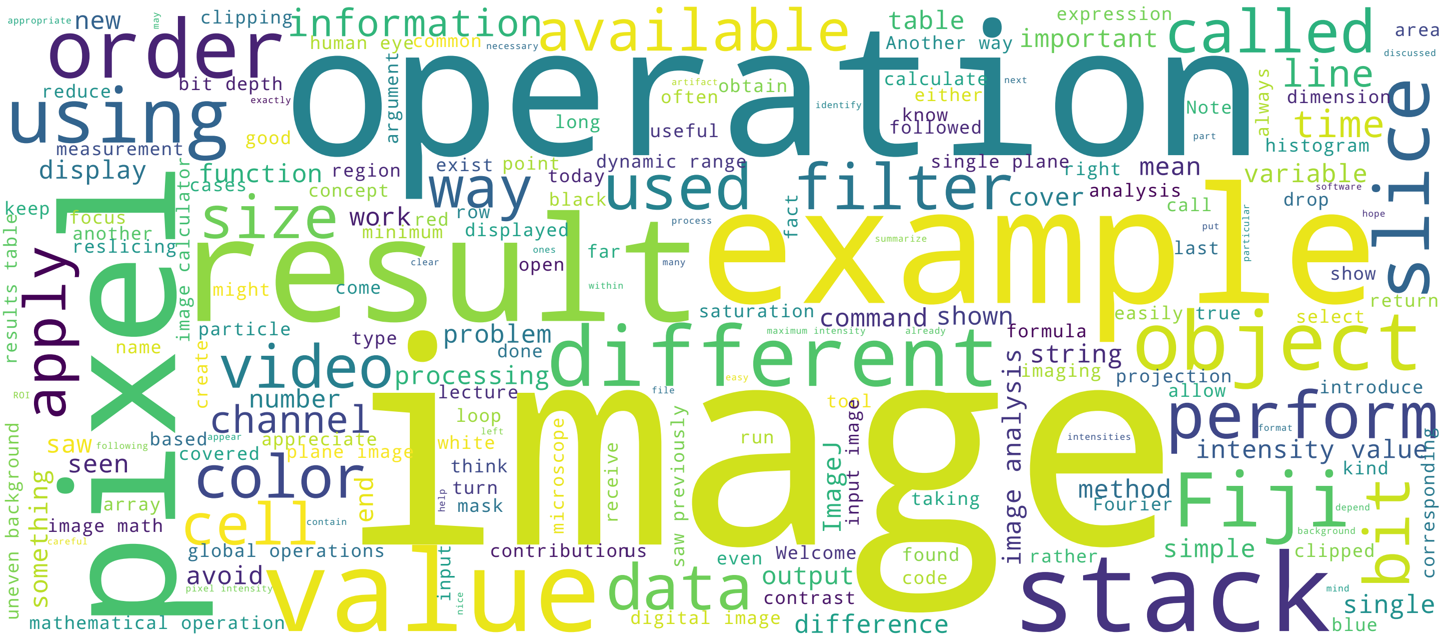
Mathematical Operations On Images

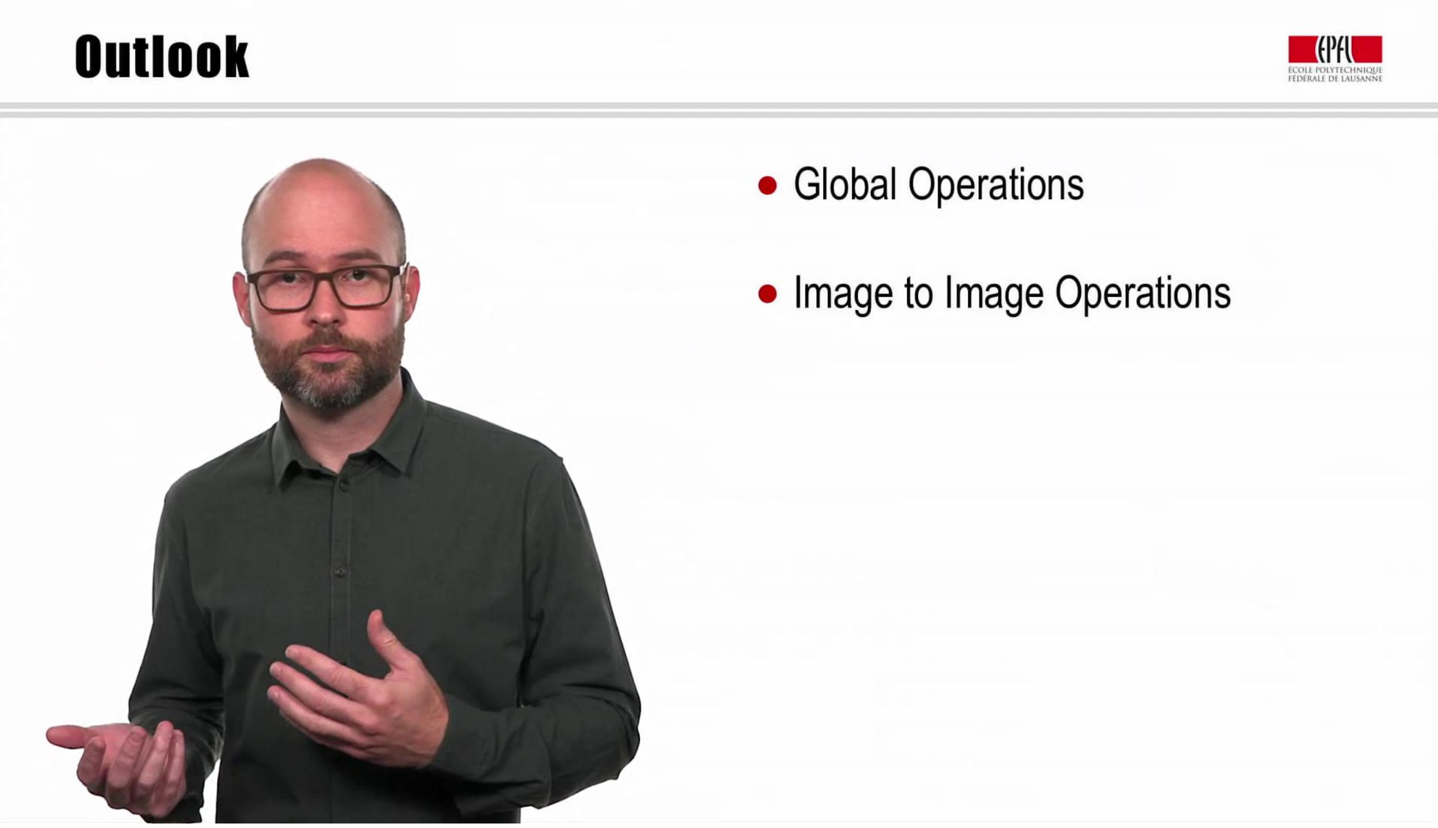
Image Processing & Analysis for Life Scientists

Olivier Burri, Romain Guiet & Arne Seitz



Search MOOC

Video

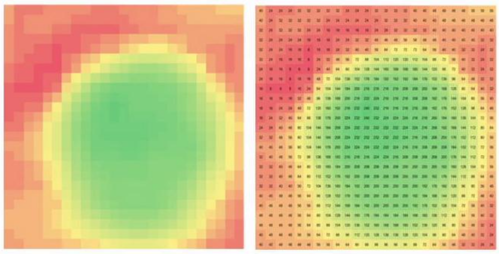


In this video, we will be talking about mathematical operations on images. We will see what kind of operations are available. As well some examples on how they can be directly useful for images processing. There are roughly 2 kinds of image operations we perform. What we will call global operations, where the image is globally modified by a value or a formula. And image to image operations: where each pixel is of one image is affected by the corresponding pixel, of another image.

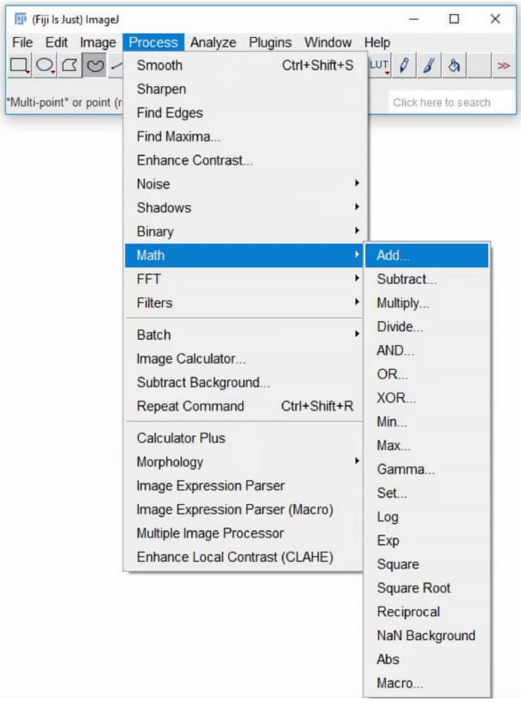
Notes

Summary

0m 05s



+ − ÷ ×

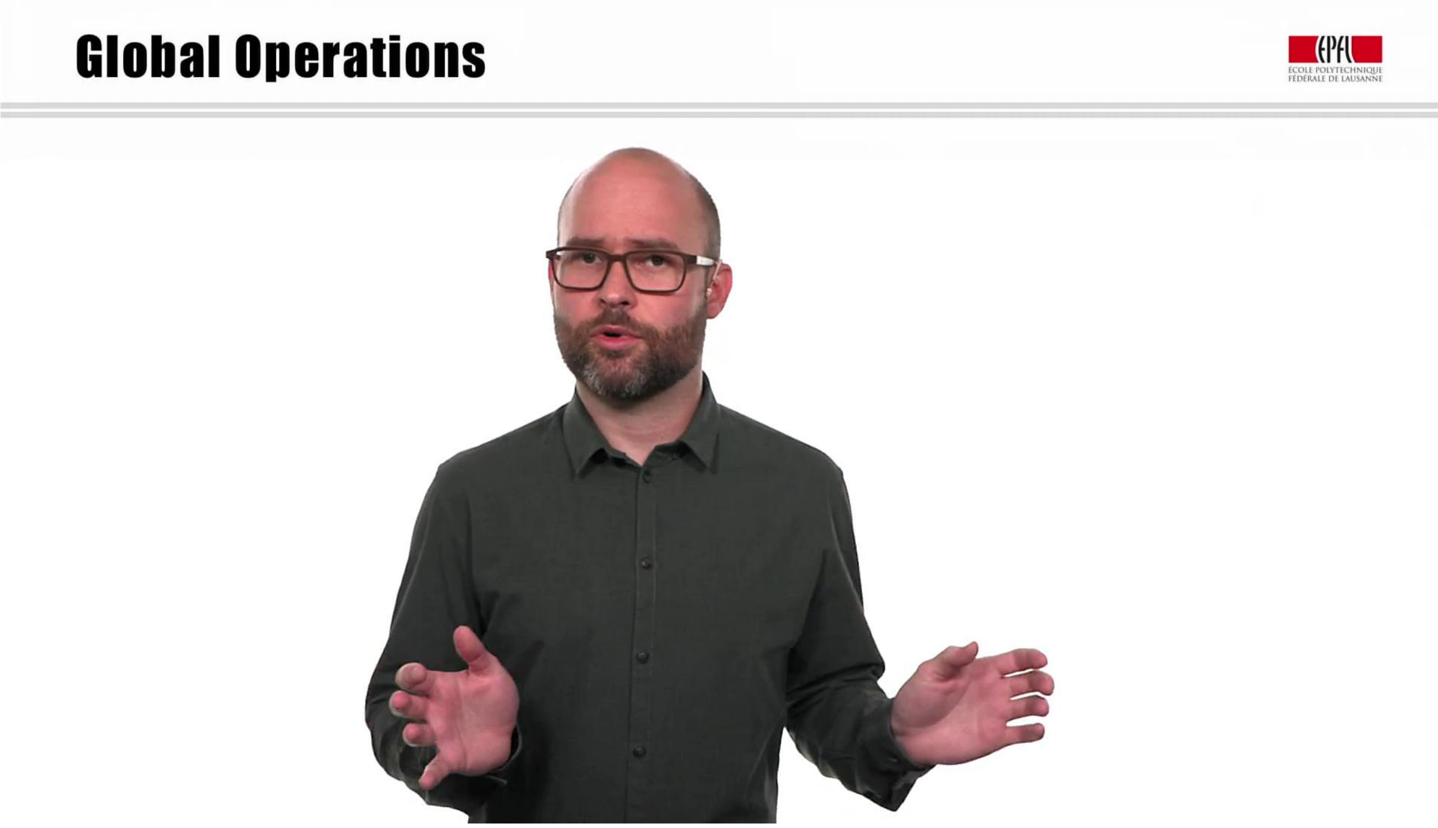


In Fiji, all global operations can be found in the process menu. These operations are available under the math submenu.

Notes

Summary

0m 33s



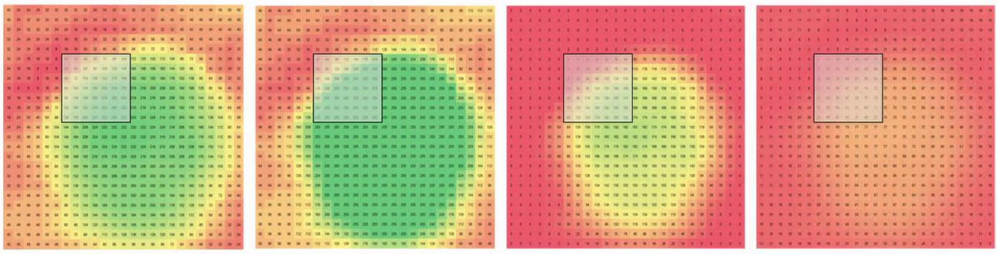
Global operations are simply about taking each pixel an apply a mathematical operation.

Notes

Summary

0m 40s

Global Operations



Original						
24	32	48	56	72	88	104
24	40	64	80	104	128	144
48	72	104	136	152	176	184
64	104	144	184	200	216	216
96	136	168	200	216	232	224
120	160	192	216	232	240	232
136	176	200	224	232	240	232

× 2						
48	64	96	112	144	176	208
48	80	128	160	208	255	255
96	144	208	255	255	255	255
128	208	255	255	255	255	255
192	255	255	255	255	255	255
240	255	255	255	255	255	255
255	255	255	255	255	255	255

- 50						
0	0	0	6	22	38	54
0	0	14	30	54	78	94
0	22	54	86	102	126	134
14	54	94	134	150	166	166
46	86	118	150	166	182	174
70	110	142	166	182	190	182
86	126	150	174	182	190	182

÷ 3						
8	11	16	19	24	29	35
8	13	21	27	35	43	48
16	24	35	45	51	59	61
21	35	48	61	67	72	72
32	45	56	67	72	77	75
40	53	64	72	77	80	77
45	59	67	75	77	80	77

We can multiply the value of each pixel by 2, or subtract 50 or divide it by 3.

Notes

Summary

0m 45s

Data Clipping

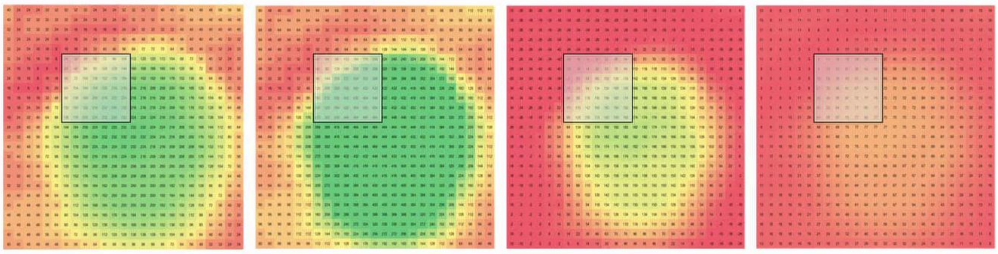
Note however that the results of these operations will depend on the bit-depth of your image. The operations here are done on an inbit image that can restore pixel intensity values from 0 to 255.

Notes

Summary

0m 53s

Data Clipping



Original 32-bit

24	32	48	56	72	88	104
24	40	64	80	104	128	144
48	72	104	136	152	176	184
64	104	144	184	200	216	216
96	136	168	200	216	232	224
120	160	192	216	232	240	232
136	176	200	224	232	240	232

× 2

48	64	96	112	144	176	208
48	80	128	160	208	256	288
96	144	208	272	304	352	368
128	208	288	368	400	432	432
192	272	336	400	432	464	448
240	320	384	432	464	480	464
272	352	400	448	464	480	464

\- 50

-26	-18	-2	6	22	38	54
-26	-10	14	30	54	78	94
-2	22	54	86	102	126	134
14	54	94	134	150	166	166
46	86	118	150	166	182	174
70	110	142	166	182	190	182
86	126	150	174	182	190	182

÷ 3

8	10.7	16	18.7	24	29.3	34.7
8	13.3	21.3	26.7	34.7	42.7	48
16	24	34.7	45.3	50.7	58.7	61.3
21.3	34.7	48	61.3	66.7	72	72
32	45.3	56	66.7	72	77.3	74.7
40	53.3	64	72	77.3	80	77.3
45.3	58.7	66.7	74.7	77.3	80	77.3

If we focus on the red box, we see that numbers going above 255, after an operation, get clipped. Which introduces a saturation. Similarly, values that drop below 0 will result in them being clipped to 0. Finally, as restoring integer data only any operation that can result in decimal values will be rounded to its nearest integer. So, to avoid this, conversion to 30 bits is necessary. And there, the operations become accurate.

Notes

Summary

1m 05s

Global Operations

Theory is nice and well. But what are these operations actually good for? So let's look at a few examples. Each of these examples has an associated image in macrocode available that will showcase how this can be done in Fiji.

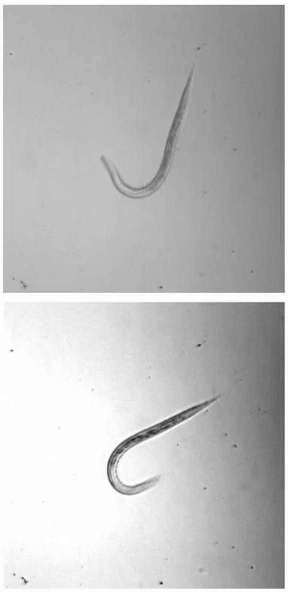
Notes

Summary

1m 33s

Global Operations

- Image Normalization



Simple Normalization

For each slice

1. Subtract min value
2. Divide by (max-min) value
3. Multiply by desired mean

$$I_{new} = \frac{I - I_{min}}{I_{max} - I_{min}} \times \mu_{new}$$

In this example here it would appear as though this movie, over wriggling worm, has suffered from a stuttering light source, unsynchronized camera, or a parasite light. In cases like these, it's always best to locate the problem on the acquisition side. As post acquisition corrections will always make the work a lot harder later on for you and anyone else using this machine. If you see something like this devise a protocol to assess your microscope set-up. And identify the source of the problem. But, luckily, in this case, the only information we need is the shape of the worm; and not some information about the intensity for example. So we could mitigate the problem by performing image normalization. This is a 3 steps process where for each slice we calculate its minimum an maximum intensity values; and then perform the operation you see here. We choose an arbitrary value as the mean of the image. Now after doing this for each slice we get the following result: Note that it's not perfect. This is is due to the fact that the original data already present some saturation. So we can't get a correct estimate of the real max value for each slice for the formula. But you can appreciate how we managed to drop the variations considerably.

Notes

Summary

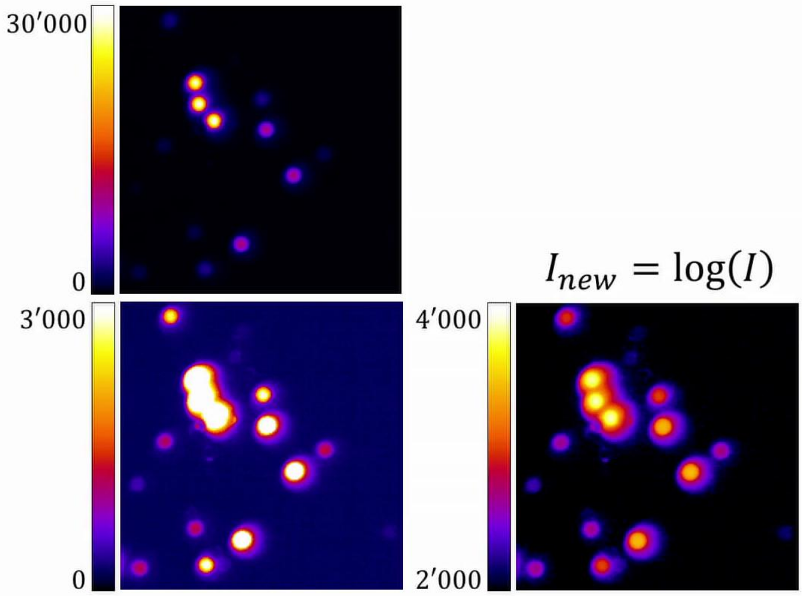
1m 45s

Global Operations

- Image Normalization
- Intensity Flattening

Reduce Large Contributions

$$I_{new} = \log(I)$$



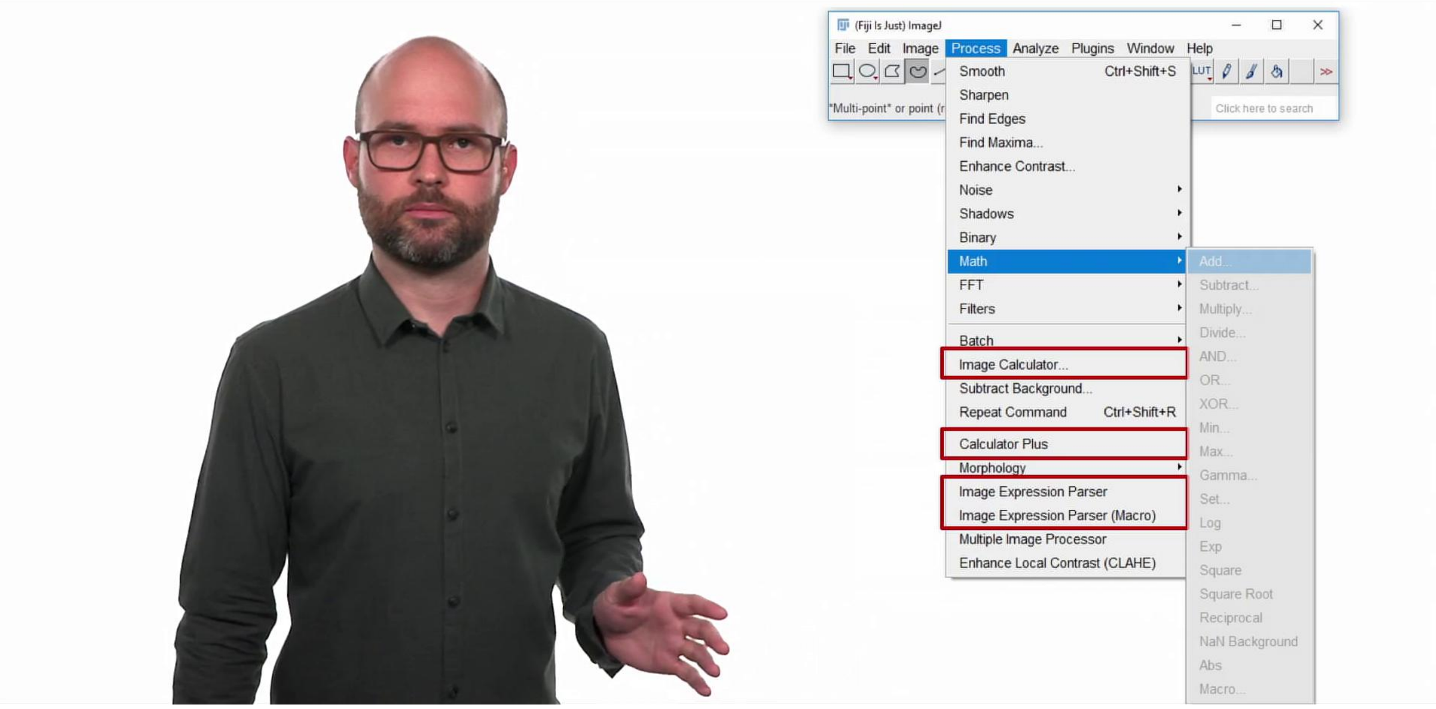
Because cameras have much higher dynamic range than our eyes, it often happens that the data there, but we cannot appreciate it with basic brightness and contrast enhancement like we saw previously. So one way to bring back high dynamic range data into something more palatable for the human eye is to reduce the contribution of very strong pixels, while keeping the contribution of weak pixels relatively small. For example, by taking the square root or log of the image. This results in an image that is simpler to visually inspect, and is a perfectly acceptable pratice. As long as it is properly reported.

Notes

Summary

2m 58s

Image to Image Operations



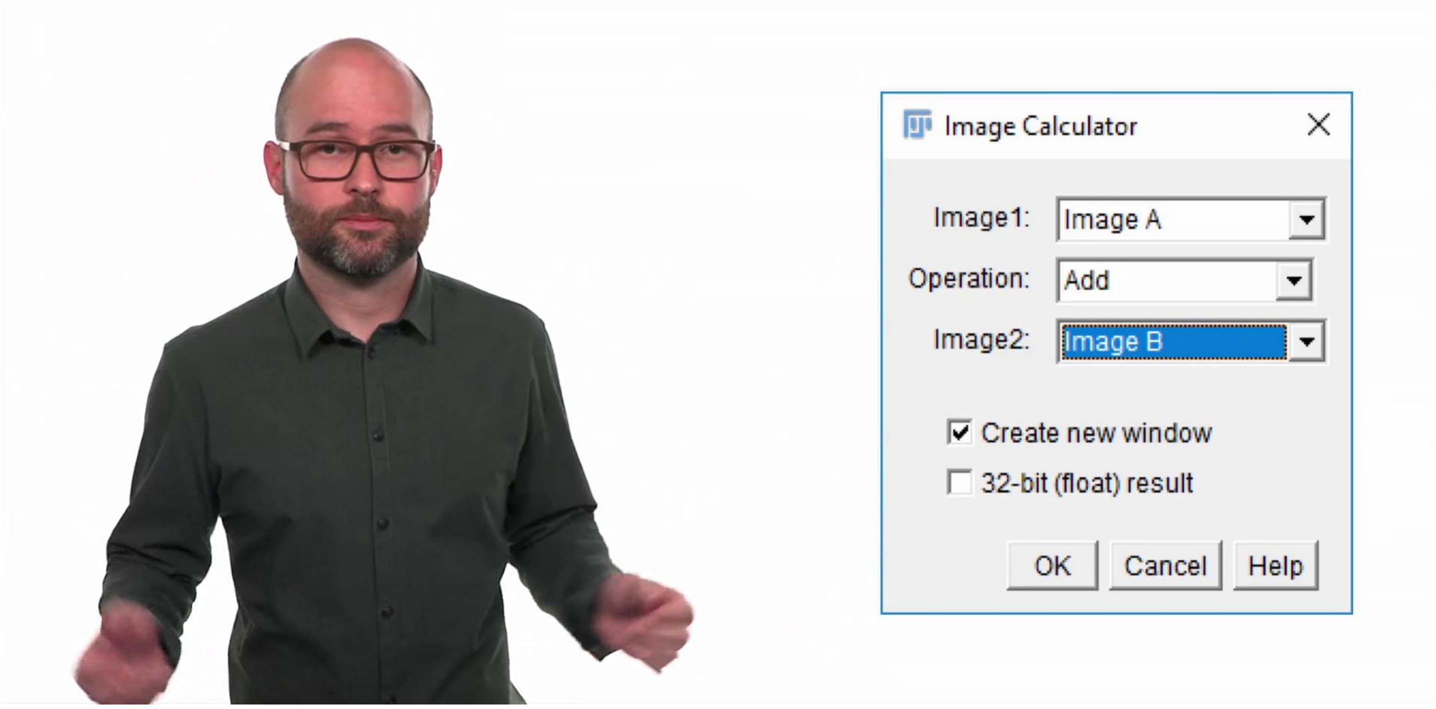
But we can do so much more with images. We can apply the pixel of one image to the pixel of another. Which is what we call image to image operations. There are multiple Tools to perform this in Fiji such as: the image expression parser, the macro version of the parser. But the most common and easy to use in macros is the image calculator.

Notes

Summary

3m 32s

Image to Image Operations

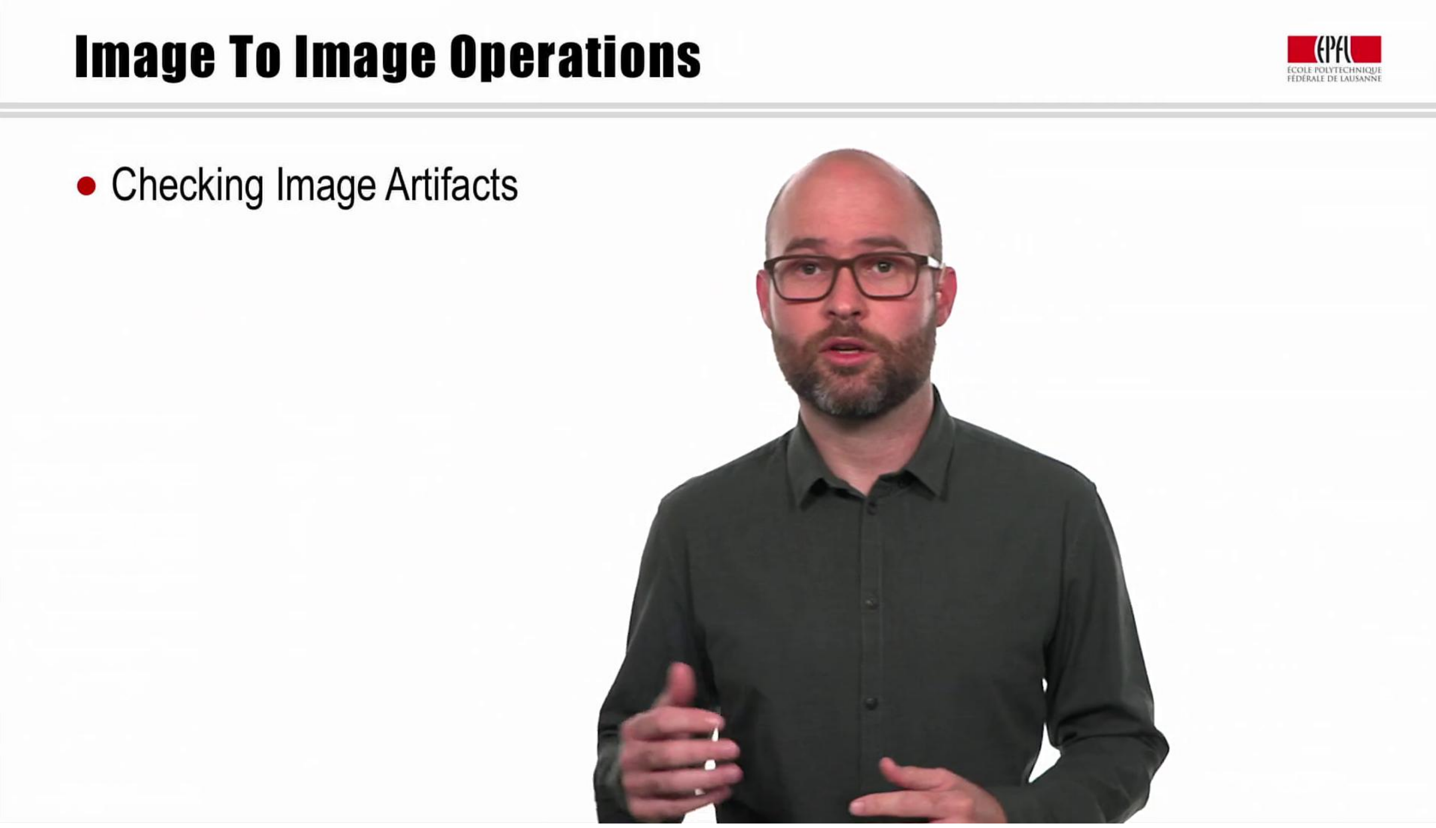


The interface is rather straightforward. You can select your 2 input images and the operation you wish to perform. The output can either be put in Image A, or into a new window. Similarily, you can force the result to be 32 bit, in order to avoid the clipping and rounding problems we saw previously. So let's look at some examples of what we can do with these.

Notes

Summary

3m 52s



We've already covered problem with lossy compression. Another way we can use to see the difference between these 2 images involves simply taking the original image and subtracting the compressed image.

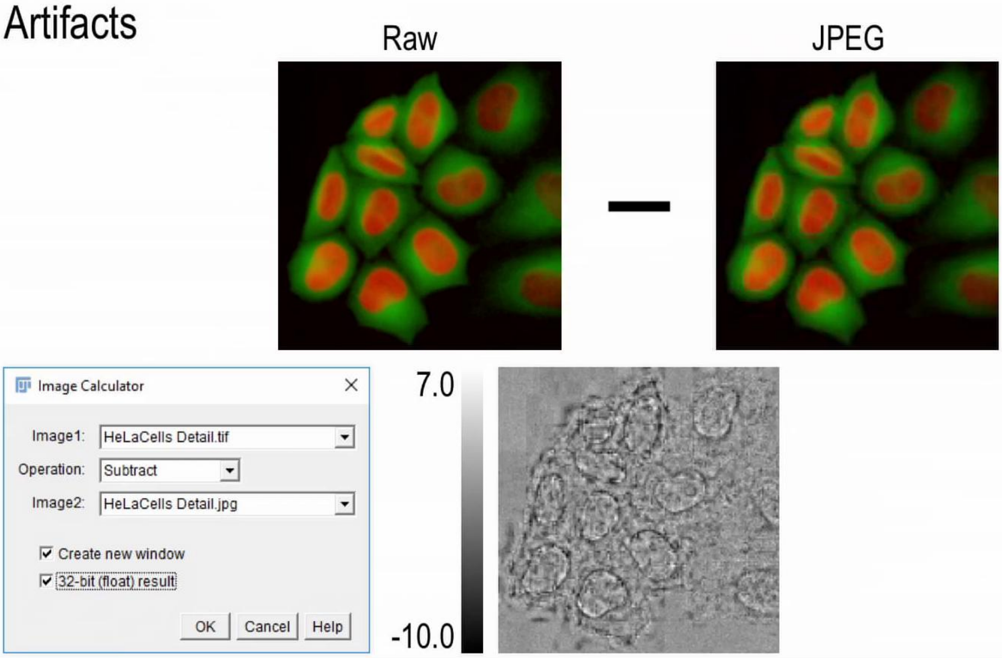
Notes

Summary

4m 14s

Image To Image Operations

- Checking Image Artifacts



The result here is 32 bits image, showing the difference in average intensity.

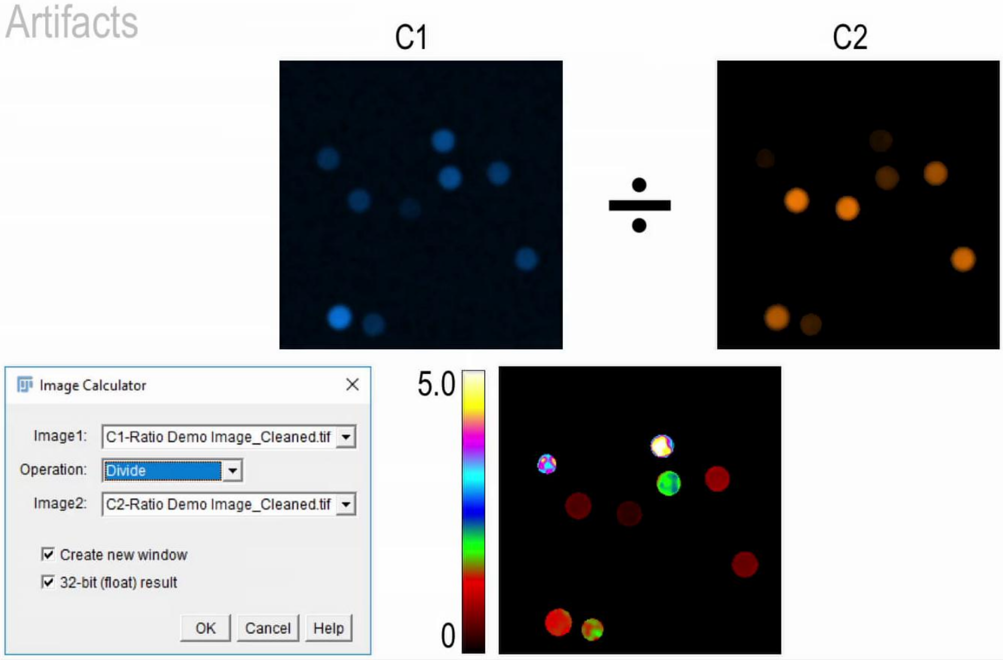
Notes

Summary

4m 26s

Image To Image Operations

- Checking Image Artifacts
- Ratio Imaging



Another way to operate on images is in the case you need to establish some sort of relationship between two channels. While there are several preprocessing steps we will not cover in this video, we can easily obtain a ratio of these 2 images and visualize them with an appropriate look up table.

Notes

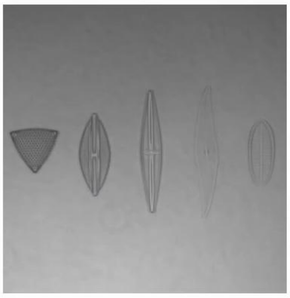
Summary

4m 32s

Image To Image Operations

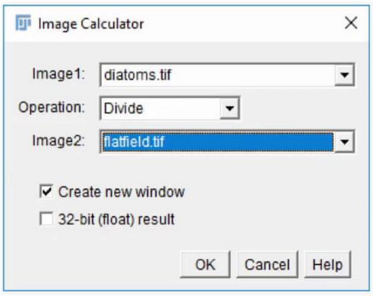
- Checking Image Artifacts
- Ratio Imaging
- Flatfield Correction

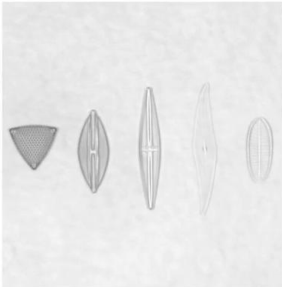
Sample



÷

Empty



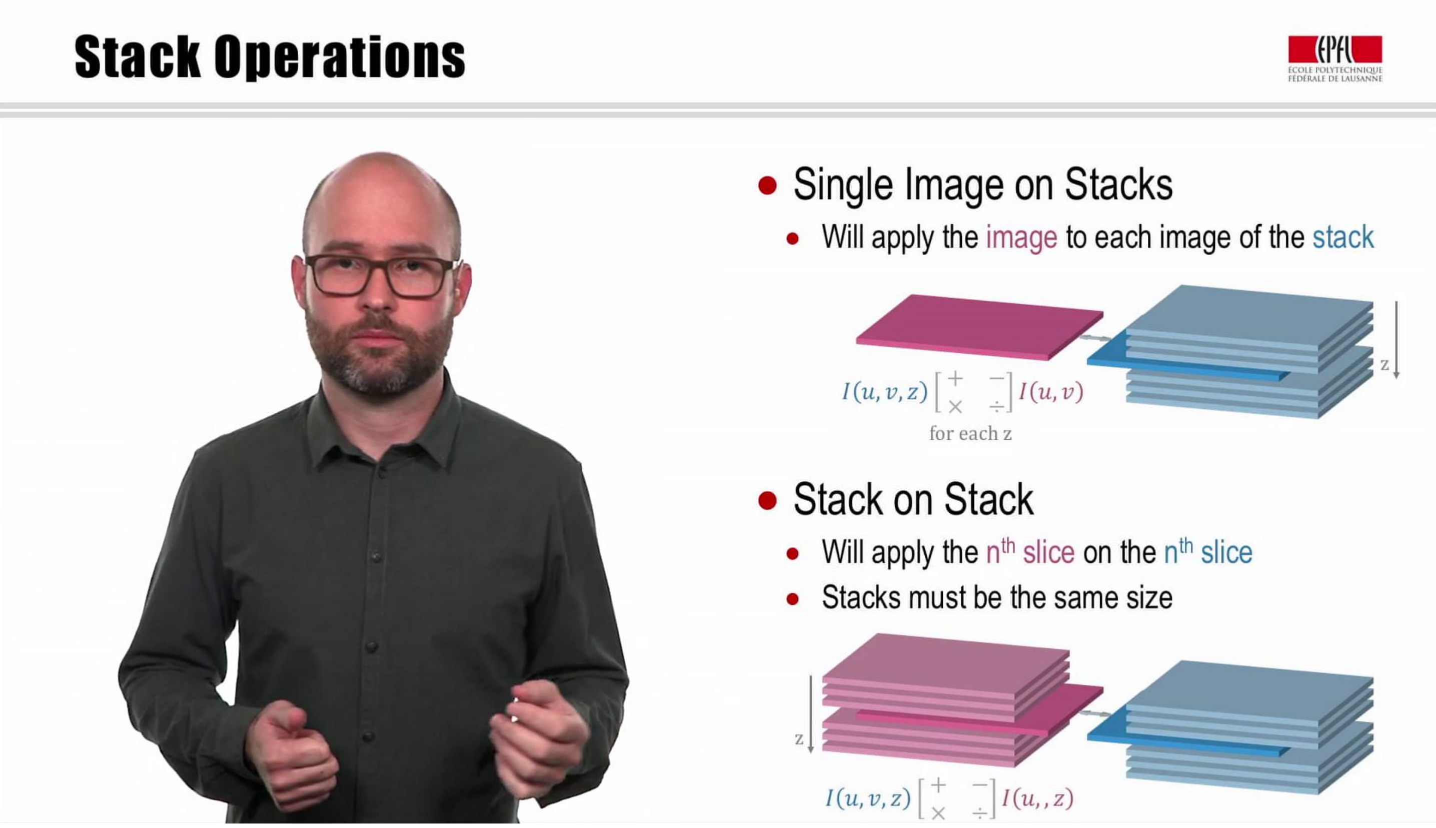


Perhaps some more concise example is in the case of Brighfiled microscopy. Flatfield correction is ofetn used to compensate for uneven background; or even small optical problems in a microscope [inaudible] path. They are independant of the samples. This requires 2 images. The sample followed by an image, where the sample was removed. By deviding the 2 images we are able to clear the uneven background and artifacts, and enhance the contrast of our image.

Notes

Summary

4m 48s

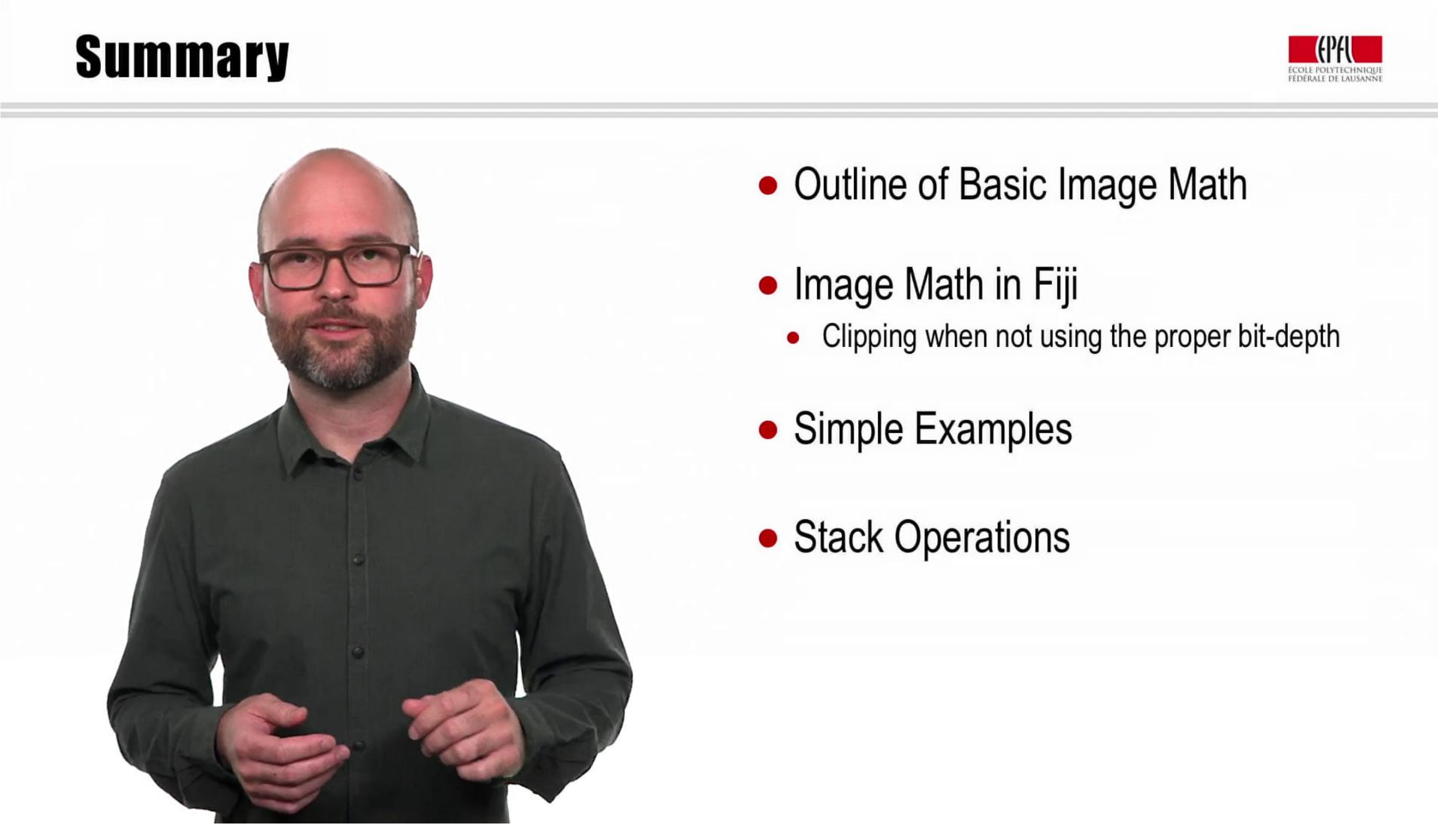


So far, we've only seen operations from single plane image to single plane image. But, what about stack operations? It turns out they are rather intuitive. And you have 2 ways of using them. The first one is that you can apply a single image into a stack. And the result will be that each slice of the stack will receive the same treatment. You can also work with a stack on a stack: where each slice of the image 1 will be operated by each slice of image 2. Now these operation will only work if the stack you're using are the same size. Otherwise you will receive an error message.

Notes

Summary

5m 15s



So that covers the topic of image math. We've covered the basic operations we can perform and seen where in Fiji the're located. And how to use them. We saw how we need to be careful about clipping if we use incorrect bit depths. We also saw a few useful examples where you can see how simple image math can help us enhance our images. And finally, we touched on how stacks are treated by the image calculator in Fiji. That's it for today. Thank you!

Notes

Summary

5m 49s