



# GPU-accelerated real-time image analysis: key to smart microscopy

Robert Haase, Daniela Vorkel, Akanksha Jain, Nicola Maghelli, Pavel Tomancak, Eugene W. Myers

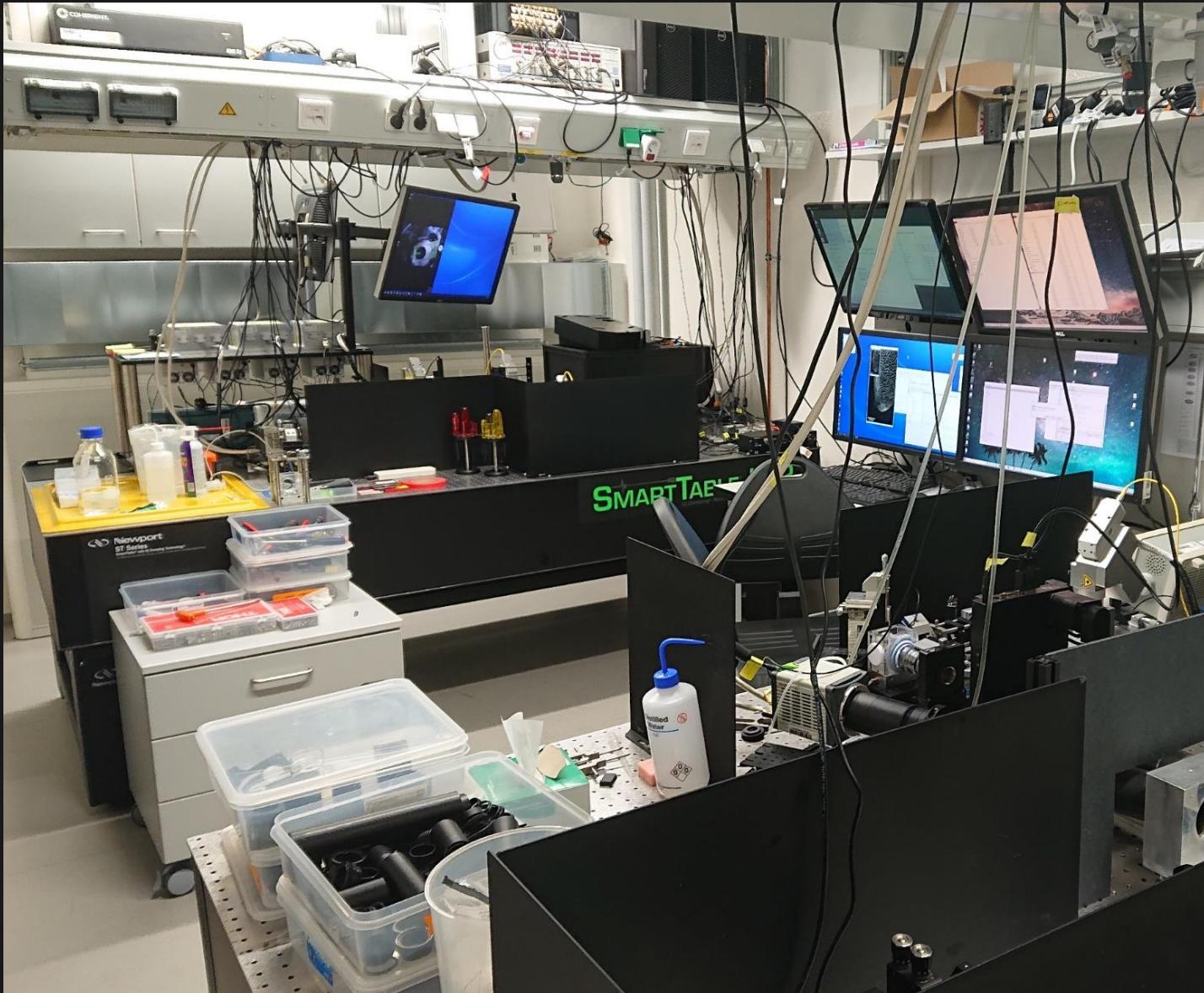
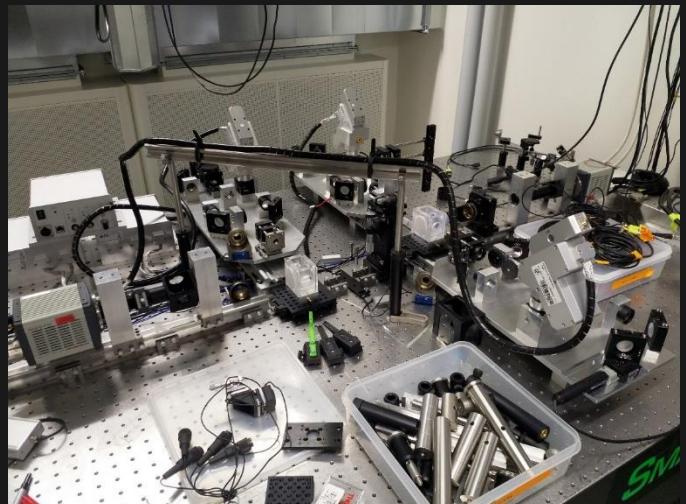
Myers lab, MPI CBG / CSBD Dresden

#QBI2020

# Introduction: Gene Myers lab – Smart Microscopy



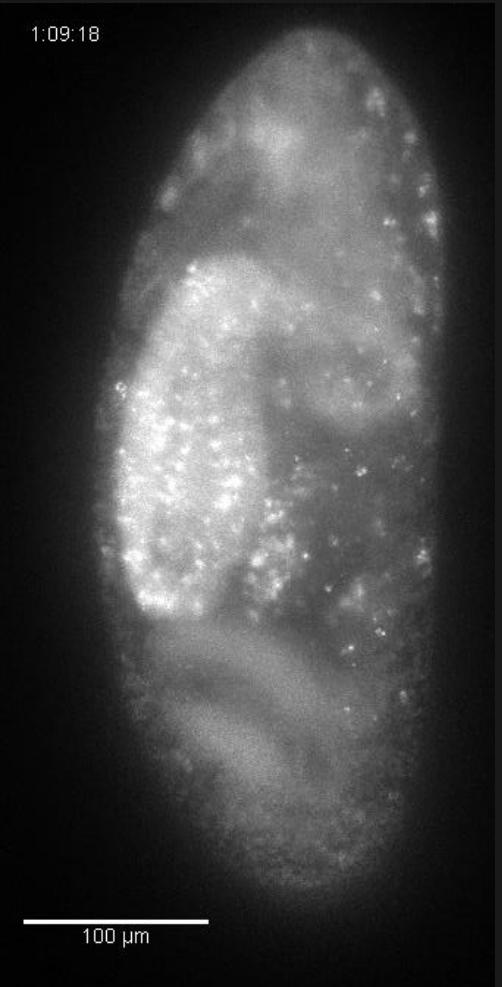
- 5 Microscopes
  - Spinning disc confocal
  - Meso-scope
  - 3 light sheet microscopes
- Closest collaborators
  - Advanced Imaging Facility @ MPI CBG
  - Tomancak lab @ MPI CBG
  - Jug lab @ CSBD / MPI CBG
  - Royer lab @ CZ Biohub



# Fast long-term live imaging



- Imaging fast

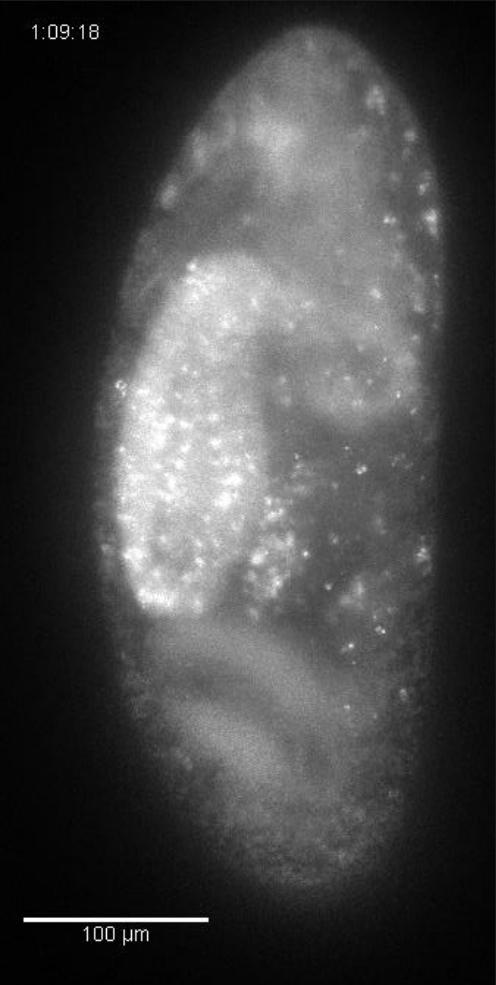


Hatching Drosophila larva @ 20 fpm

# Fast long-term live imaging



- Imaging fast



Hatching Drosophila larva @ 20 fpm

and

long-term



Imaging 1 week  
with 20 fpm  
200 MB each  
=====  
200000 frames = 40 TB



Tribolium embryo development:  
1 week, 3506 frames

# Smart Microscopy

---



Dear microscope, we just put a *Tribolium castaneum* embryo in your chamber. Could you please

- image ventral furrow formation at increased frame rate?

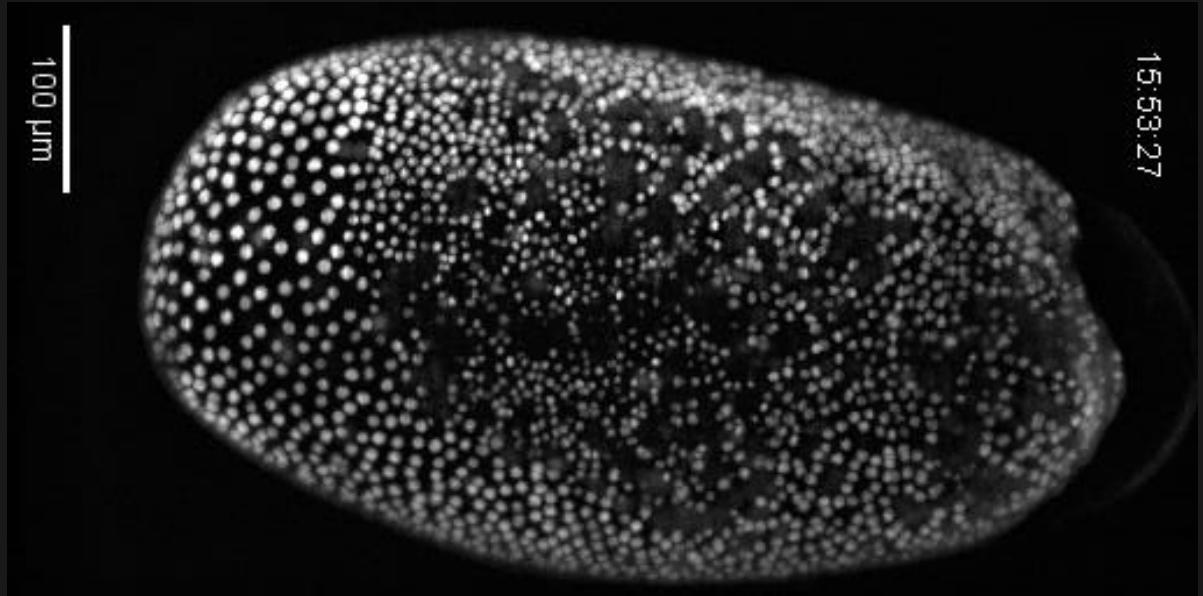
# Smart Microscopy



Dear microscope, we just put a *Tribolium castaneum* embryo in your chamber. Could you please

- image ventral furrow formation at increased frame rate?

Sure! I increased frame rate after 17 h 50 min.



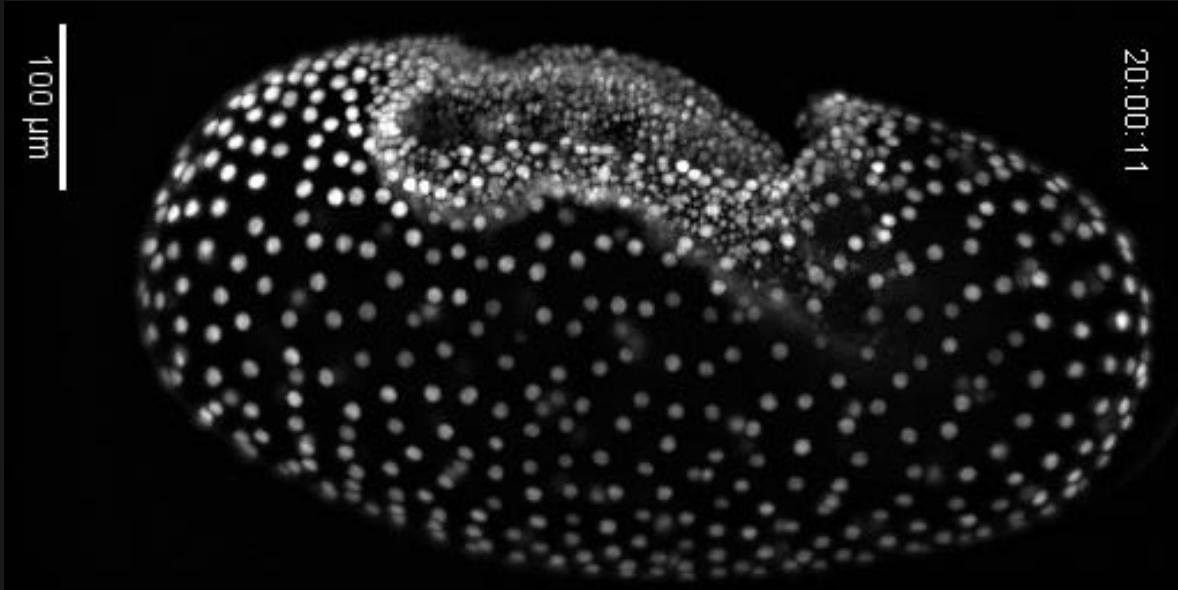
# Smart Microscopy



Dear **microscope**, we just put a *Tribolium castaneum* embryo in your chamber. Could you please

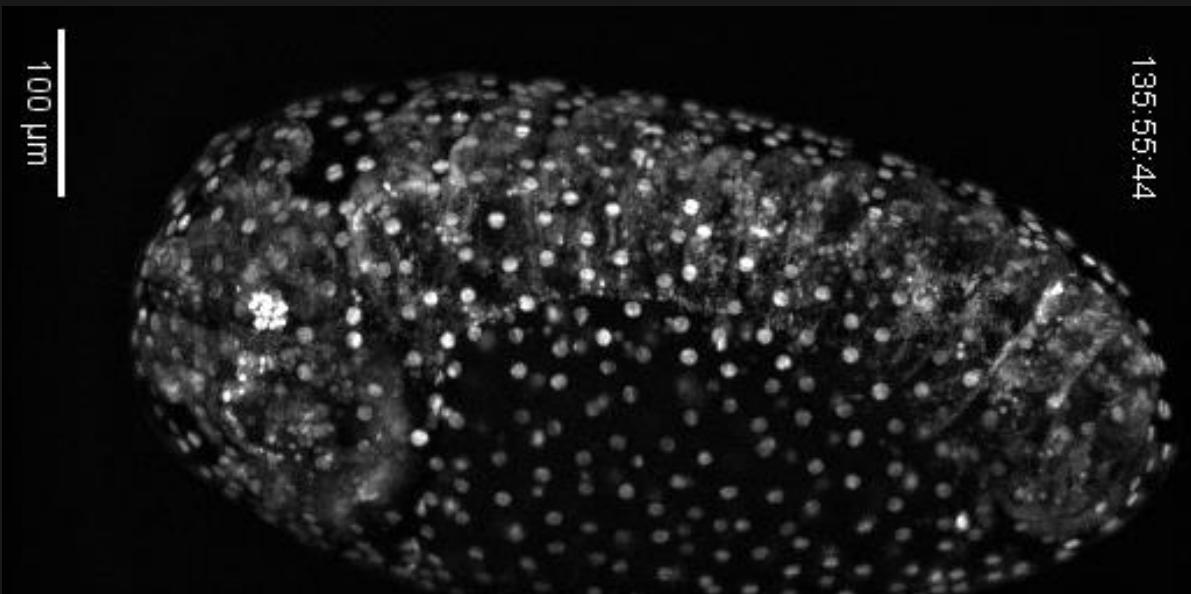
- image ventral furrow formation at increased frame rate?

Sure! I increased frame rate at 17:50.



- take a time lapse of serosa rupture?

Sure! Serosa rupture happened after 139 h 35 min

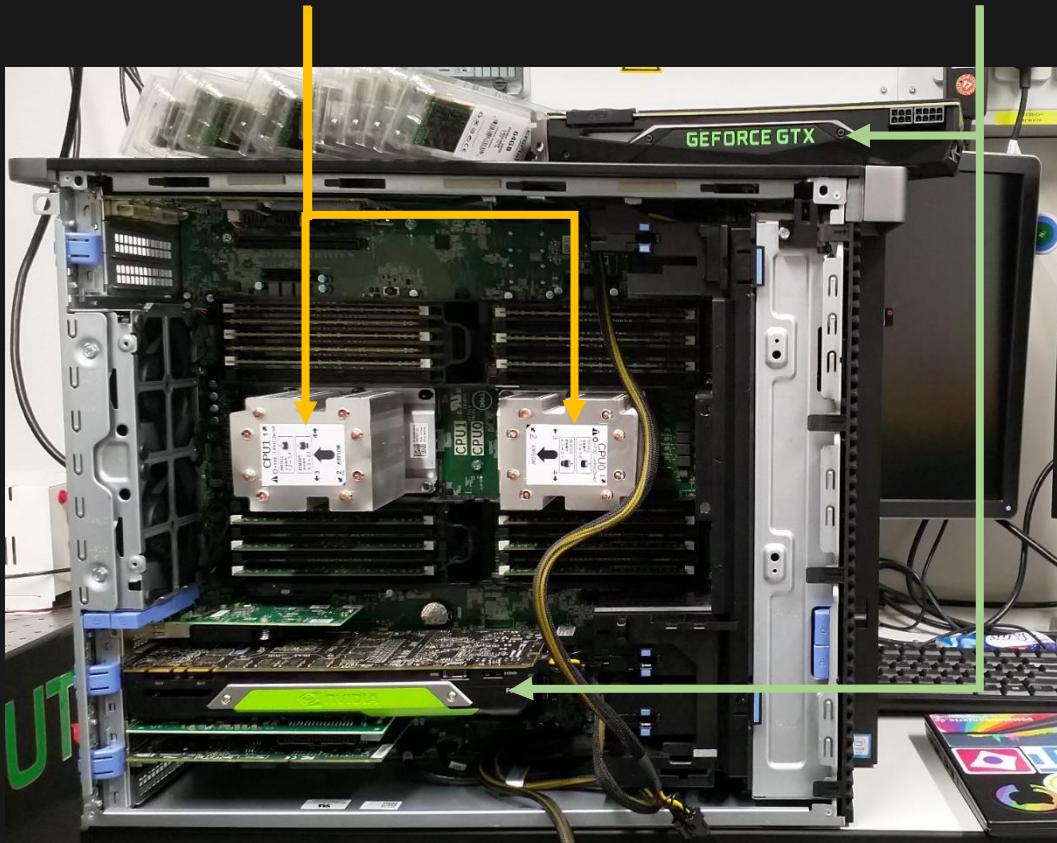


# GPU-accelerated image processing

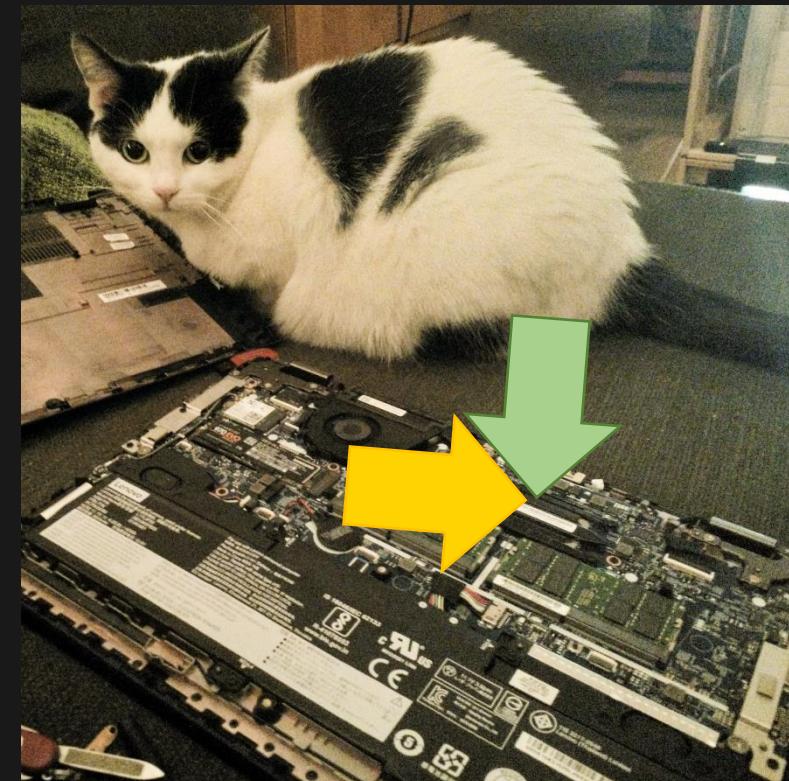


- Typical computers contain Graphics Processing Units

Central Processing Unit (CPU)



Graphics Processing Unit (GPU)



Most laptops contain *integrated GPUs*

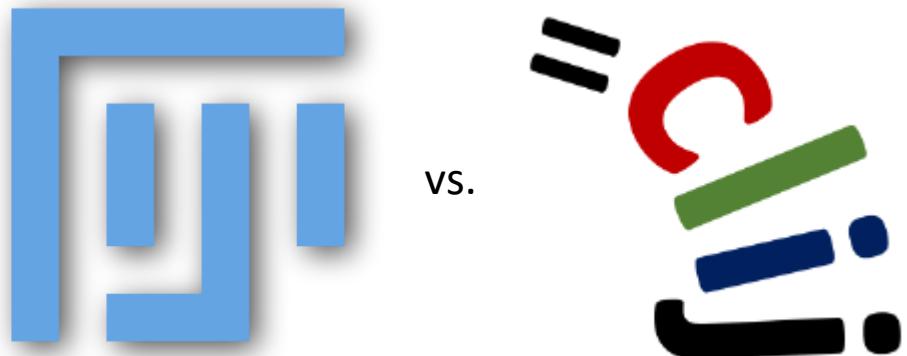


Alternative: *external GPUs*

# GPU-accelerated image processing



- ... depends on operation, image size, parameters, hardware, ....



Workstation CPU

2x Intel Xeon Silver 4110

Workstation GPU

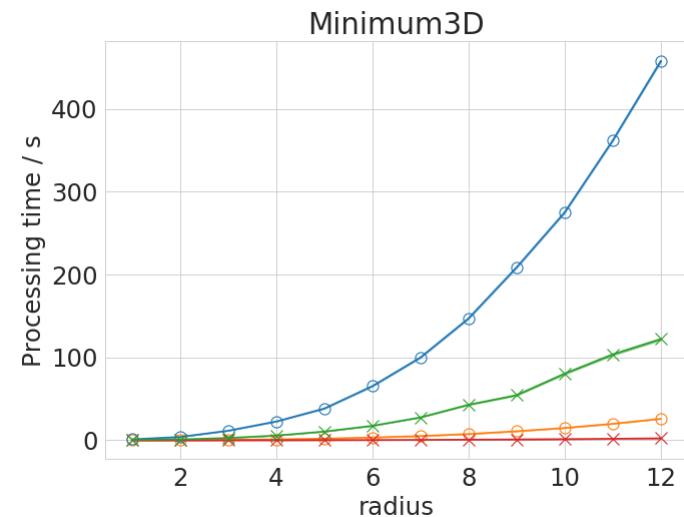
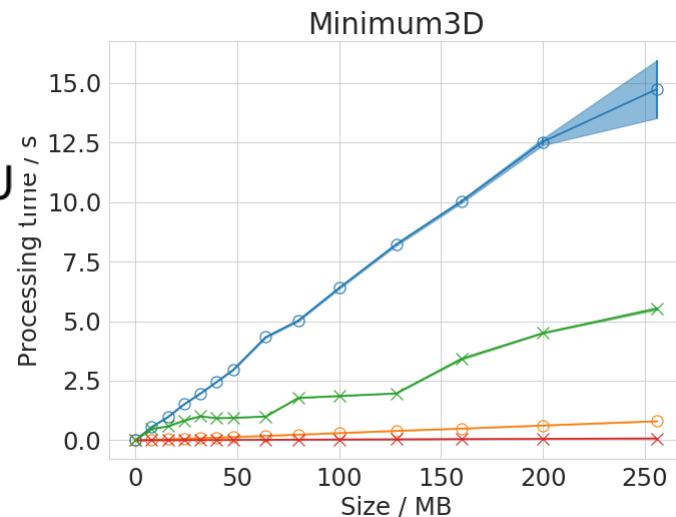
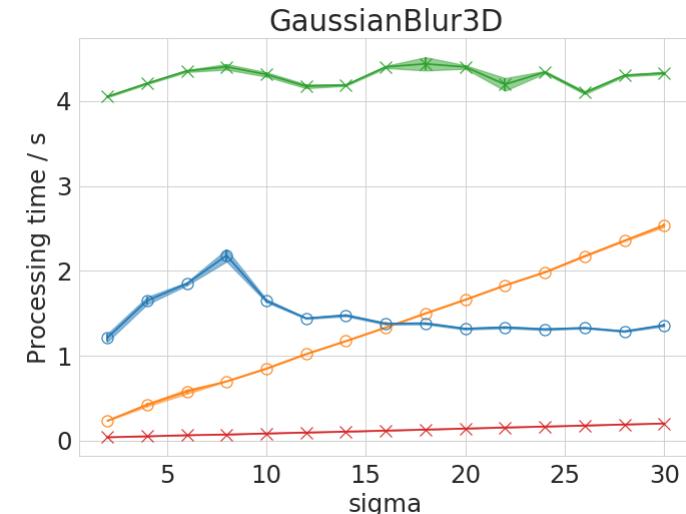
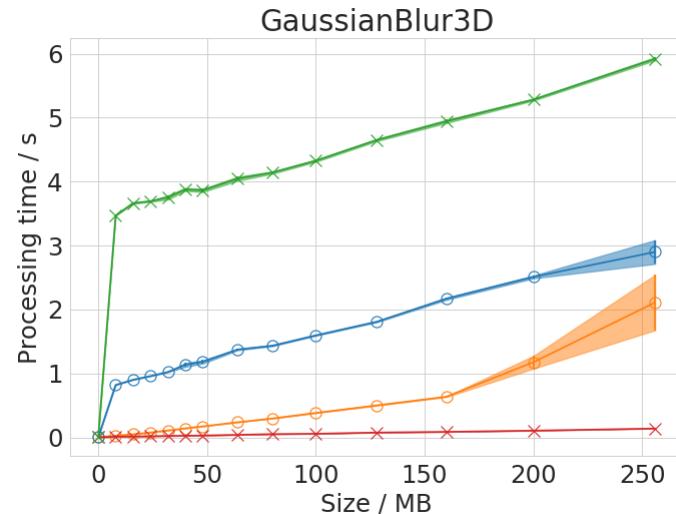
Nvidia Quadro P6000

Laptop CPU

Intel Core i7-8650U

Laptop GPU

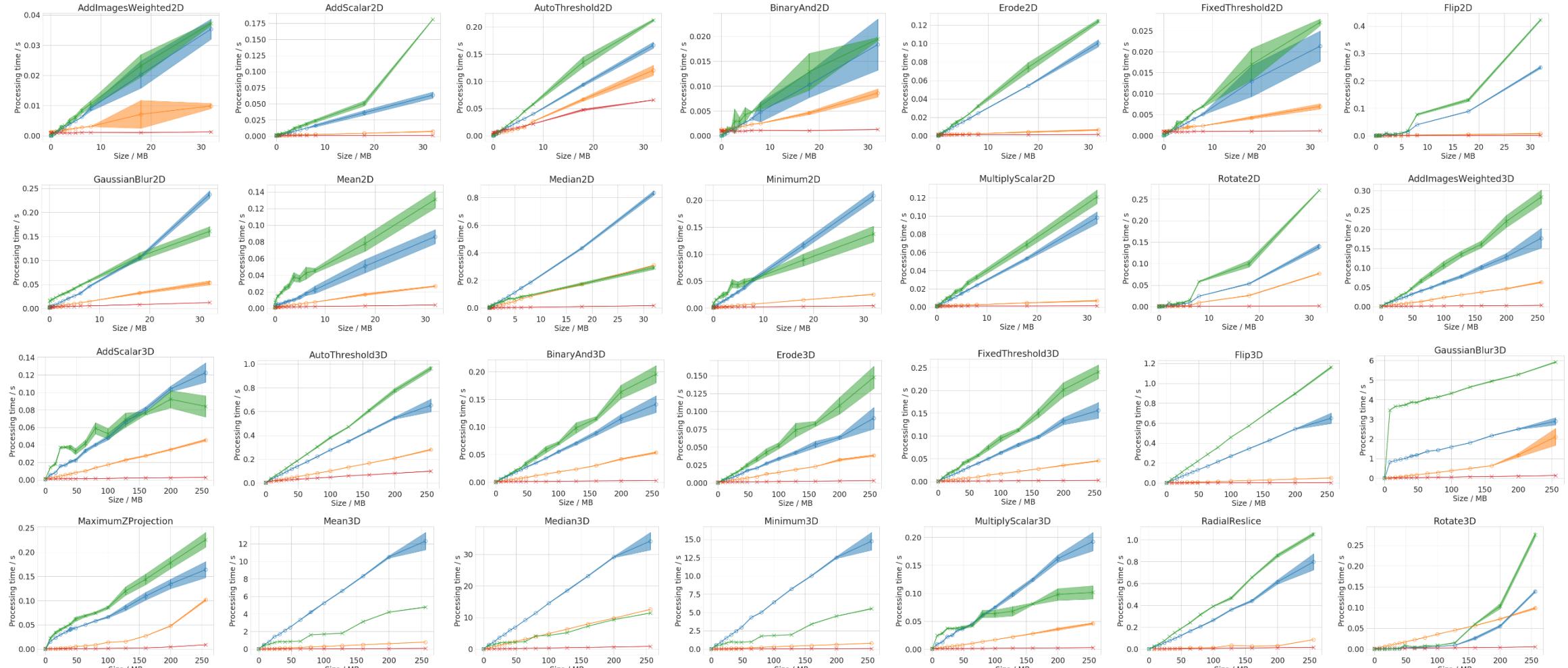
Intel UHD 620 GPU



# GPU-accelerated image processing



- ... depends on operation, image size, parameters, hardware, ....



laptop CPU    laptop GPU    workstation CPU    workstation GPU

# GPU-accelerated image processing



- 8 MB (2D)
- 64 MB (3D)



Speedup compared to Laptop CPU

	Laptop	Workstation
	GPU	GPU
AddImagesWeighted2D	3	8
AddScalar2D	7	14
AutoThreshold2D	2	2
BinaryAnd2D	2	4
Erode2D	11	20
FixedThreshold2D	2	5
Flip2D	16	37
GaussianBlur2D	3	9
Mean2D	3	10
Median2D	2	35
Minimum2D	7	22
MultiplyScalar2D	10	21
Rotate2D	3	22
AddImagesWeighted3D	3	26
AddScalar3D	3	23
AutoThreshold3D	3	5
BinaryAnd3D	3	24
Erode3D	2	13
FixedThreshold3D	4	30
Flip3D	15	119
GaussianBlur3D	6	35
MaximumZProjection	7	46
Mean3D	18	150
Median3D	3	43
Minimum3D	23	188
MultiplyScalar3D	4	28
RadialReslice	14	42
Rotate3D	0.1	2

# Event driven smart microscopy



- Spot detection for developmental stage estimation

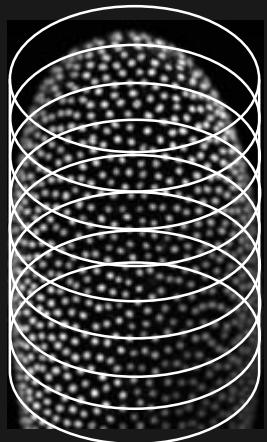
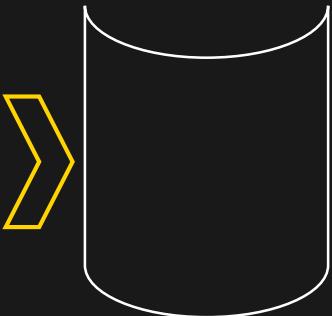
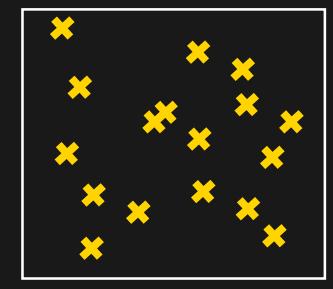
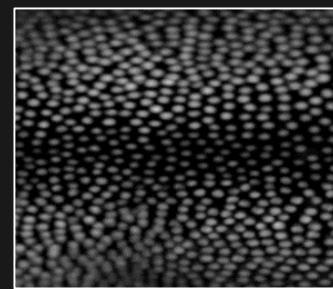


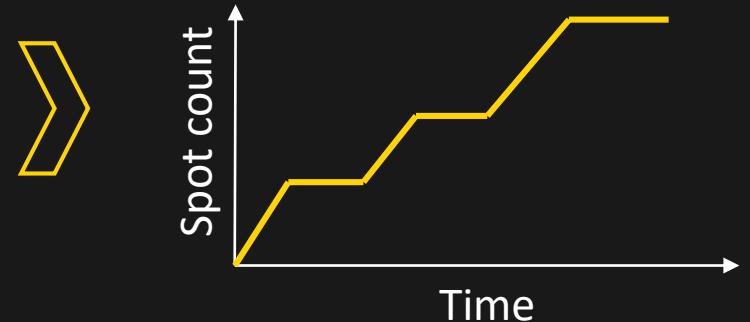
Image stack



Cylinder maximum projection



Spot detection



Spot count over time

# Event driven smart microscopy



- Spot detection for developmental stage estimation

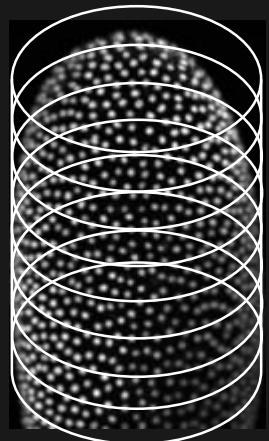
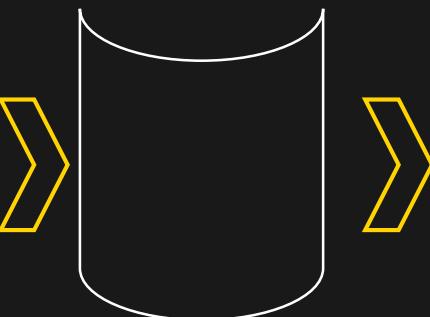
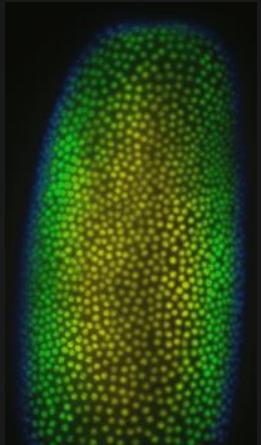
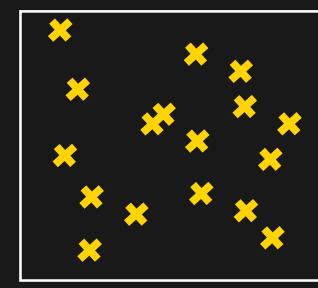
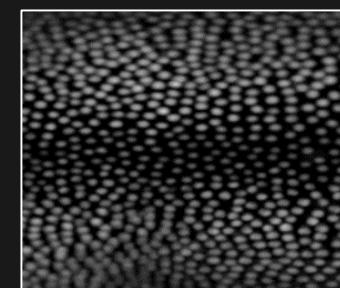


Image stack



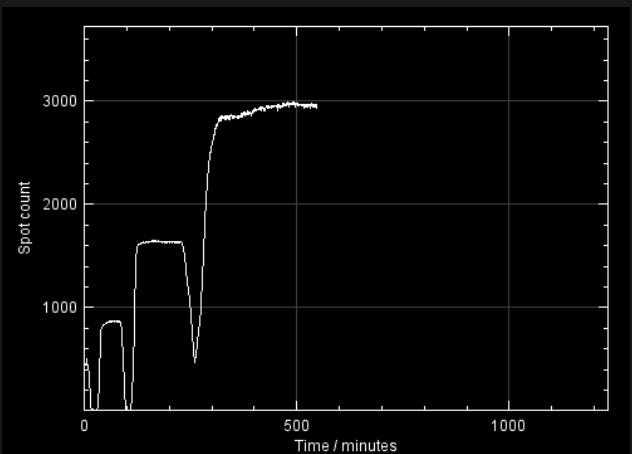
Cylinder maximum projection



Spot detection



Spot count over time

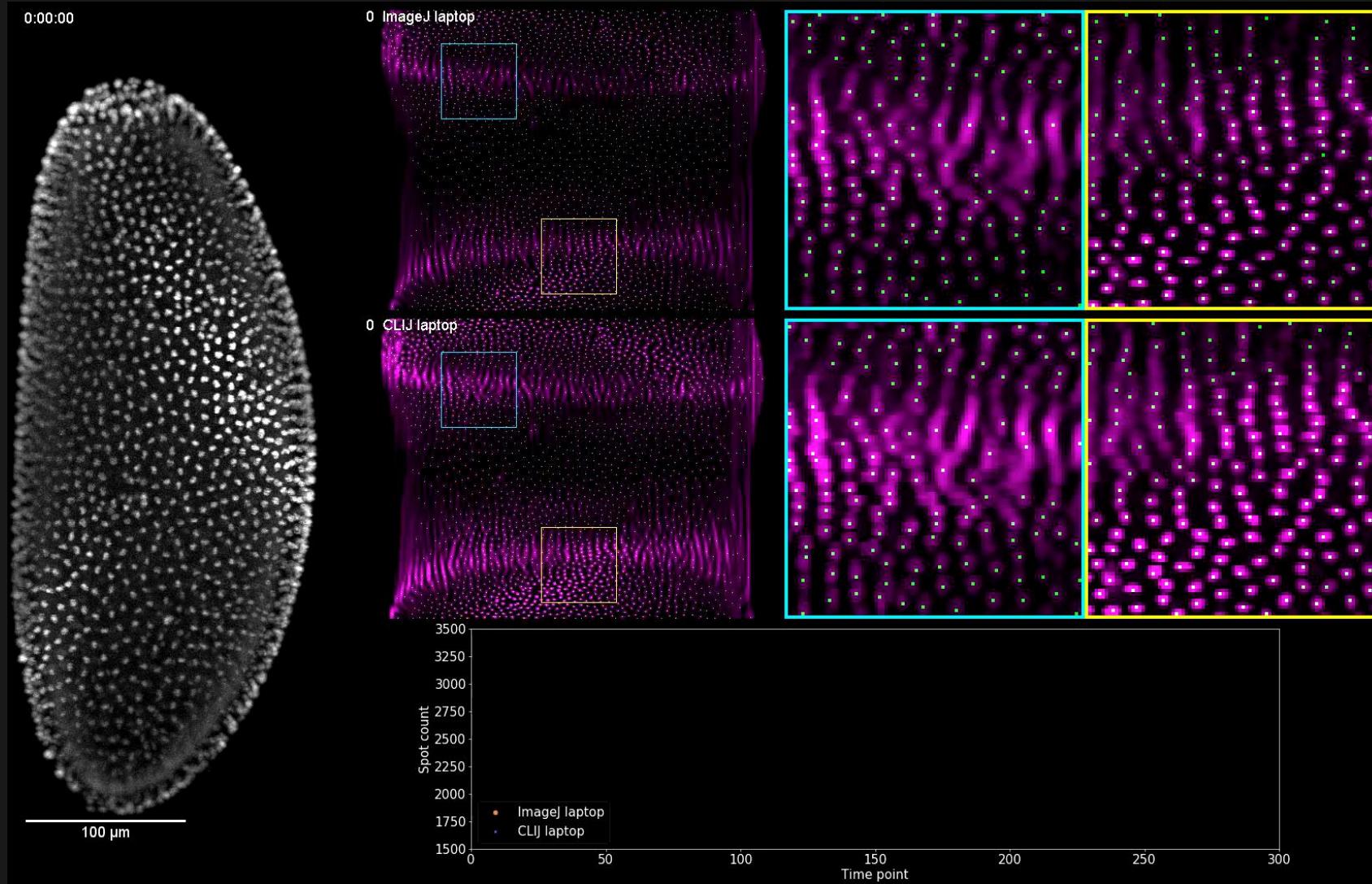


# Real-time image processing



- Counting spots in 300 frames of light sheet data (including I/O)

Drosophila melanogaster, histone-RFP



ImageJ on CPU (laptop)  
33 seconds per frame  
2:44 h (timelapse)

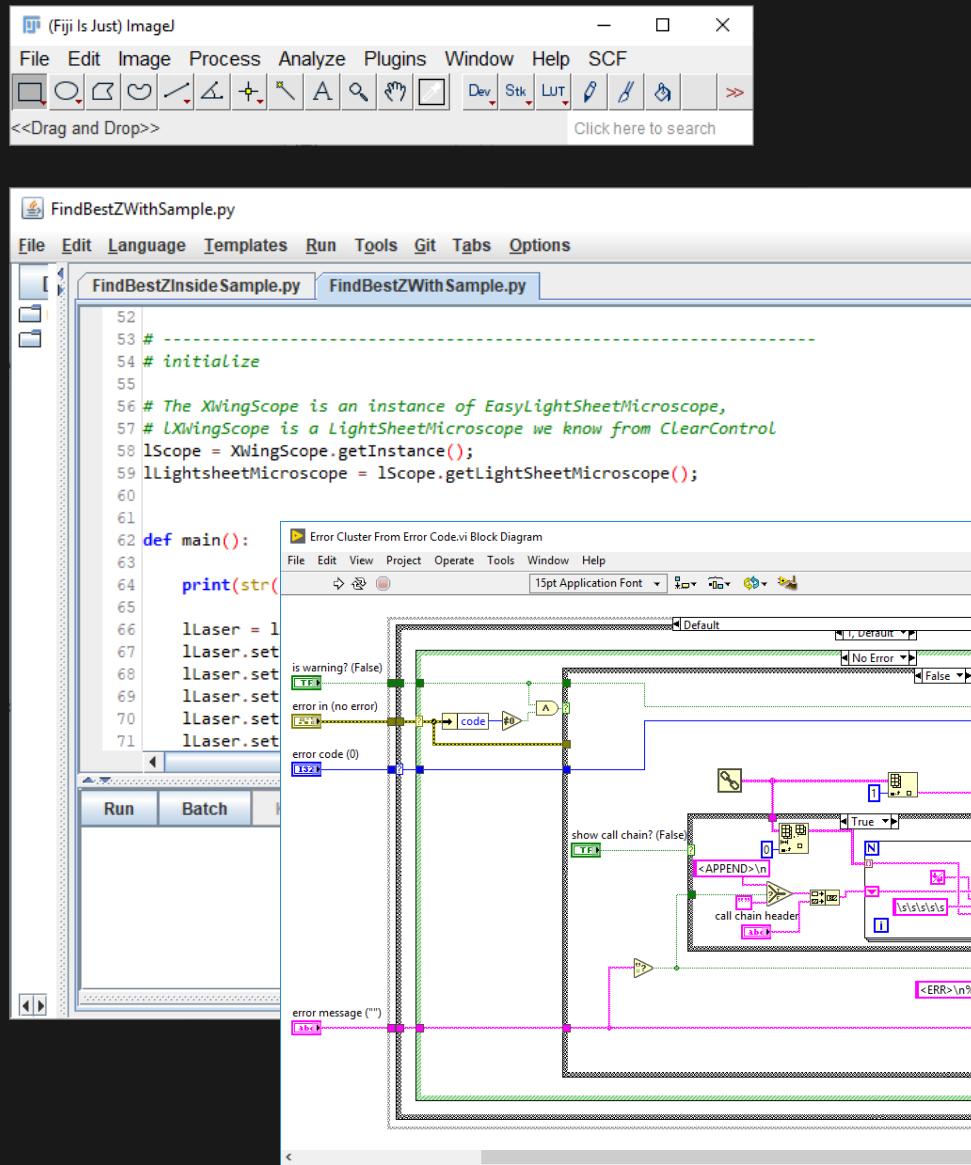


ImageJ using the GPU (laptop)  
2.2 seconds per frame  
11 min (timelapse)



ImageJ using a dedicated GPU  
(workstation)  
1 second per frame  
5 min (timelapse)

# Smart microscopy: in practice



The screenshot shows the IntelliJ IDEA interface with the project 'R2D2XWing' open. The code editor displays the 'XWingMain.java' file, which contains Java code for starting an XWing simulation. The code includes imports for `java.awt`, `java.util`, and `java.awt.event`. It defines a method `startXWing` that takes several parameters and uses a try-with-resources block to handle `ClearCL` devices. The code editor has syntax highlighting and code completion suggestions visible. On the right side, there are toolbars for navigation, search, and Git integration. The bottom right corner shows the Maven tool window with build-related options like clean, validate, compile, test, package, verify, install, site, and deploy.

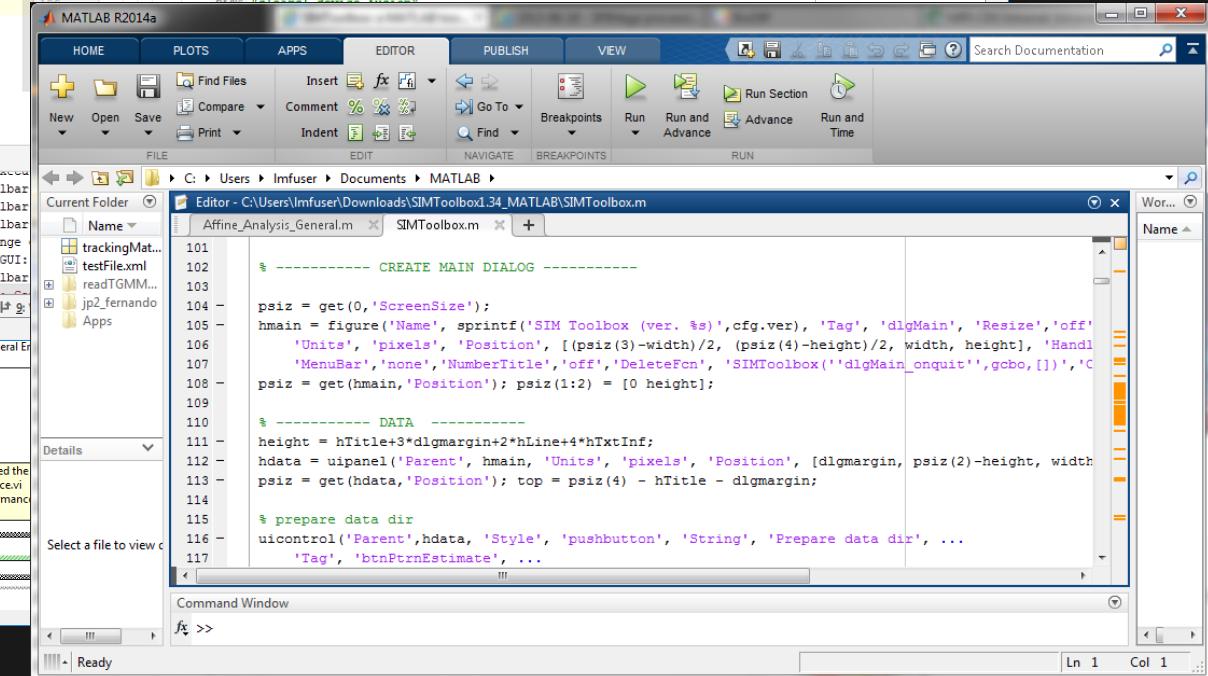
```
public XWingMicroscope startXWing(boolean pSimulation,
                                    Stage pPrimaryStage,
                                    boolean p2DDisplay,
                                    boolean p3DDisplay,
                                    boolean pUseStages)

{
    int pNumberOfDetectionArms = 2;
    int pNumberOfFlightSheets = 4;

    int lMaxStackProcessingQueueLength = 32;
    int lThreadPoolSize = 1;
    int lNumberOfControlPlanes = 8;

    try ( ClearCL lClearCL = new ClearCL(ClearCLBackends.getBestBackend()))
    {
        for (ClearCLDevice lClearCLDevice : lClearCL.getAllDevices())
            info( pFormat("OpenCL devices available: %s \n",
                         lClearCLDevice.getName()));

        ClearCLContext
        lStackFusionContext =
        lClearCL.getDeviceByName(sMachineConfiguration.getStringProperty('
        lMachineConfiguration.getMachineName());
    }
}
```



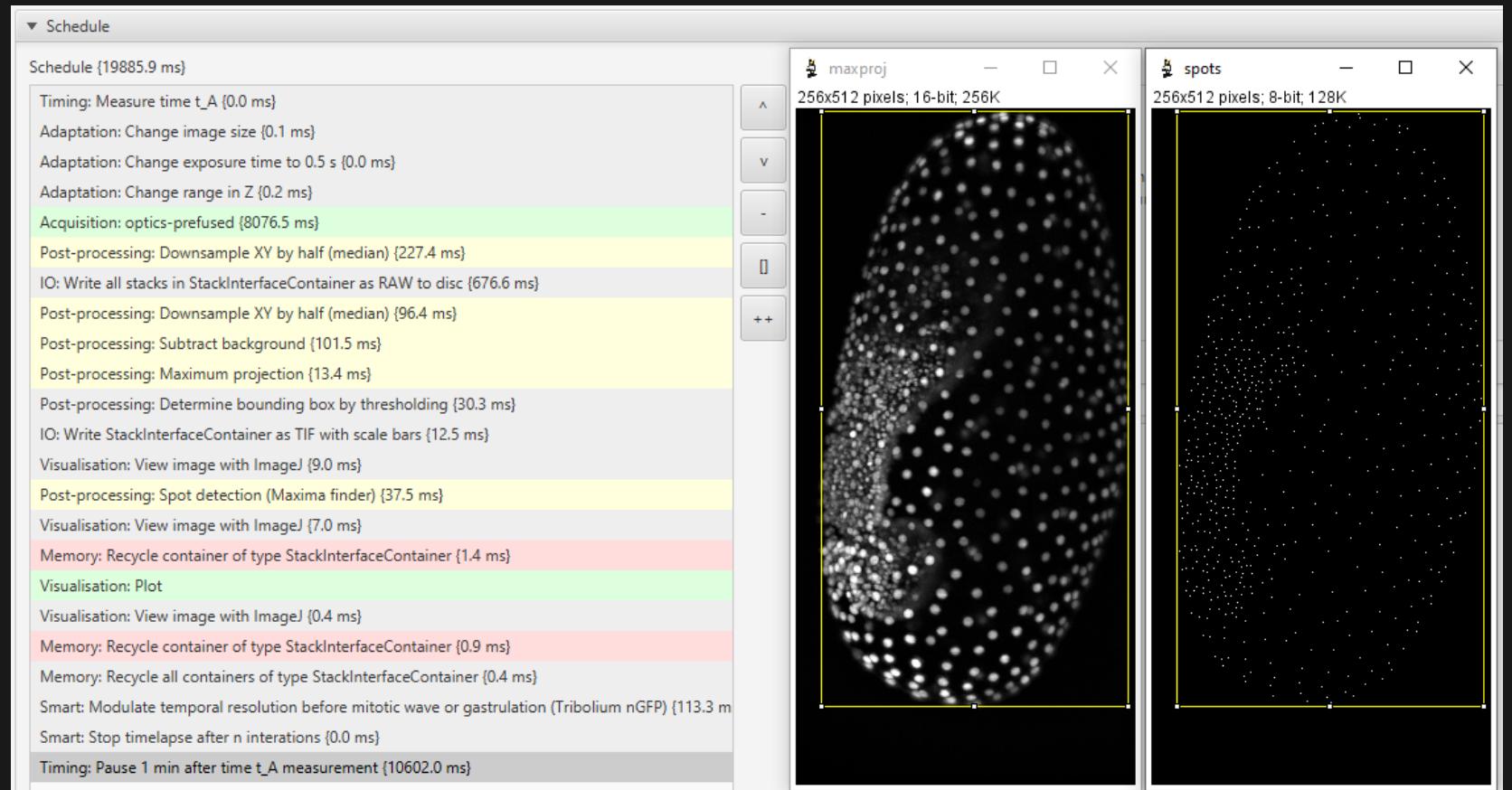
# Smart microscopy for the end user



Acquisition + I/O: 9 s

Image analysis: 0.7 s

- Downsampling
- Background subtraction
- Maximum projection
- Determine bounding box
- Spot detection

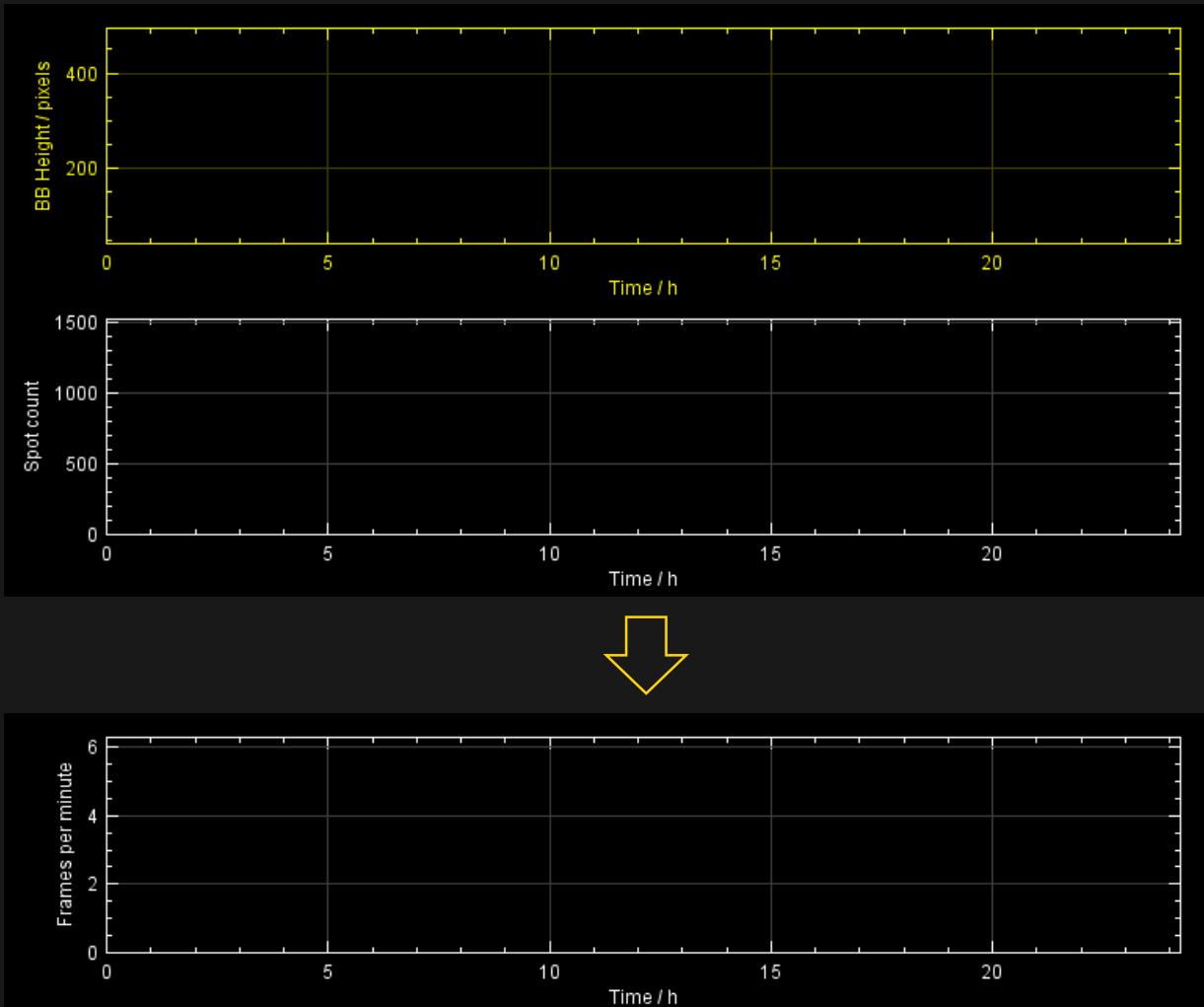


# Modulating temporal resolution



Increasing temporal detail when it matters.

- Measurements



- Frame rate

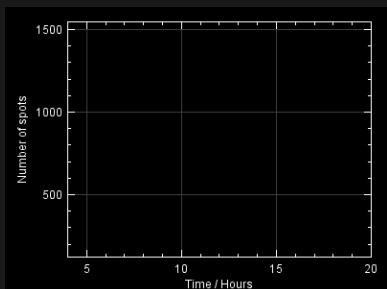
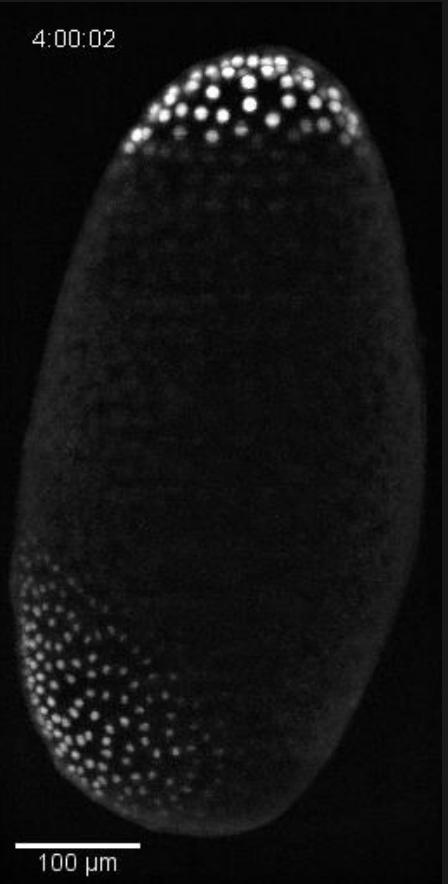


# Outlook: Complex image analysis enabled by GPU-acceleration



- Algorithmic complexity is the challenge towards real-time analysis

Complexity



nuclei-GFP,  
Background subtracted

Cylinder-max-projection  
+ spot count



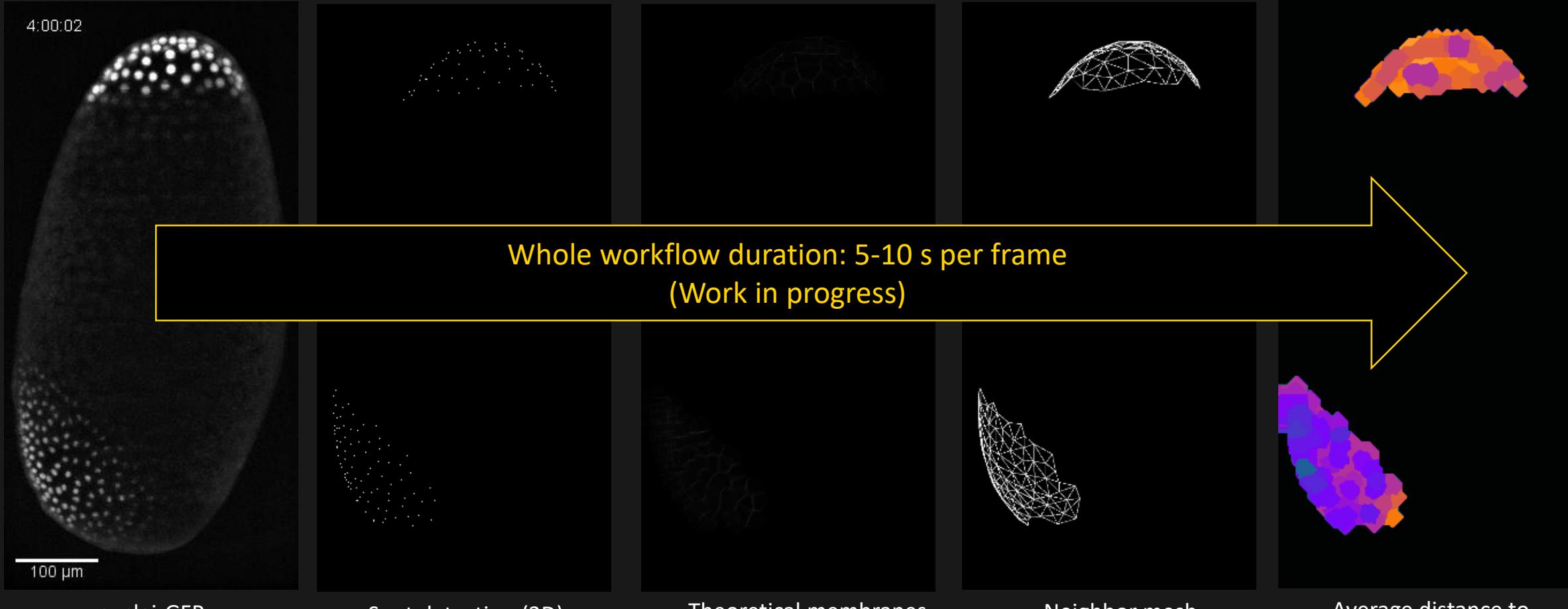
@haesleinhuepf

 <https://clij.github.io/>

# Outlook: Complex image analysis enabled by GPU-acceleration



- Algorithmic complexity is the challenge towards real-time analysis



@haesleinhuepf

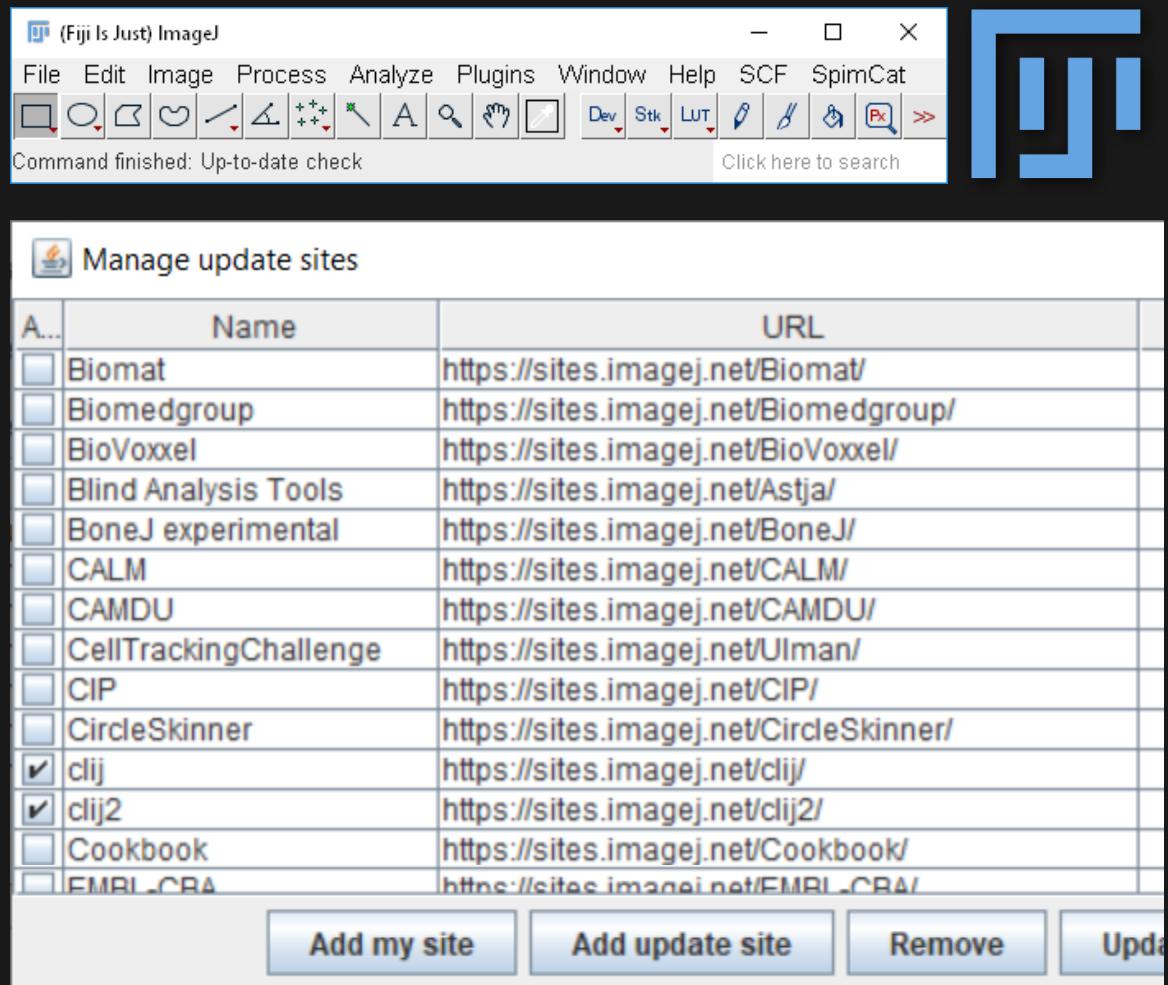
 <https://clij.github.io/>

0  35 μm

# GPU-accelerated image processing for everyone



- Just activate/enter the CLIJ update site(s)



- Online documentation

A screenshot of the CLIJ GitHub page (<https://clij.github.io>). The page features the CLIJ logo (a stylized 'c' and 'l' with a magnifying glass) and the title 'CLIJ: GPU-accelerated image processing in Fiji'. Below the title is a section titled 'Introduction' which describes CLIJ as an OpenCL - ImageJ bridge and a Fiji plugin. It mentions increased efforts in documentation, code examples, interoperability, and extensibility. CLIJ is based on ClearCL, Imglib2, ImageJ, and SciJava. There is also a section for citing the work and a forum link.

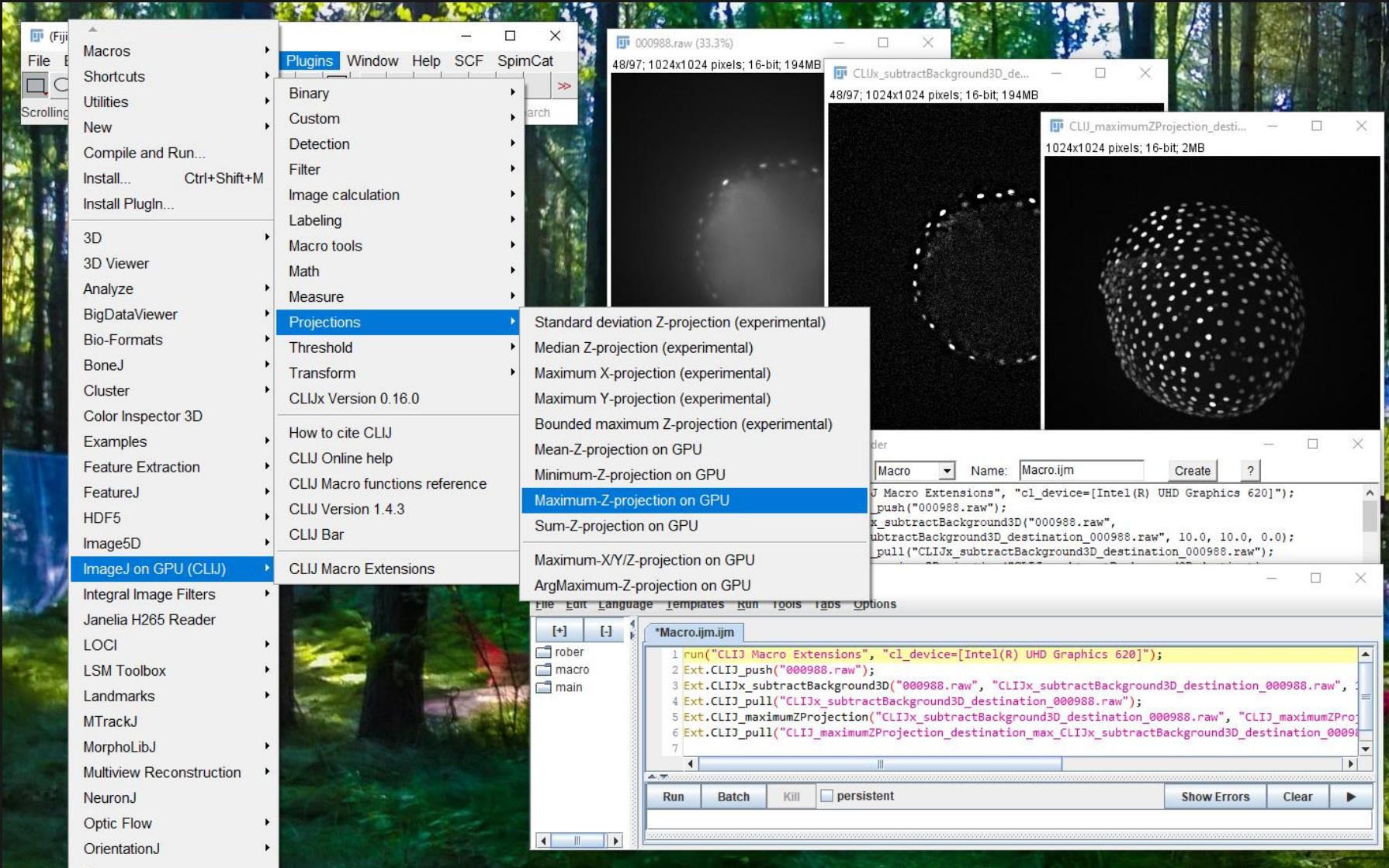
If you search for support, please open a thread on the [image.sc forum](#).

The page includes a decorative illustration of a castle with various banners and speech bubbles. One banner says 'We have many plugins', another says 'We speak Macro', and others say 'We process super fast', 'We speak OpenCL', and 'We build bridges'. The URL <https://clij.github.io> is at the bottom of the illustration.

# GPU-accelerated image processing for everyone



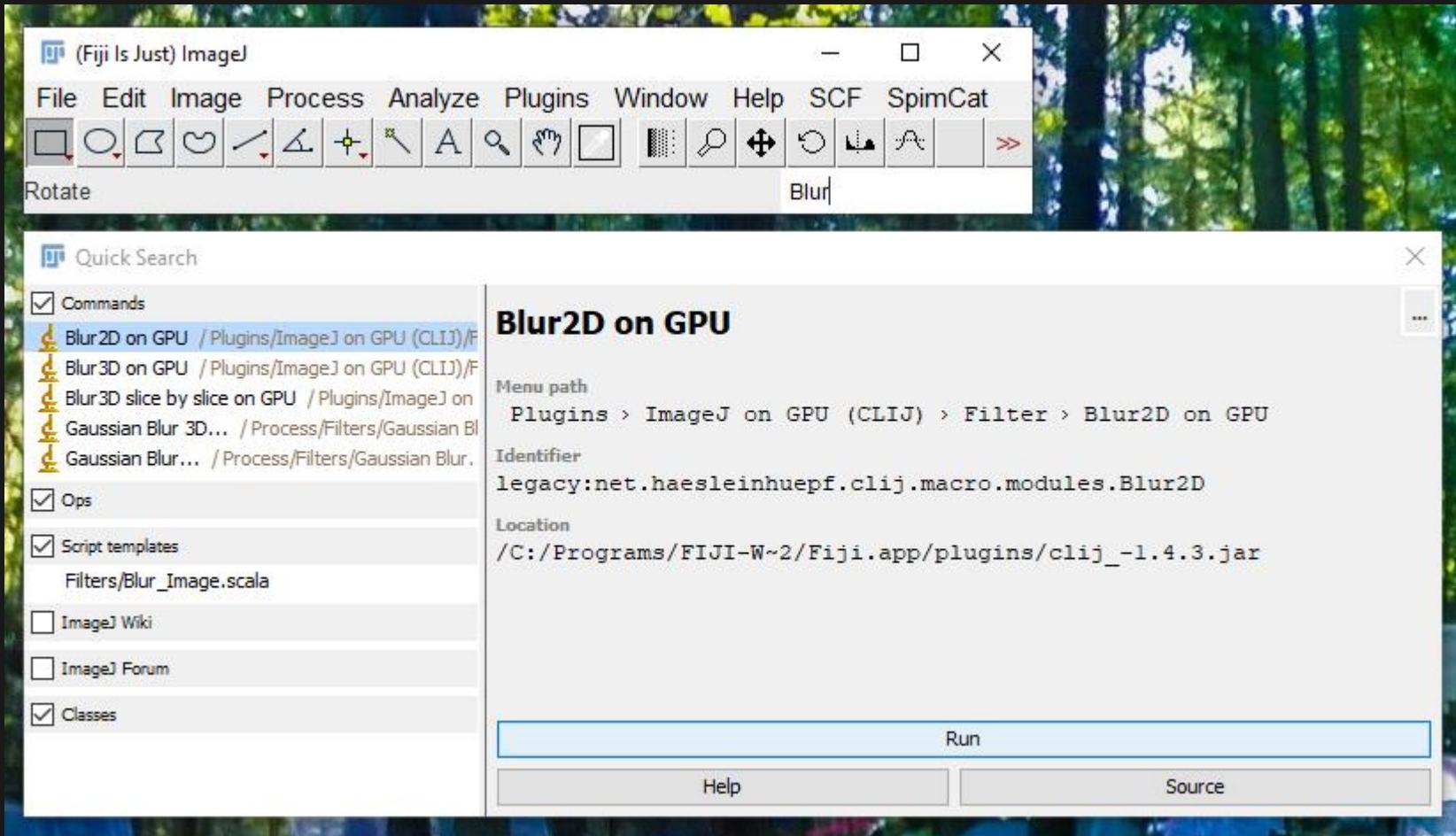
- The ImageJ macro recorder does the main part of the job!



# GPU-accelerated image processing for everyone



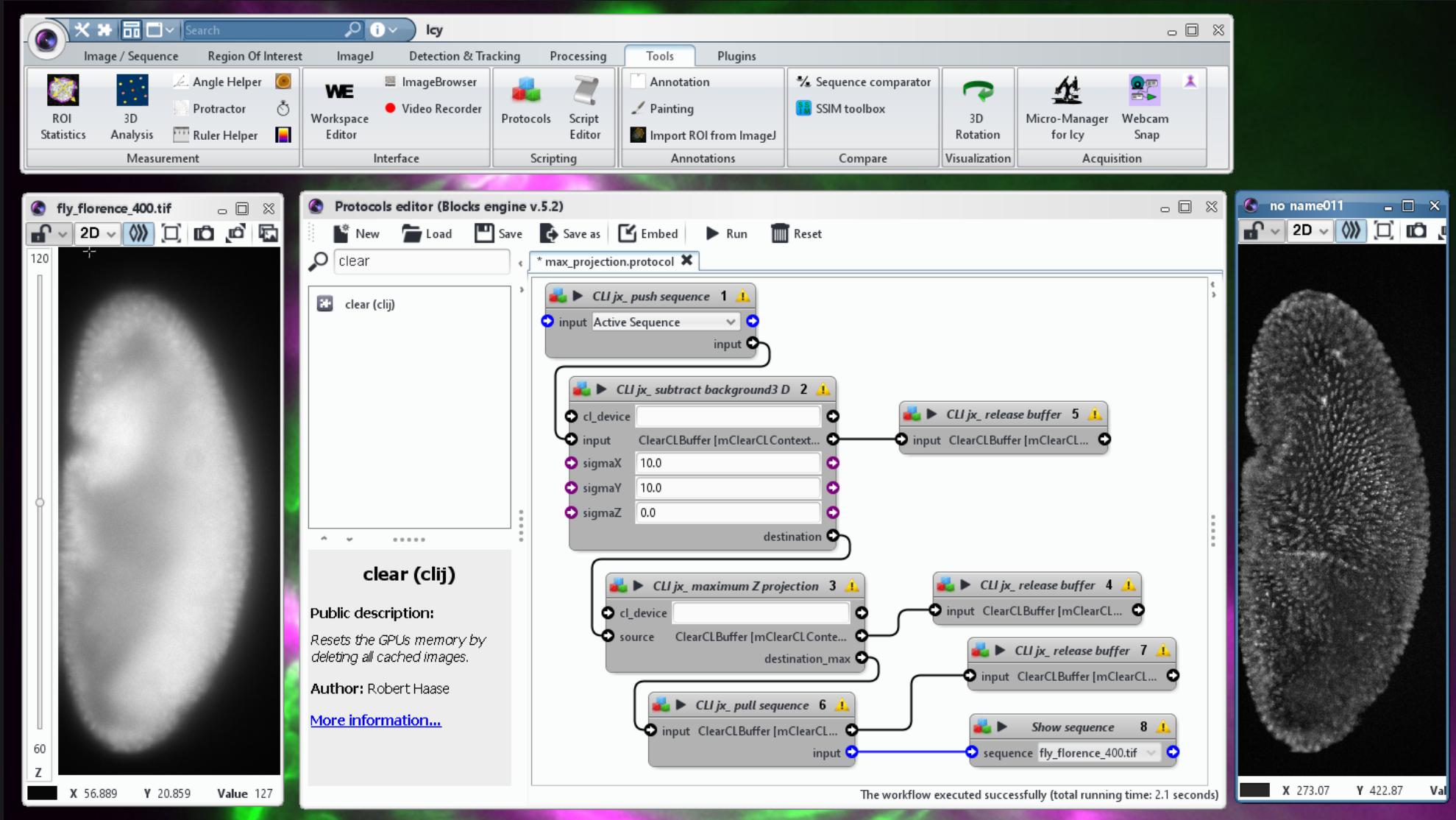
- Discover operations with Fijis search bar



# GPU-accelerated image processing for everyone



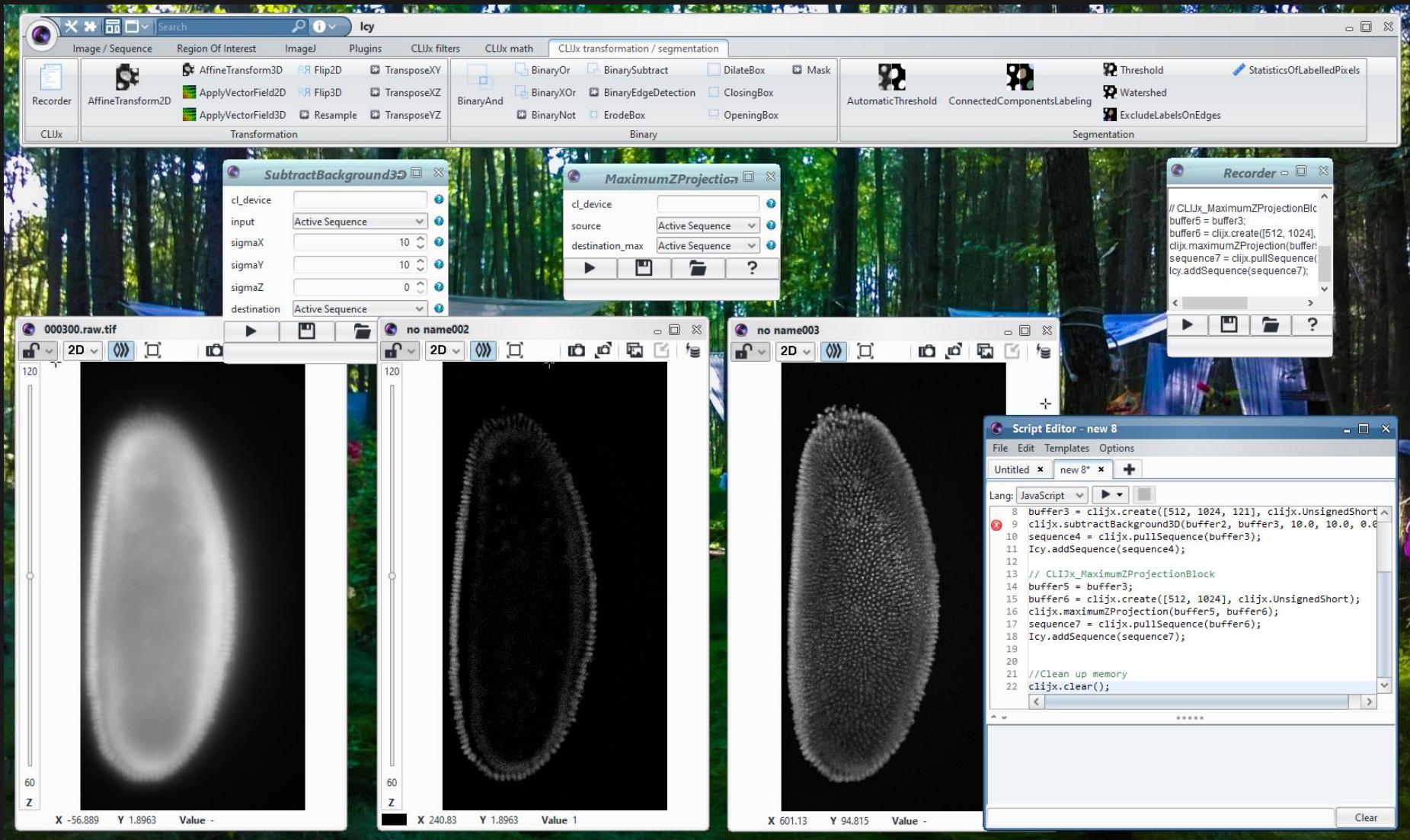
- Icy Bioimaging



# GPU-accelerated image processing for everyone



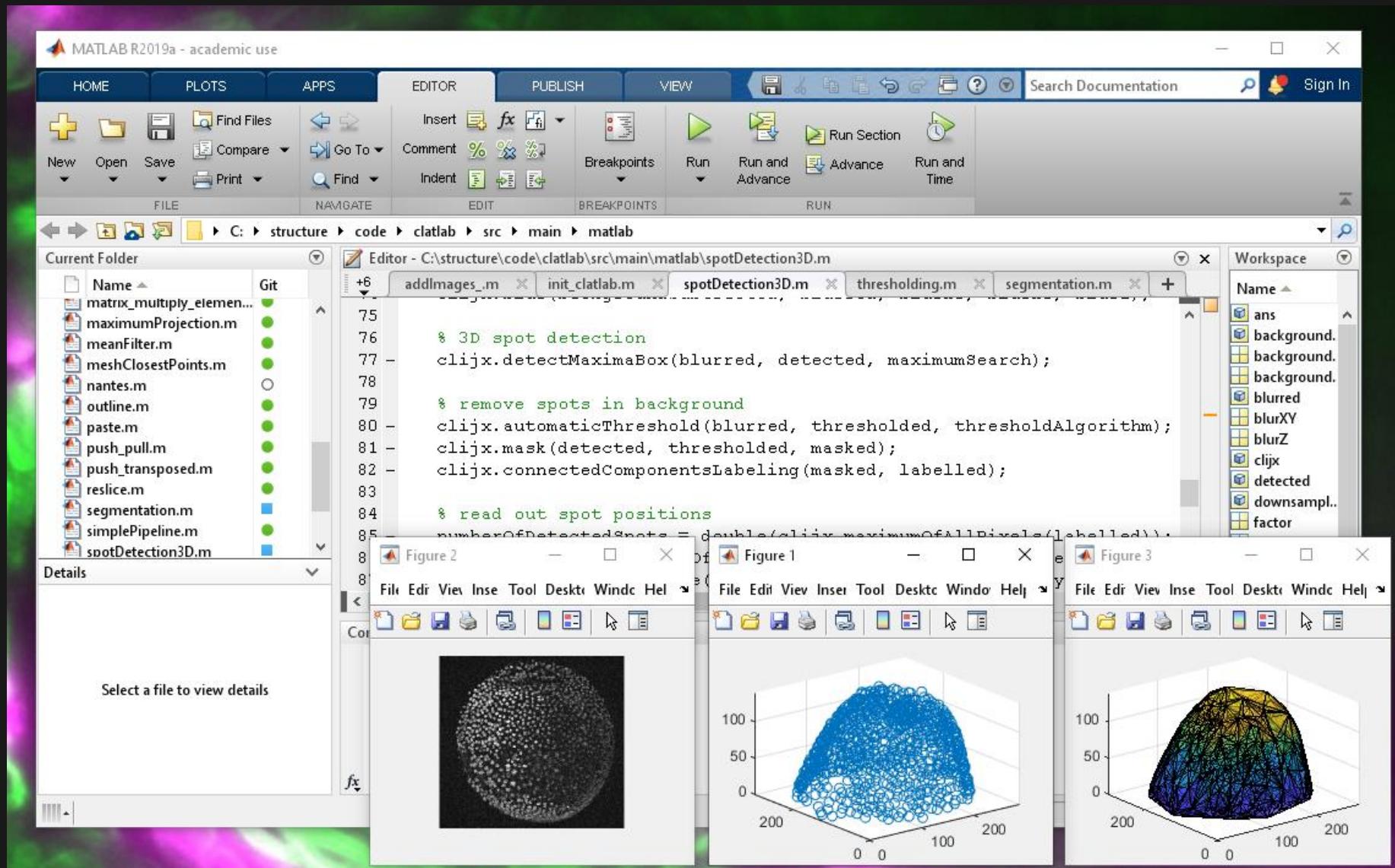
- Icy got a JavaScript recorder!



# GPU-accelerated image processing for everyone



- Try it in Matlab!



# GPU-accelerated image processing for everyone



- Python via PylImageJ

The screenshot shows a PyCharm interface with the following details:

- Project:** python C:\structure\code\clijpy
- Code Editor:** demo.py (selected)
- Code Content:**

```
python [C:\structure\code\clijpy\src\main\python] - ...\.demo.py - PyCharm
python > demo.py
demo.py x __init__.py x
18: # init clijpy to get access to the GPU
19: from jnius import autoclass
20: CLIJPY = autoclass('net.haesleinhuepf.clipy.CLIJPY')
21: clijpy = CLIJPY.getInstance()
22:
23: # convert array to an ImageJ2 img:
24: import numpy as np
25: np_arr = np.array(sk_img)
26: ij_img = ij.py.to_java(np_arr)
27:
28: # push the image to the GPU
29: input = clijpy.push(ij_img)
30: # allocate memory for the result image
31: output = clijpy.create(input)
32:
33: # blur the image
34: Float = autoclass('java.lang.Float')
35: clijpy.op.blur(input, output, Float(5.0), Float(5.0))
36:
37: # pull image back from GPU
38: ij_img_result = clijpy.pull(output)
39: # convert to numpy/python
40: np_arr_result = ij.py.rain_to_numpy(ij_img_result)
41:
42: # show the input and the result image
```
- Tool Window:** Figure 1 (displaying two heatmaps)
- Terminal:** Local  
(imagej) C:\structure\code\clijpy\src\main\python>python demo.py  
Added 403 JARs to the Java classpath.
- Bottom Bar:** Python Console, Terminal, TODO, Event Log

# GPU-accelerated image processing for everyone



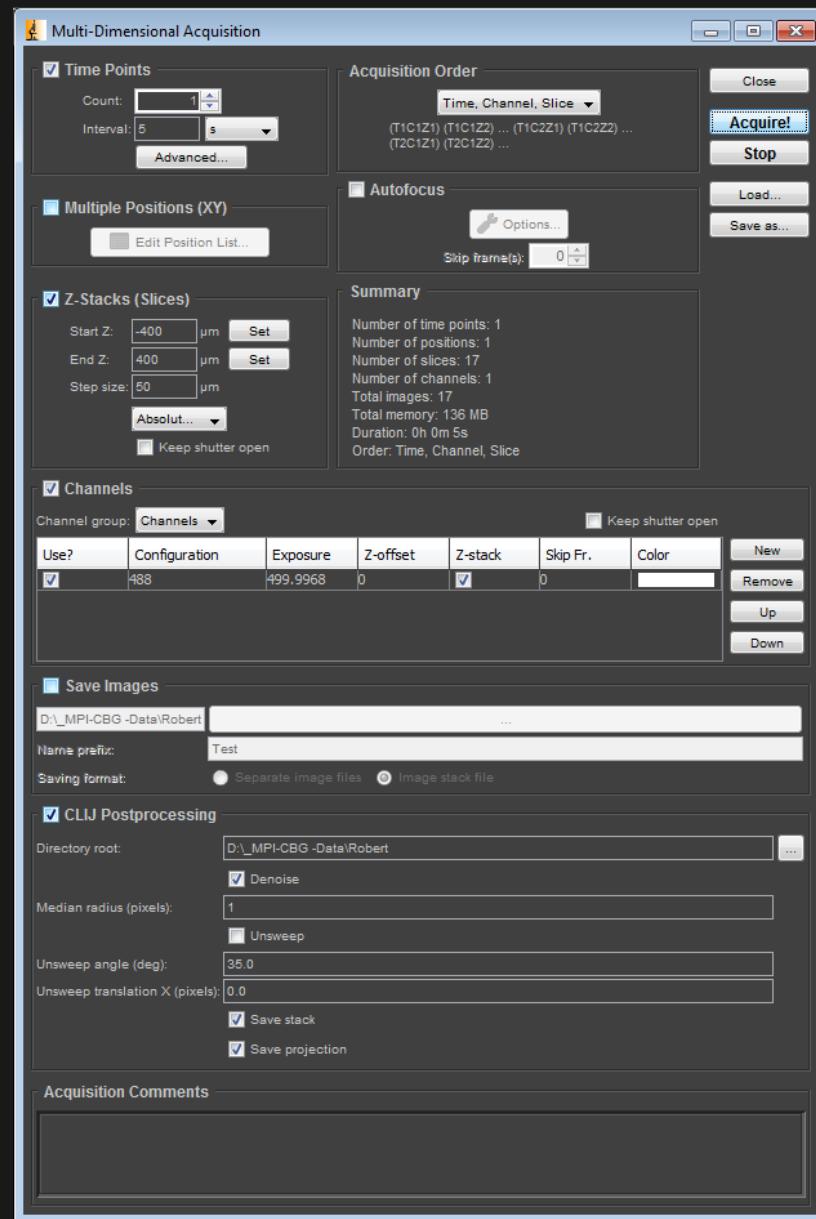
- Available in the Zeiss Apeer cloud service: <https://github.com/clij/clij-apeer-template>

The screenshot shows the Zeiss Apeer cloud service interface. On the left, the 'My Modules' page displays the 'clij-apeer-template' module, which is a 'Public' draft created by Robert Haase on August 24, 2019. The module summary states it's a template for GPU-accelerated image processing using CLIJ, applying Gaussian blur to an image for demo purposes. It includes links to its GitHub repository and buttons for 'Create Workflow' and 'Publish Module'. On the right, a separate window titled 'My Workflows - My Workspace' shows a successful workflow named 'clij-apeer-demo' with a result named 'output\_image.jpg'. The workflow results page displays a blurred image preview.

# GPU-accelerated image processing for everyone



- Work in progress:  
MicroManager integration



# Support: Image.sc



Image.sc Forum x + https://forum.image.sc

image.sc search menu profile

Community Partners

All Topics	Bio-Formats	BoneJ	CellProfiler	CLIJ	Cytomine
DeepLabCut	Fiji	IDR	ilastik	ImageJ	ImagePy
ImgLib2	ImJoy	MIB	MiToBo	NEUBIAS	OME
OMERO	OpenSPIM	Orbit	QuPath	Scenery	SCIFIO
scikit-image	SciView	SLIM Curve	...	Your Icon Here	

all categories ▾ all tags ▾ all ▾ Latest New (45) Unread (110) Top Categories Unanswered + New Topic

# Acknowledgements



Dani Vorkel  
(Myers lab)  
@happifocus



Alex Dibrov  
(Myers lab)  
@a\_dibrov



Uwe Schmidt  
(Myers lab)  
@uschmidt83



Martin Weigert  
(now at EPFL)  
@martweig



David Chen  
(Myers lab)  
@bigimaginglab



Debayan Saha  
(Myers lab)  
@debayan102



Gene Myers  
@TheGeneMyers



Nicola Maghelli  
(Advanced Imaging Facility)  
@aif\_cbg



Loic A. Royer  
(now at CZ Biohub)  
@loicaroyer



Johannes Girstmair  
(Tomancak lab)  
@jogirstmair



Akanksha Jain  
(now Treutlein lab)  
@jain\_akanksha\_



Pavel Tomancak  
@PavelTomancak



Deborah Schmidt  
(Jug lab)  
@frauuzfall



Florian Jug  
@florianjug



<https://fiji.sc>



@haesleinhuepf

<https://image.sc>



<https://clij.github.io/>

HZDR

- Peter Steinbach

MPI CBG Core Facilities

- Advanced Imaging Facility
- Light Microscopy Facility
- Scientific Computing
- IT Department
- Fly Facility

Community contributors / testers

- Alex Herbert (University of Sussex),
- Bram van den Broek (Netherlands Cancer Institute),
- Brenton Cavanagh (RCSI),
- Brian Northan (True North Intelligent Algorithms),
- Bruno C. Vellutini (MPI CBG),
- Curtis Rueden (UW-Madison LOCI),
- Damir Krunic (DKFZ),
- Daniel J. White (GE),
- Gaby G. Martins (IGC),
- Guillaume Witz (Bern University),
- Siân Culley (MRC LMB),
- Giovanni Cardone (MPI Biochem),
- Jan Brocher (Biovoxxel),
- Jean-Yves Tinevez (Institute Pasteur),
- Juergen Gluch (Fraunhofer IKTS),
- Kota Miura,
- Laurent Thomas (Acquier),
- Matthew Foley (University of Sydney),
- Nico Stuurman (UCSF),
- Peter Haub,
- Pete Bankhead (University of Edinburgh),
- Pradeep Rajasekhar (Monash University),
- Ruth Whelan-Jeans,
- Tanner Fadero (UNC-Chapel Hill),
- Thomas Irmer (Zeiss),
- Tobias Pietzsch (MPI-CBG),
- Wilson Adams (VU Biophotonics)

Funding:



Federal Ministry  
of Education  
and Research