From Images to Knowledge with ImageJ & Friends

virtual conference

Nov 30 - Dec 2, 2020

Stephan Preibisch, Stephan Saalfeld, Anna Kreshuk, Pavel Tomancak and Virginie Uhlmann

hhmi | janelia Conferences

Designing GPU-accelerated Image Data Flow Graphs for CLIJ2 and clEsperanto and deployment to ImageJ, Fiji, Icy, Matlab, QuPath, Python, Napari and C++

Tutors: Robert Haase (rhaase@mpi-cbg.de)
Stéphane Rigaud (stephane.rigaud@pasteur.fr)

Session 1: 2020-12-01 16:00 UTC – 2020-12-01 20:00 UTC
Session 2: 2020-12-02 09:00 UTC – 2020-12-02 13:00 UTC

Designing GPU-accelerated Image Data Flow Graphs for CLIJ2 and clEsperanto and deployment to ImageJ, Fiji, Icy, Matlab, QuPath, Python, Napari and C++

Robert Haase[123], Stéphane Rigaud[4]

1 Center for Systems Biology Dresden

2 Max Planck Institute for Molecular Cell Biology and Genetics Dresden

3 Physics of Life Cluster of Excellence, TU Dresden

4 Image Analysis Hub, C2RT, Institut Pasteur, Paris

Abstract

The current rise of graphics processing units (GPUs) in the context of image processing boosts the need for accessible tools for building GPU-accelerated image analysis workflows. Typically, designing data analysis procedures utilizing GPUs involves coding skills and knowledge of GPU-specific programming languages such as the Open Computing Language (OpenCL) [1]. For facilitating end-user access to modern computing hardware such as GPUs, the CLIJ platform [2] was developed and documented in detail [3, 4]. It targets programming beginners through an abstraction layer that allows to call GPU-accelerated image processing operations without the need for learning a new programming language such as OpenCL.

To further lower the entrance bounden, a recently introduced a graphical user interface for the Fiji platform [5] utilizes an image data flow graph for designing image processing workflows interactively on screen [6]. From a configured graph, the user can generate scripts reproducing the data flow in scripting languages such as ImageJ Macro, Fiji Groovy, ImageJ JavaScript, Fiji Jython, Icy [7] JavaScript and Matlab. Furthermore, experimentalists among the workshop attendees are welcome to try the upcoming integration with QuPath [8] Groovy scripting. Attendees can also get a first impression of C++, Python, Jupyter and Napari [9] integration which is currently developed under the umbrella of the clEsperanto project [10]. A Fiji-plugin generator utilizing Maven and Java and a Icy Protocol generator are also ready for testing. Fans of the ImageJ Macro language can test the upcoming integration with ImageJ-Macro-Markdown [11] notebooks.

Workshop attendees with skill sets on multiple levels are welcome: On the one hand, the exercises can be accomplished with basic ImageJ Macro programming skills. On the other hand, the attendees can learn how to generate and modify Java code and compiling it for example as plugin for Fiji.

[1] https://www.khronos.org/opencl/

[2] https://doi.org/10.1038/s41592-019-0650-1

[3] https://arxiv.org/abs/2008.11799

[4] https://clij.github.io

[5] https://fiji.sc

[6] https://clij.github.io/assistant

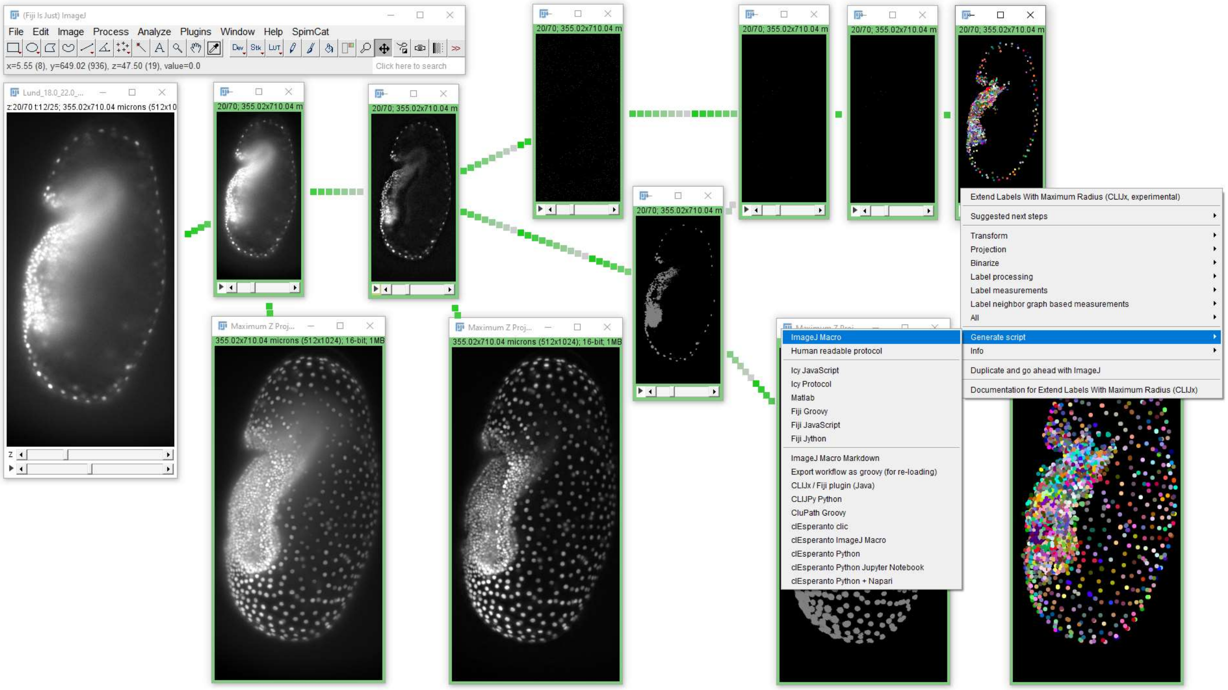
[7] http://icy.bioimageanalysis.org/

[8] https://qupath.github.io/

[9] https://napari.org/

[10] https://clesperanto.github.io/

[11] https://github.com/haesleinhuepf/imagejmacromarkdown


(Fiji Is Just) ImageJ
File Edit Image Process Analyze Plugins Window Help SpimCat
x=5.55 (8), y=649.02 (936), z=47.50 (19), value=0.0
Click here to search
Maximum Z Proj...
355.02x710.04 microns (512x1024); 16-bit; 1MB
Extend Labels With Maximum Radius (CLIJx, experimental)
Suggested next steps
Transform
Projection
Binarize
Label processing
Label measurements
Label neighbor graph based measurements
All
Generate script
Info
Duplicate and go ahead with ImageJ
Documentation for Extend Labels With Maximum Radius (CLIJx)
ImageJ Macro
Human readable protocol
Icy JavaScript
Icy Protocol
Matlab
Fiji Groovy
Fiji JavaScript
Fiji Jython
ImageJ Macro Markdown
Export workflow as groovy (for re-loading)
CLIJx / Fiji plugin (Java)
CLIJPy Python
CluPath Groovy
clEsperanto clic
clEsperanto ImageJ Macro
clEsperanto Python
clEsperanto Python Jupyter Notebook
clEsperanto Python + Napari

Preparation (homework)

Before attending the workshop, attendees are asked to install requirements on their computers which depend on the platform they want to work with. Minimum requirement is Fiji with the update sites “clij”, “clij2” and “clijx-assistant” activated. More information can be found on the website: https://clij.github.io/assistant/installation

The NEUBIAS Academy seminar about GPU-Accelerated image processing using CLIJ2 is a worthwhile preparation for this course: https://www.youtube.com/watch?v=uMj0OS1TiQE

If attendees would like to generate CLIJ-Fiji plugins during the exercise, they might want to learn how make Fiji plugins in advance: https://www.youtube.com/watch?v=YIWpoBnnLio

Outline

- Introduction to GPU-accelerated image processing using CLIJ2 and clEsperanto (30 min)
    - ImageJ Macro (10 min)
    - Python (10 min)

  - C++ (15 min)
- Exercise (30 Min)
- Live-demo of interactive data flow graph design for image processing and deployment to various platforms (20 min)
  - Fiji (5 min)
  - Icy (5 min)
  - QuPath (5 min)
  - Napari (5 min)
- Exercises (60 min)
- Work on your own data (60 min)

Time zone availability

The course can be held between 10 am and 8 pm CET – ideally not the same day :D

About the tutors

Robert Haase is computer scientist by training and focuses on accessible GPU-accelerated image processing, smart autonomous microscopy and image data mining in the life sciences. He received a PhD from the Faculty of Medicine Carl Gustav Carus of the TU Dresden for his work on swarm intelligence based algorithms for medical image segmentation in the cancer research context. He is lecturer for bio-image analysis, bio-statistics and programming at the Biotechnology Center of the TU Dresden. In his postdoctoral research in Gene Myers lab at the Center for Systems Biology and the Max Planck Institute for Molecular Cell Biology and Genetics he concentrated on bridging the disciplines computer science and biology to forward understanding of how tissues and organisms form.	Stéphane Rigaud is computer scientist by formation and specialized himself on Image processing and analysis applied to bio-medical images. He received a Ph.D. from the University of Pierre And Marie Curie, Paris 6 for his work on Neural Stem Cell proliferation analysis and modeling during neurosphere growth in vitro. Following his interest to biology, he worked on quantification and characterization of tissue development and deformation in vivo at the Institut Curie, using the D.Melanogaster model. He recently joined the Image Analysis Hub of the Institut Pasteur of Paris to provide his expertise in bio-image analysis to the Institute.