identified by $p(x,y,z)$.
- **3D time-series:** A sequence of 2D image acquisitions. The additional dimension is the time of the acquisition. Each pixel p is identified by $p(x,y,t)$.
- **3D multi-channel images:** A 2D image is acquired with >1 color channel (multiple fluorophores). The additional dimension is the color. Each pixel is identified by $p(x,y,c)$.
- **4D images:** a combination of 4 already discussed dimensions: z-stacks with multiple color channels, sequence of z-stacks over time or 2D multicolor image over time.
- **5D images:** a combination of all discussed dimensions: sequence of z-stacks with multiple color channels over time. Each pixel p is identified by $p(x,y,c,z,t)$.

Fiji allows you to work with all these image types and we will discuss those methods and possible ways to visualize these data in publications in the following sections.

1.3.1 3D stacks

For historical reasons, Fiji (ImageJ) has two different structures to handle 3D data. At first, ImageJ used *stacks* to support 0-3 dimensions. This was done to z-slices or time points. To be more flexible, *hyperstacks* were introduced to support 0-5 dimensions. For compatibility reasons, both structures co-exist at the moment, but the distinction is disappearing with the hyperstack becoming the standard structure to represent multi-dimensional data. As you will see, hyperstacks are used when importing microscopy file formats. If a pixel p defined by three spatial dimensions $p(x,y,z)$, we call the pixel *voxel* (volume element).