



# *multiview-stitcher:* An Interoperable Python Toolbox for Image Registration and Fusion

Virtual tutorial @ Virtual I2K 2024

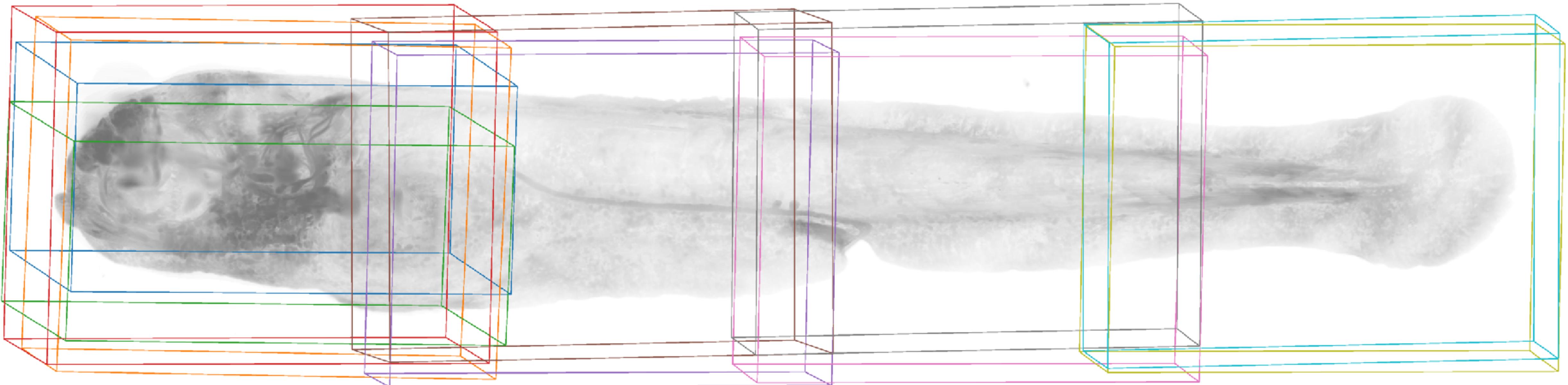


Marvin Albert

@m-albert

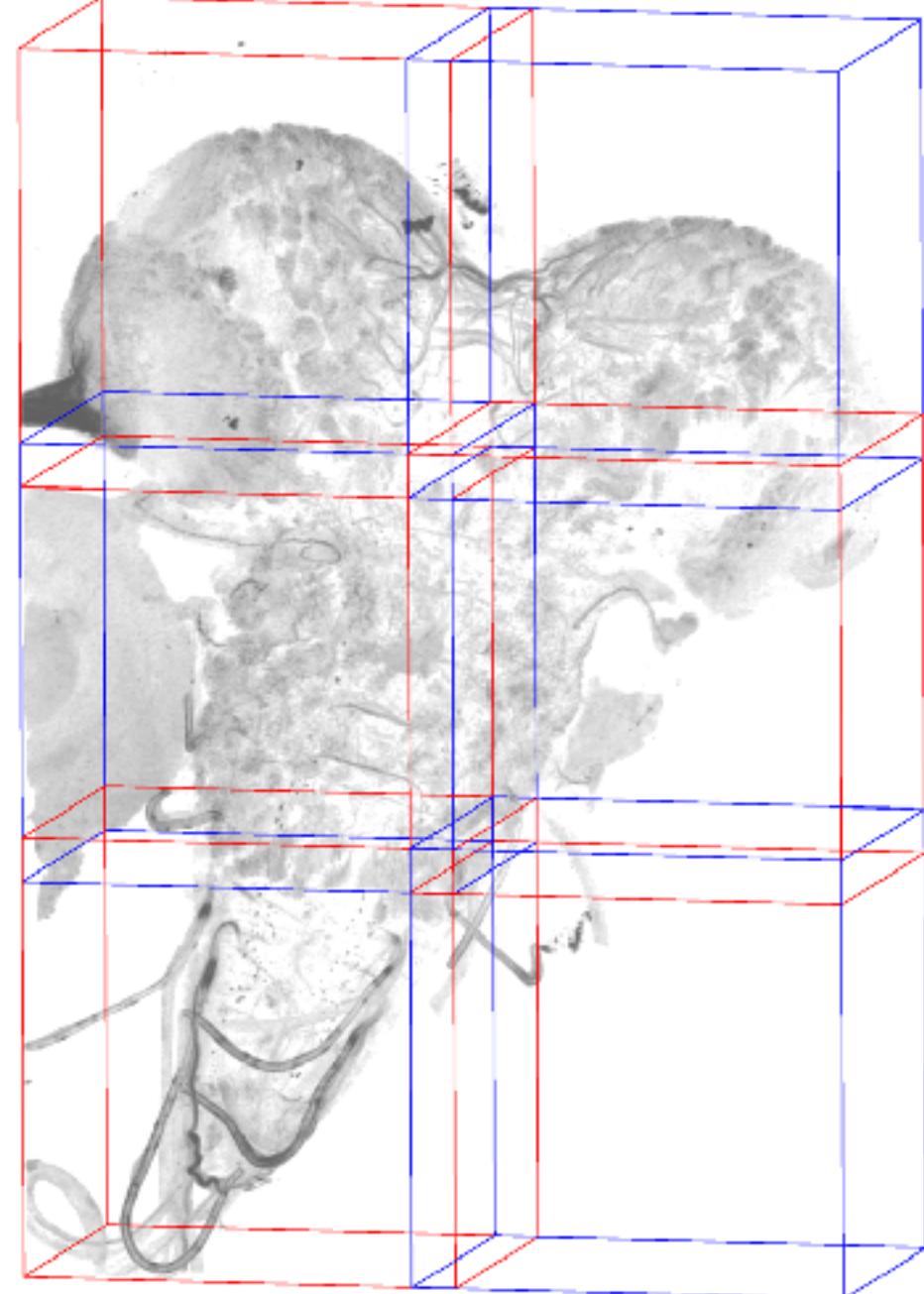
X \_malberto

Image Analysis Hub

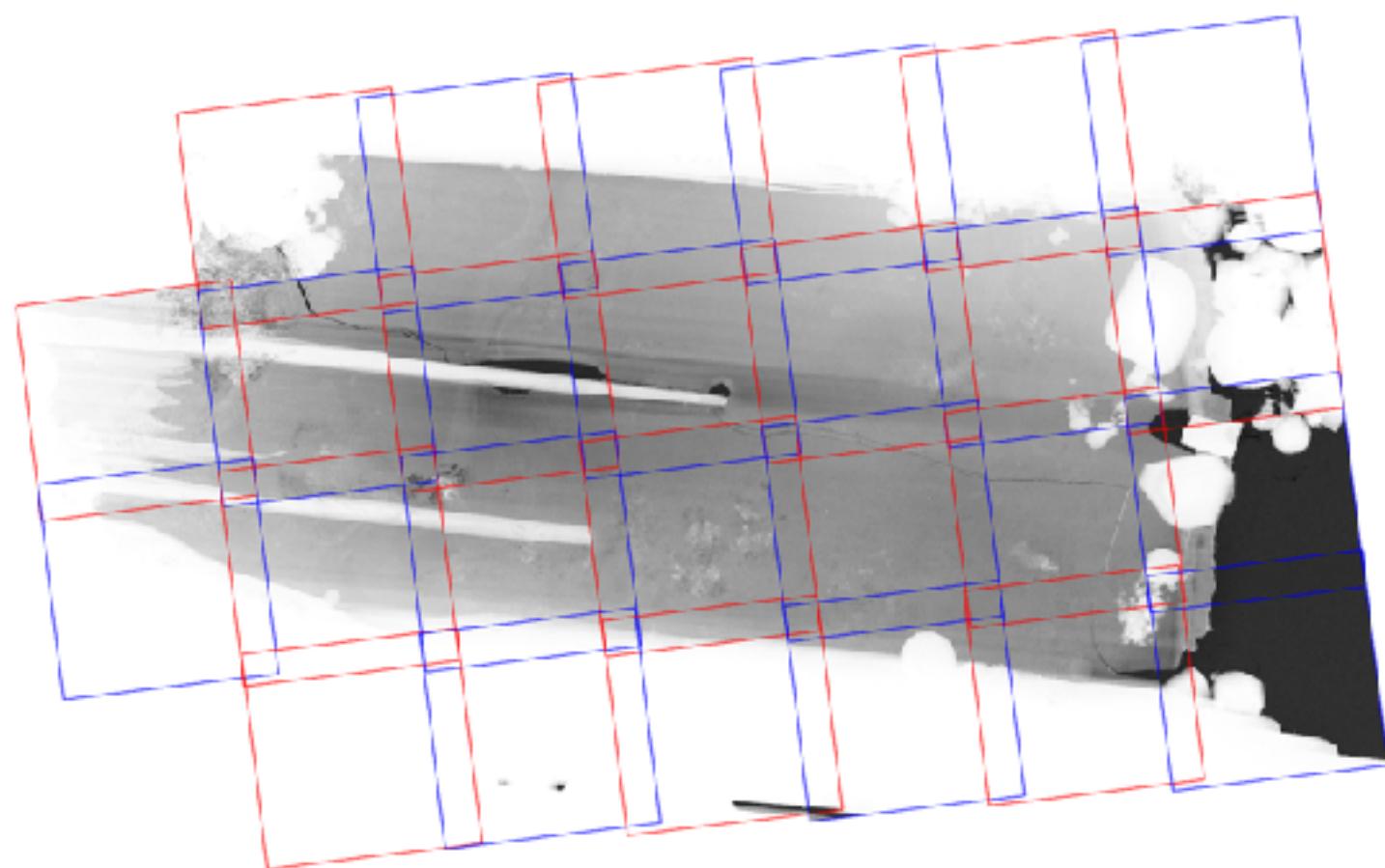


# Reconstructing multi-tile/view image modalities

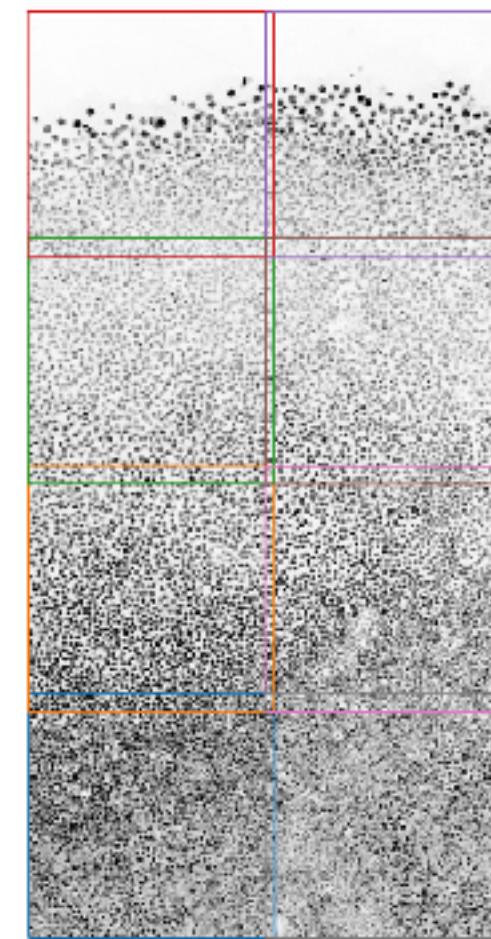
Multi-positioning / map acquisitions



BigStitcher example dataset

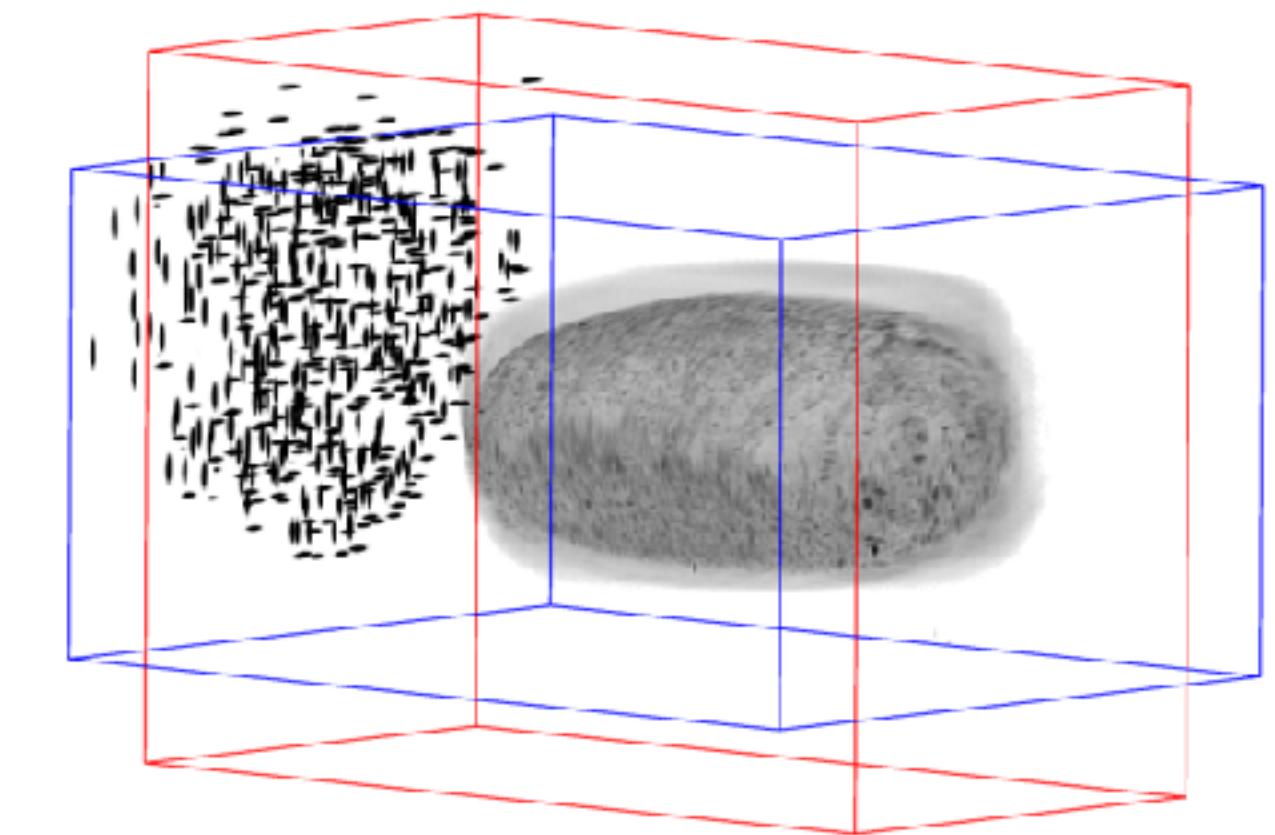


Cryo-EM Map



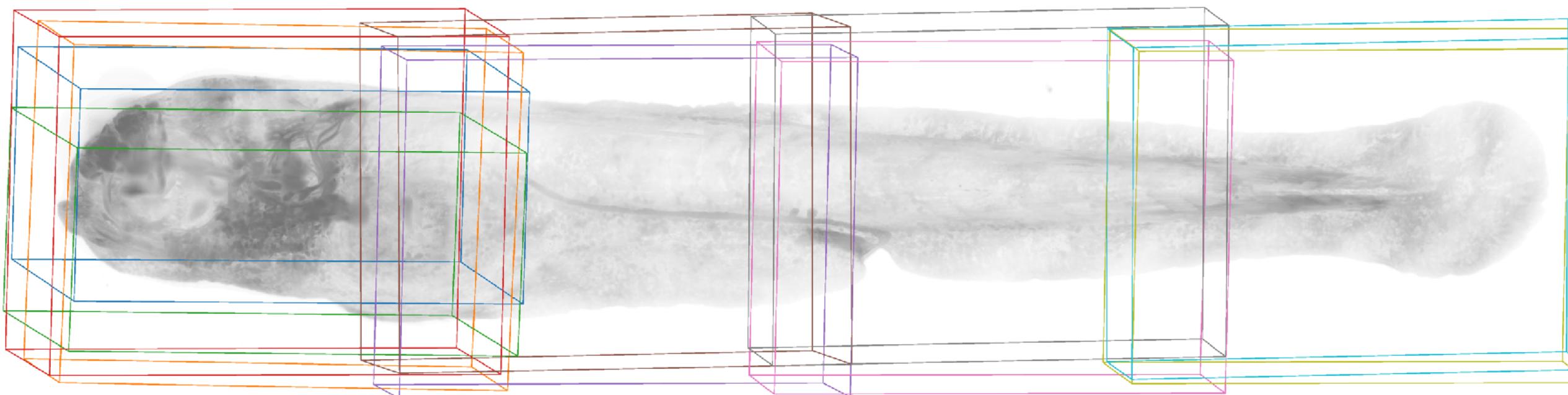
Multi-positioning confocal

Multi-view acquisitions



Multi-view light sheet microscopy

Multi-tile + multi-view



Tiled multi-view light sheet microscopy

# Tile/view registration

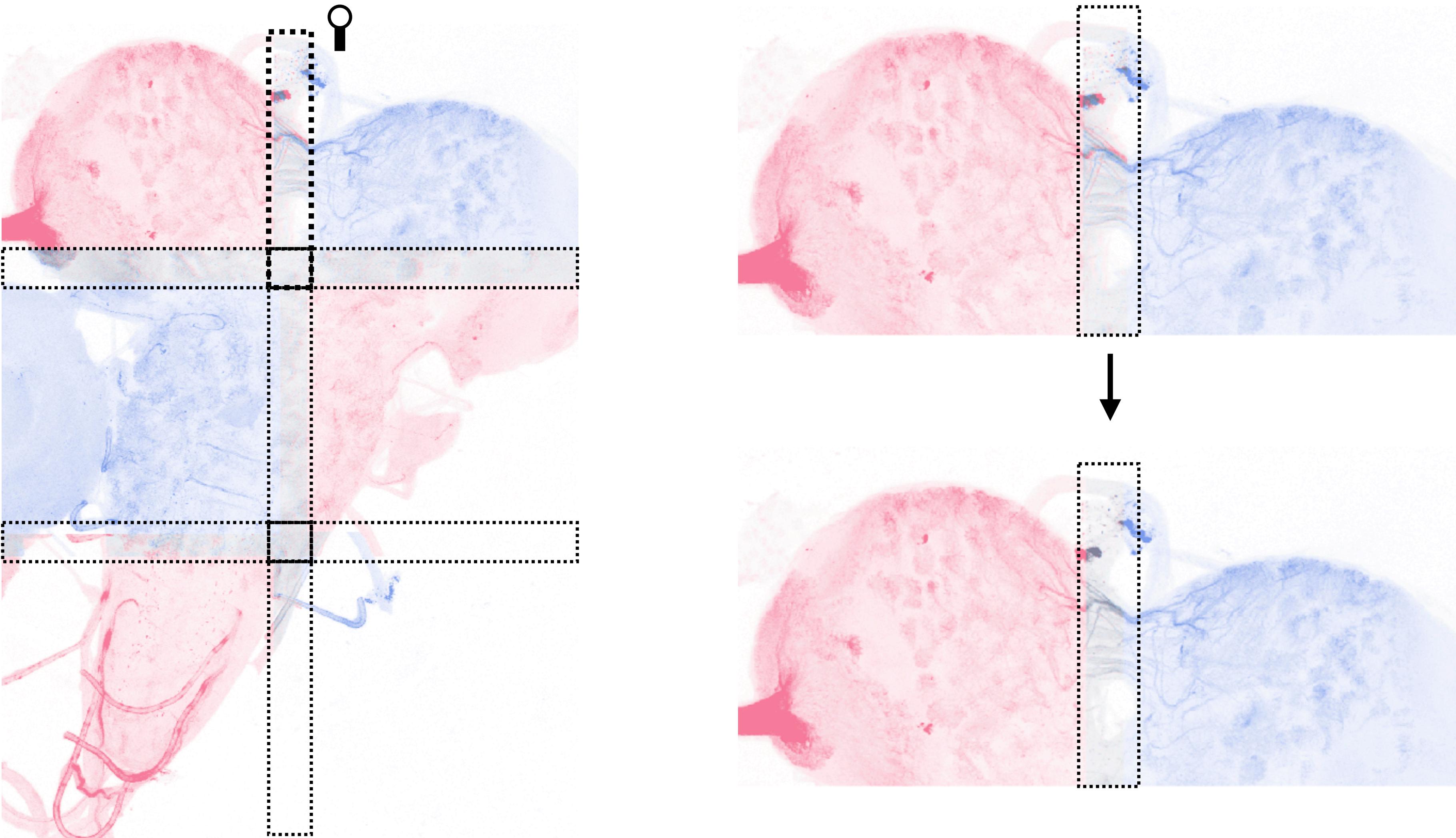


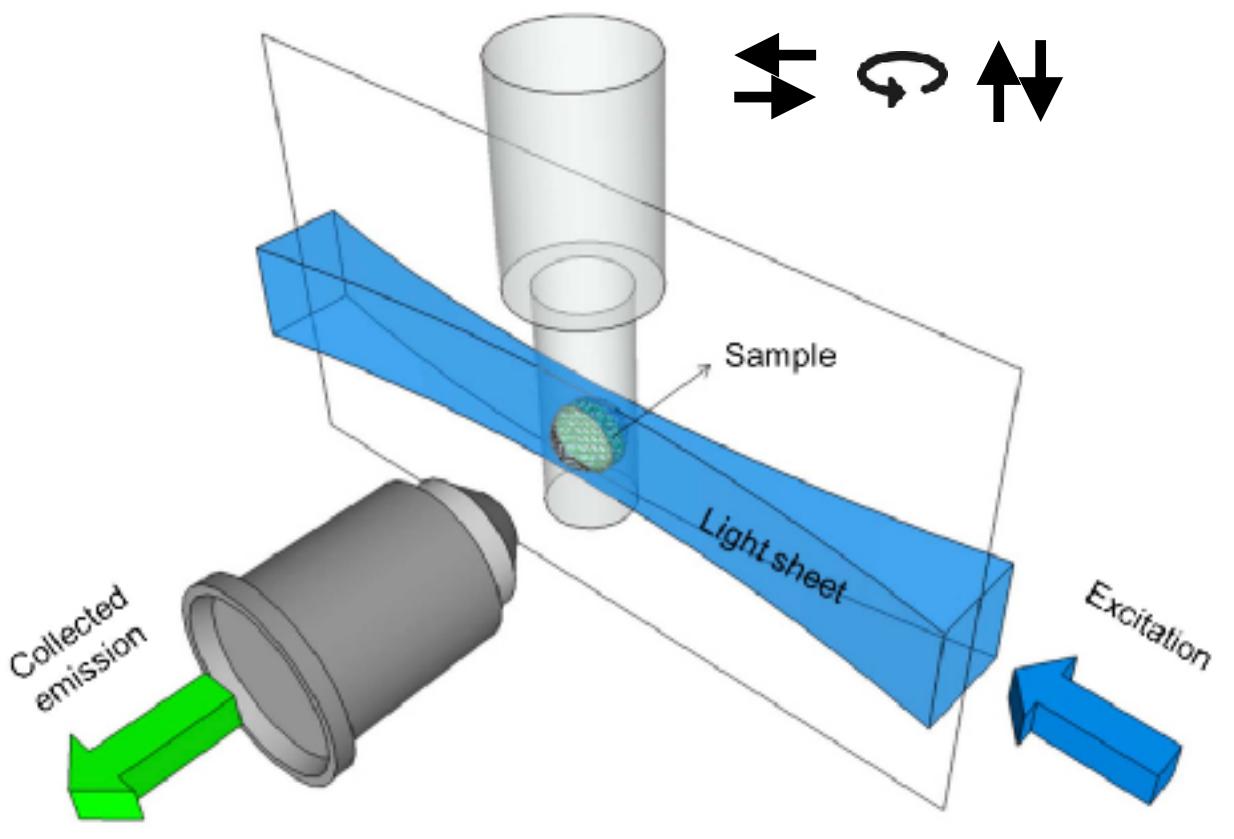


Image  
fusion



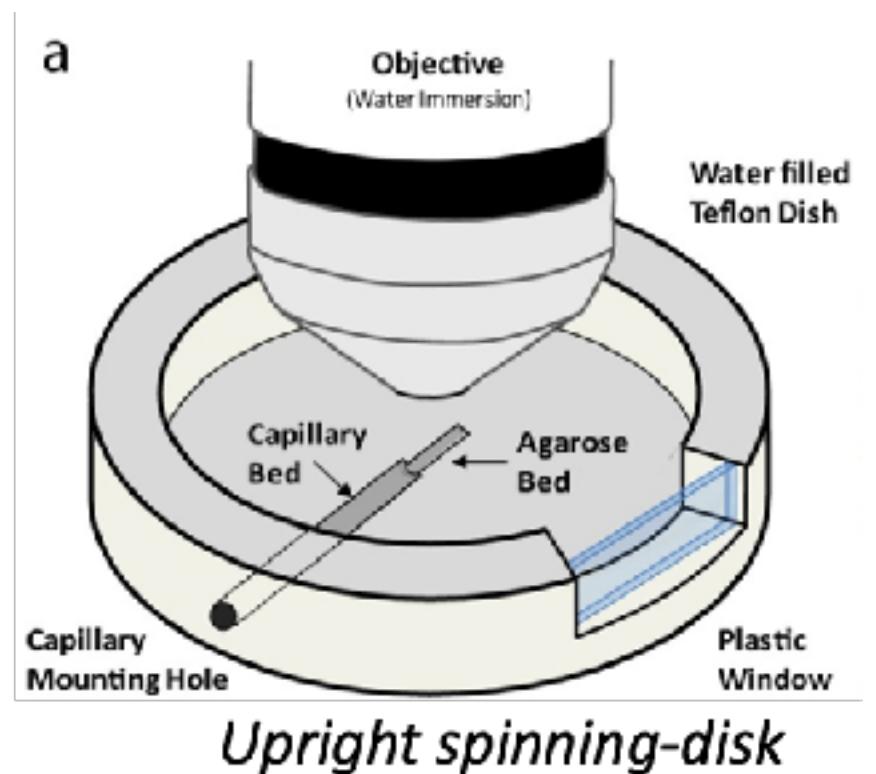
# (Multi-view) microscopy with opaque samples

Light-sheet  
microscopy

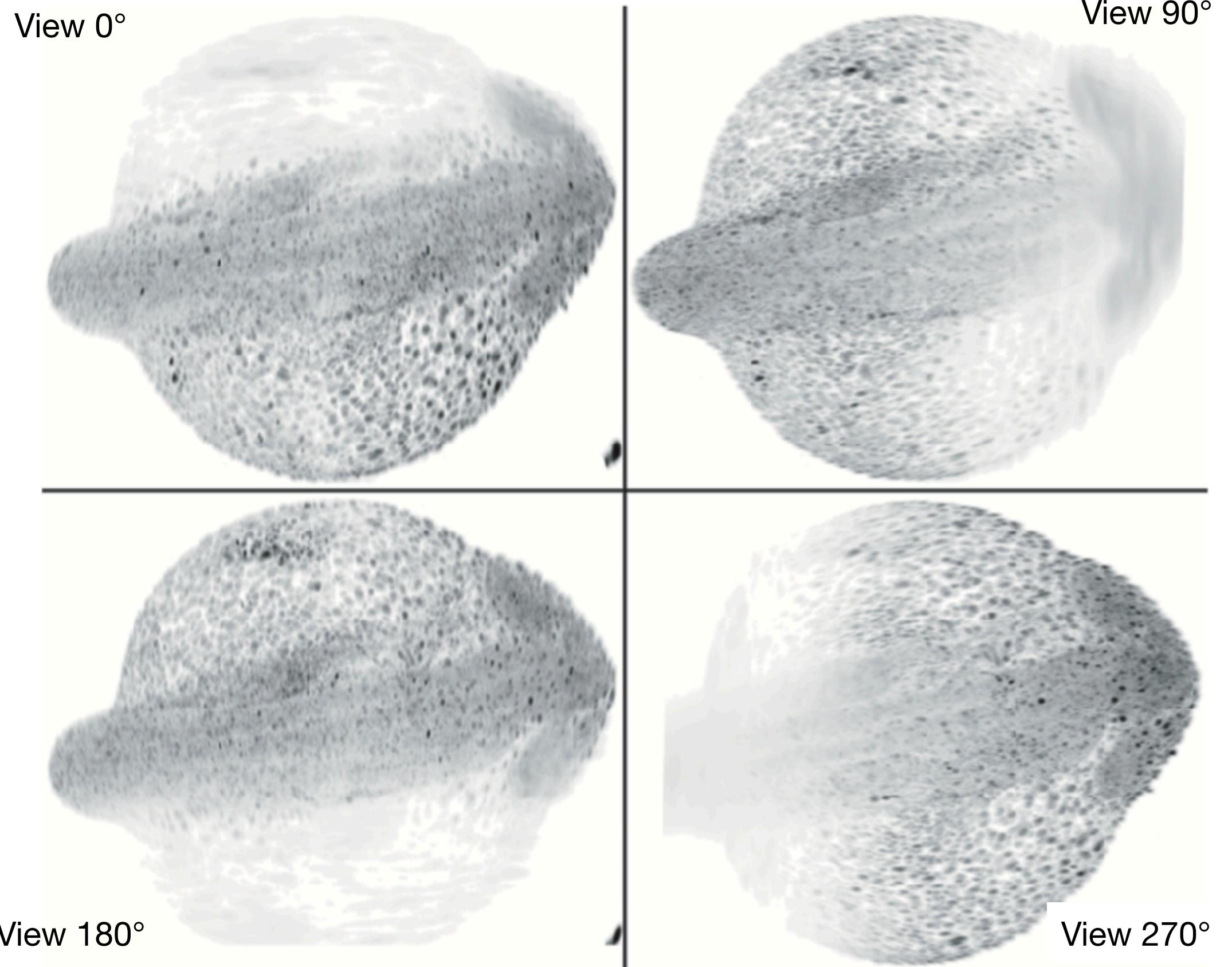


Adapted from (Olarte et al., 2018)

Confocal  
microscopy

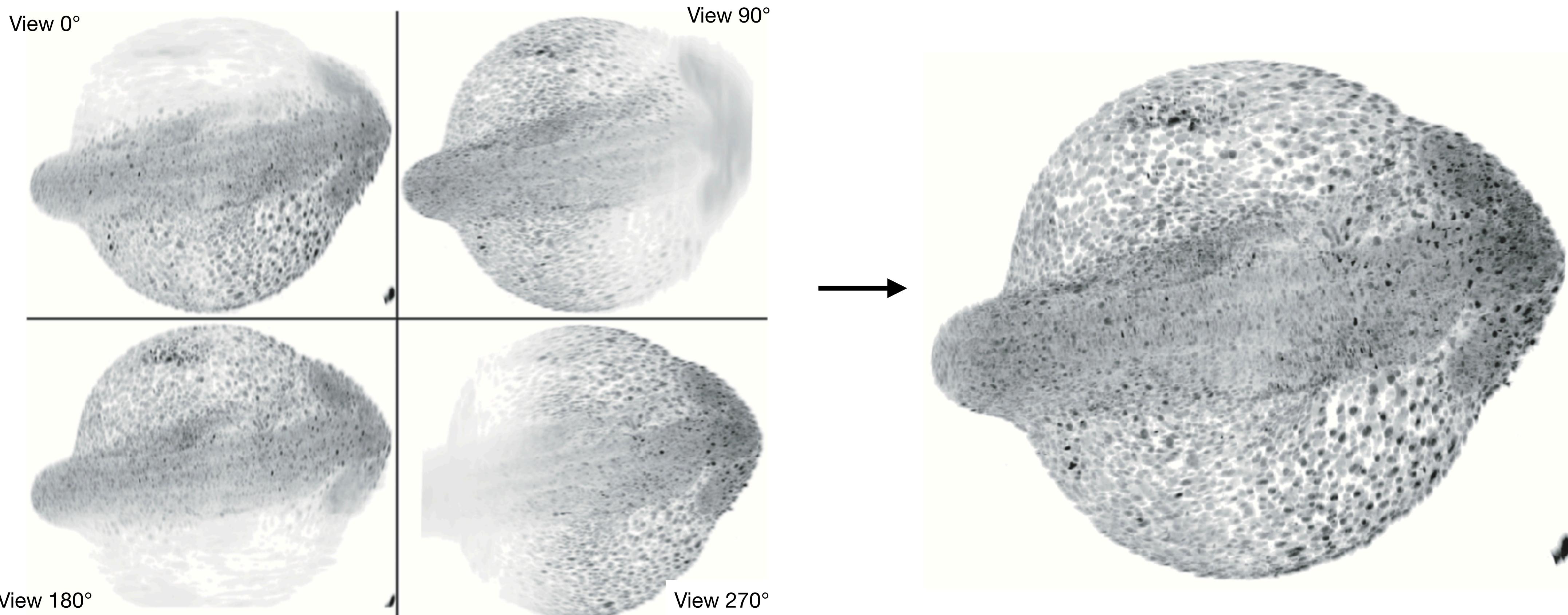


*Upright spinning-disk*



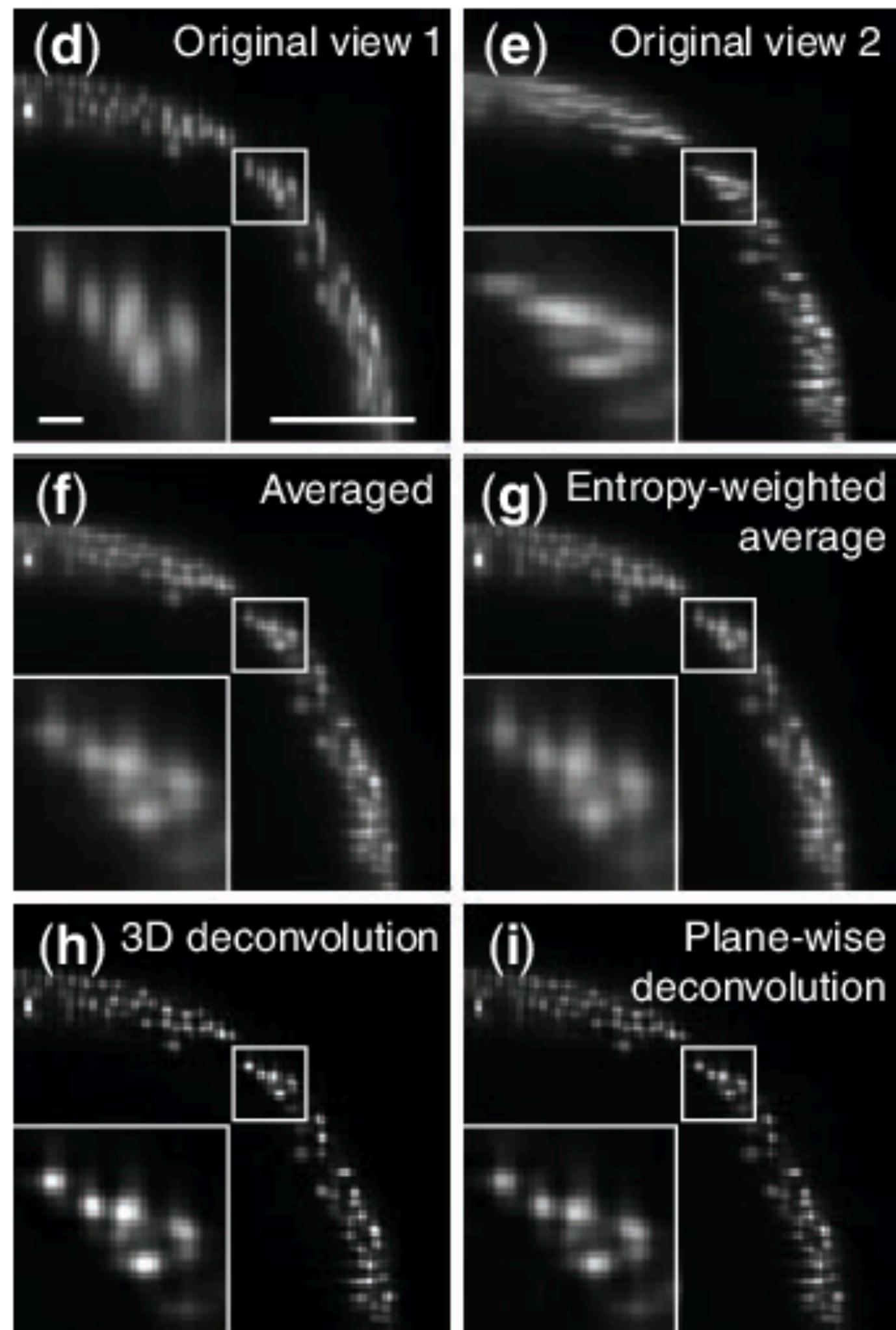
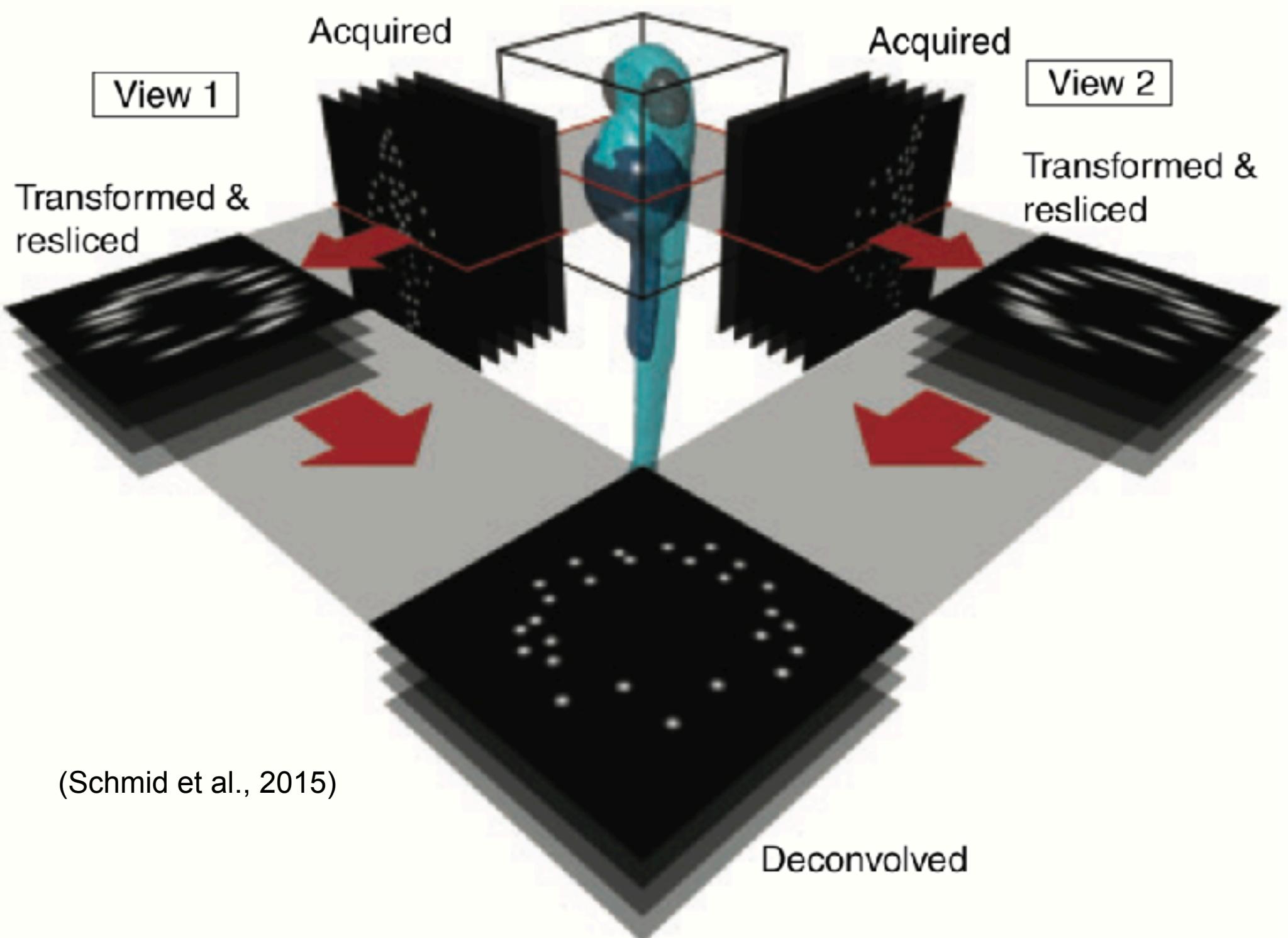
Developing zebrafish embryo (15hpf) imaged using Zeiss Z1

# (Multi-view) microscopy with opaque samples



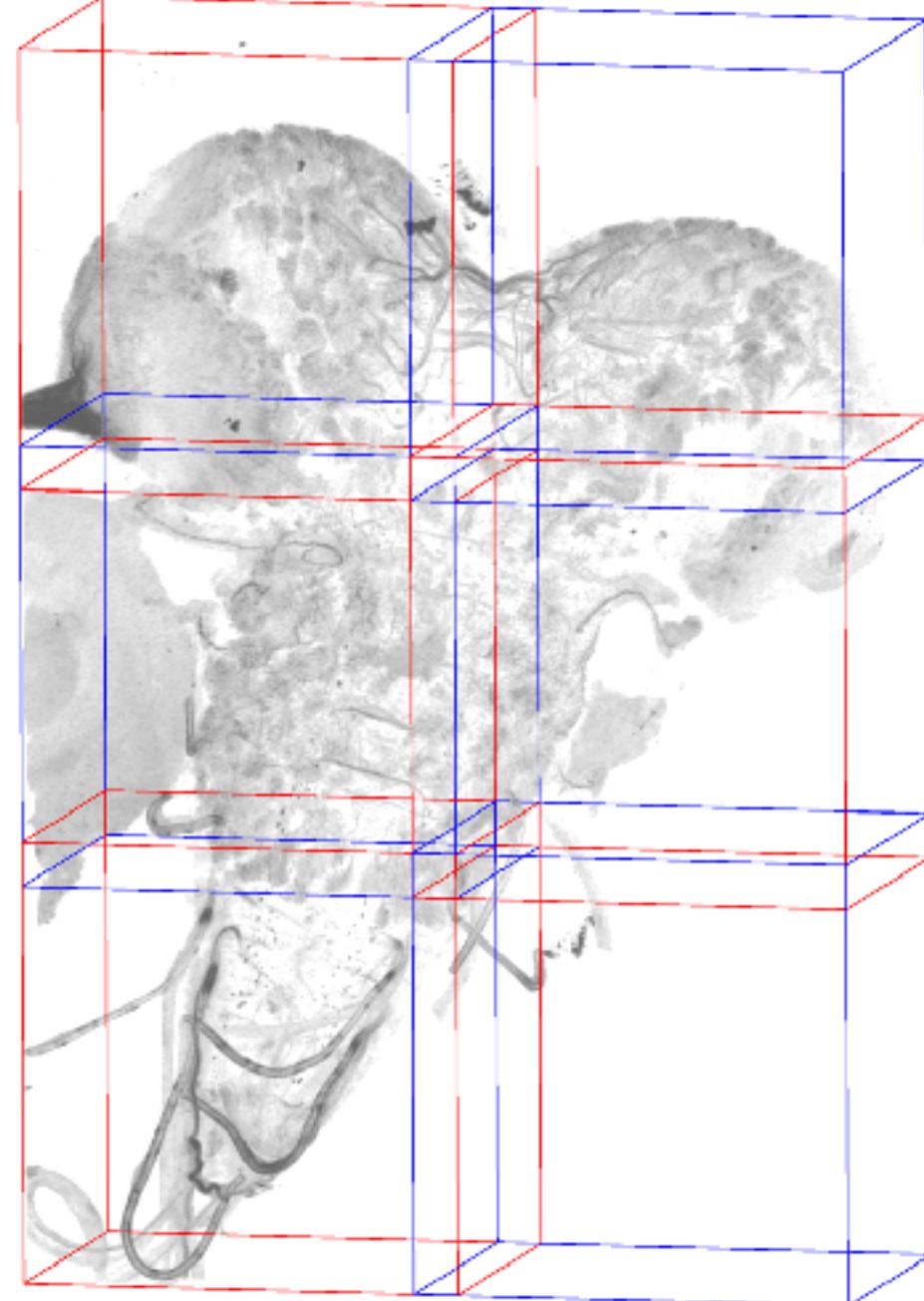
Developing zebrafish embryo (15hpf) imaged using Zeiss Z1

# Multi-view light sheet fusion

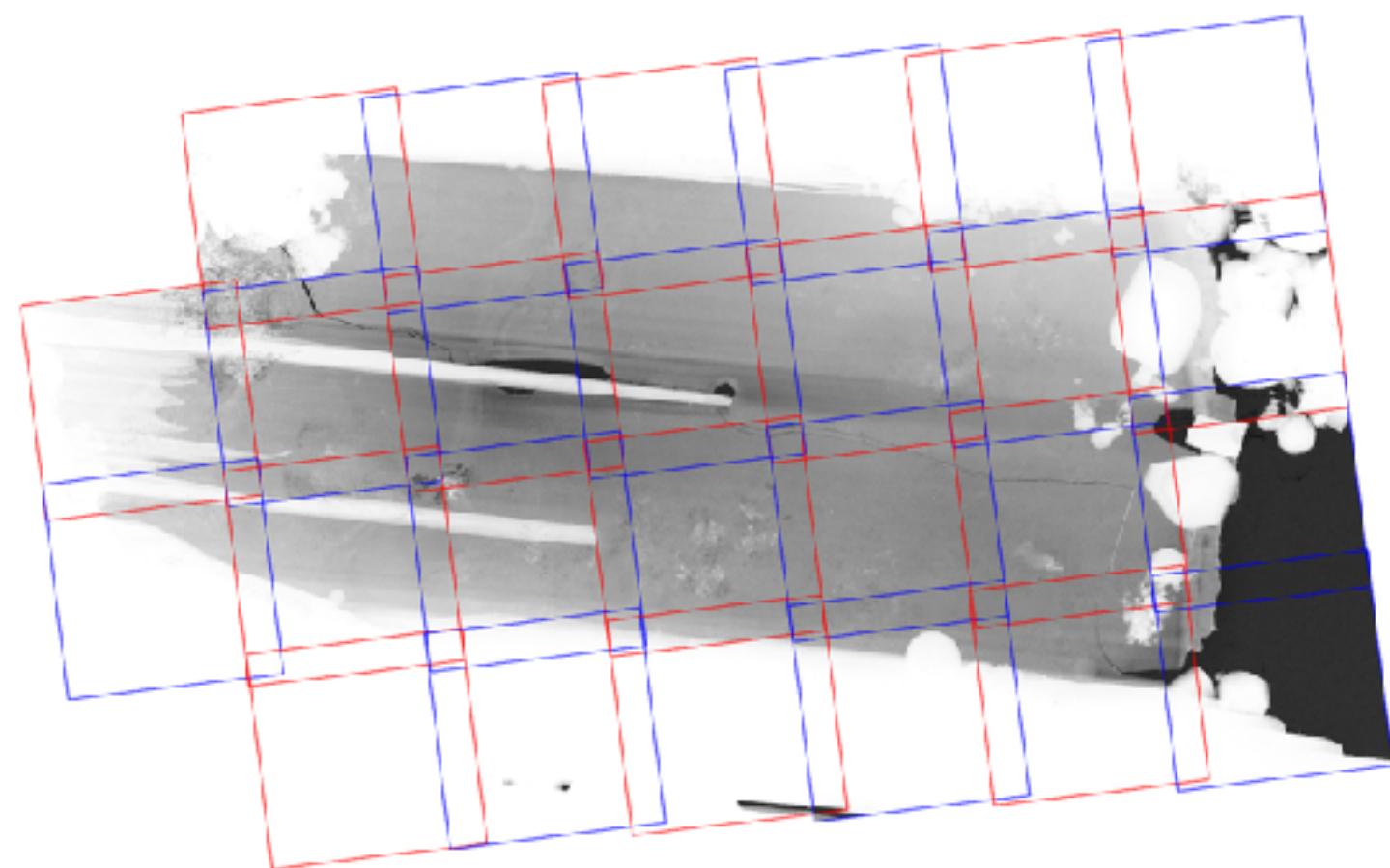


# Reconstructing multi-tile/view image modalities

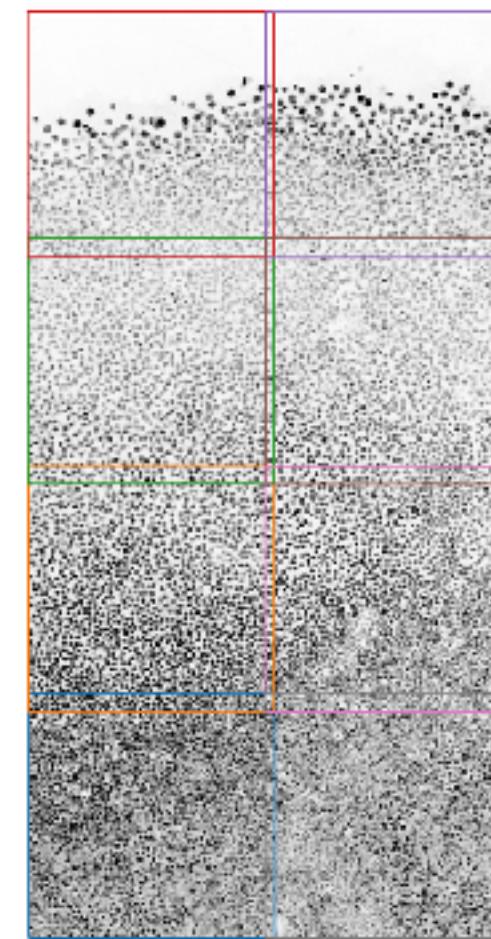
Multi-positioning / map acquisitions



3D spinning disk

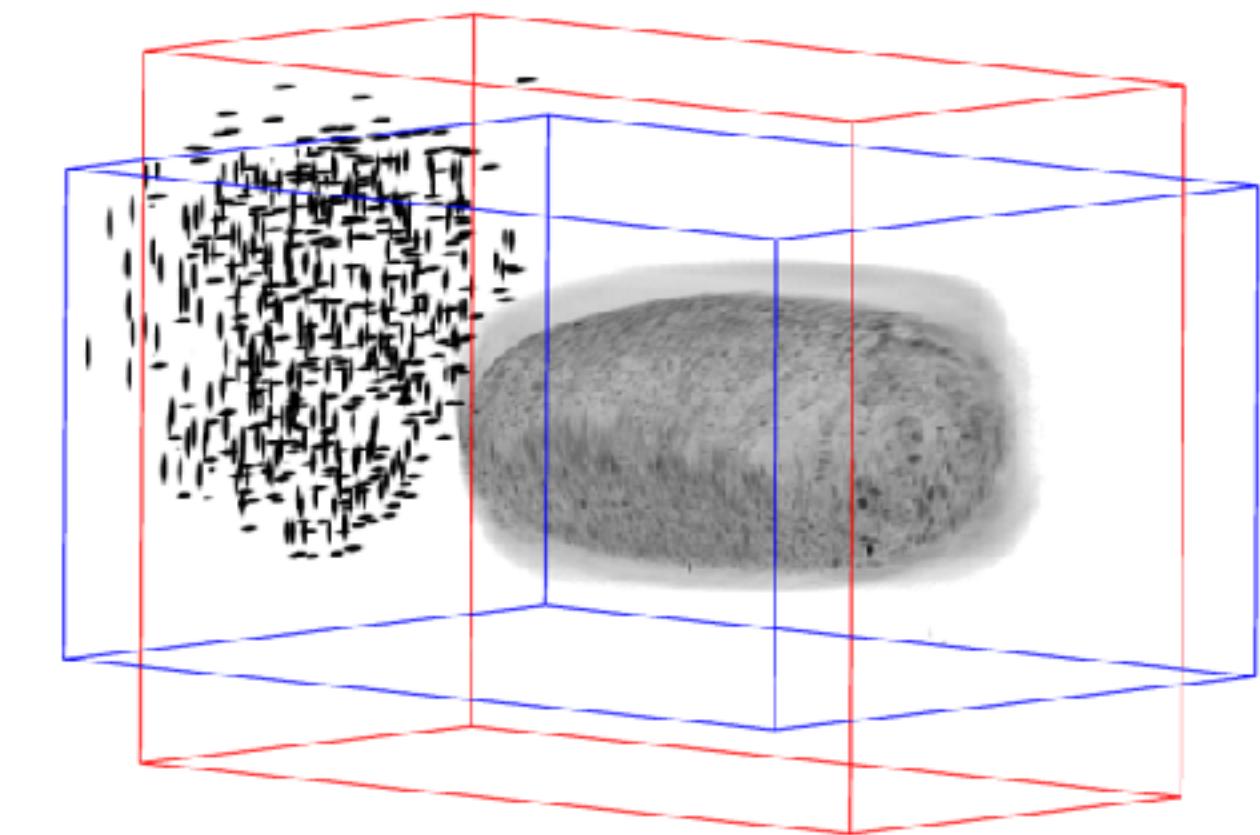


Cryo-EM Map



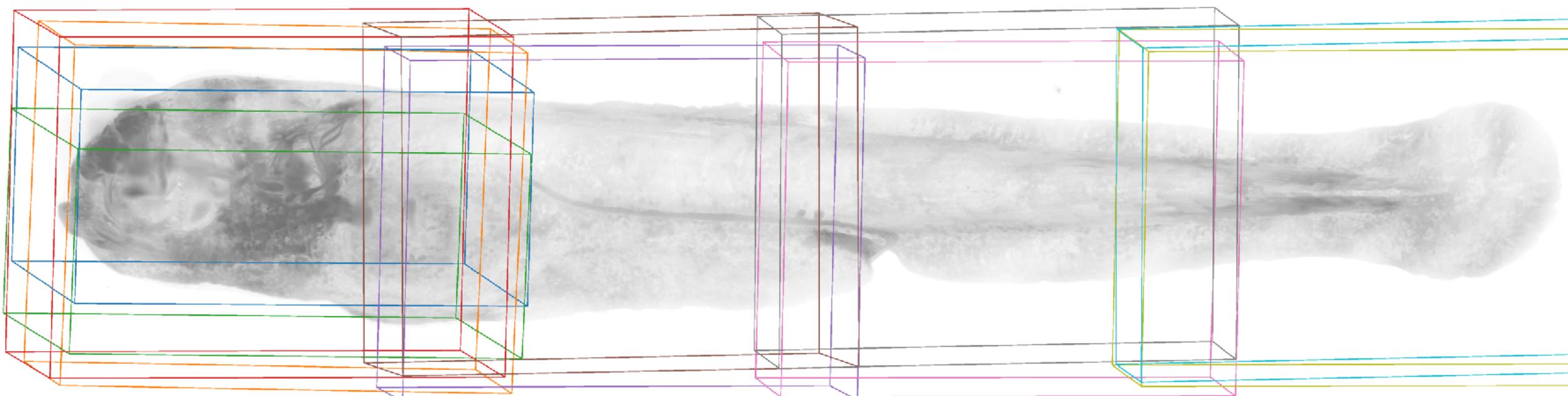
Multi-positioning confocal

Multi-view acquisitions



Multi-view light sheet microscopy

Multi-tile + multi-view



Tiled multi-view light sheet microscopy



# multiview-stitcher

multiview-stitcher / multiview-stitcher

Type  to search

Code Issues 9 Pull requests 3 Discussions Actions Projects Wiki Security Insights Settings

Edit Pins Unwatch 4 Fork 8 Starred 50

multiview-stitcher Public

main 14 Branches 18 Tags Go to file + <> Code

m-albert Made sure to carry along node attributes during shortest path... fd9189a · 15 hours ago 231 Commits

.github/workflows Updated doc building python version 2 weeks ago

docs Updated docs 2 weeks ago

notebooks multi-view example notebook: don't prune registrati... last month

src/multiview\_stitcher Made sure to carry along node attributes during sho... 15 hours ago

.gitignore Propagate chunk sizes from sim to msim and modify... 10 months ago

About

A toolbox for registering / fusing / stitching large multi-view / multi-positioning image datasets in 2-3D.

[multiview-stitcher.github.io/multiview...](https://multiview-stitcher.github.io/multiview...)

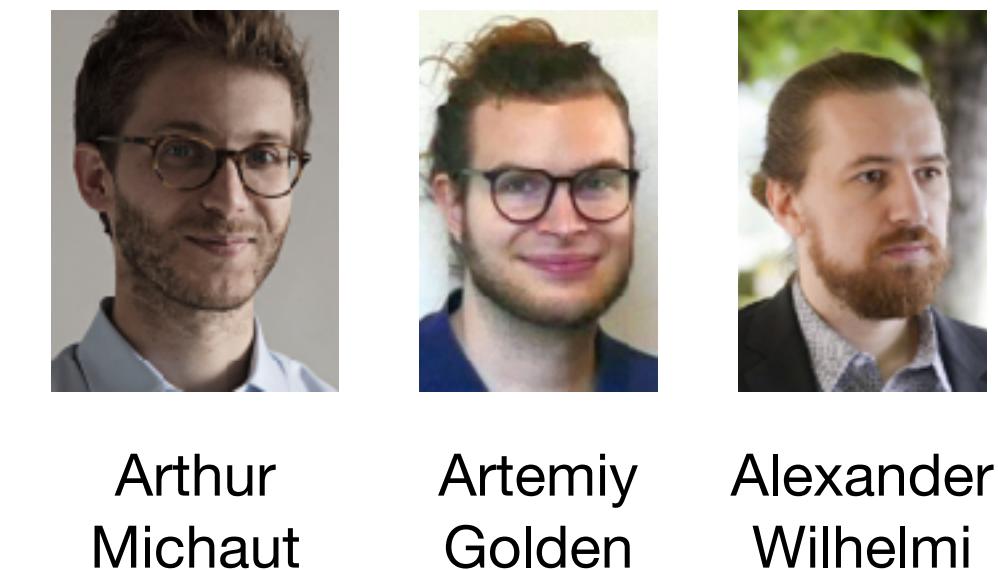
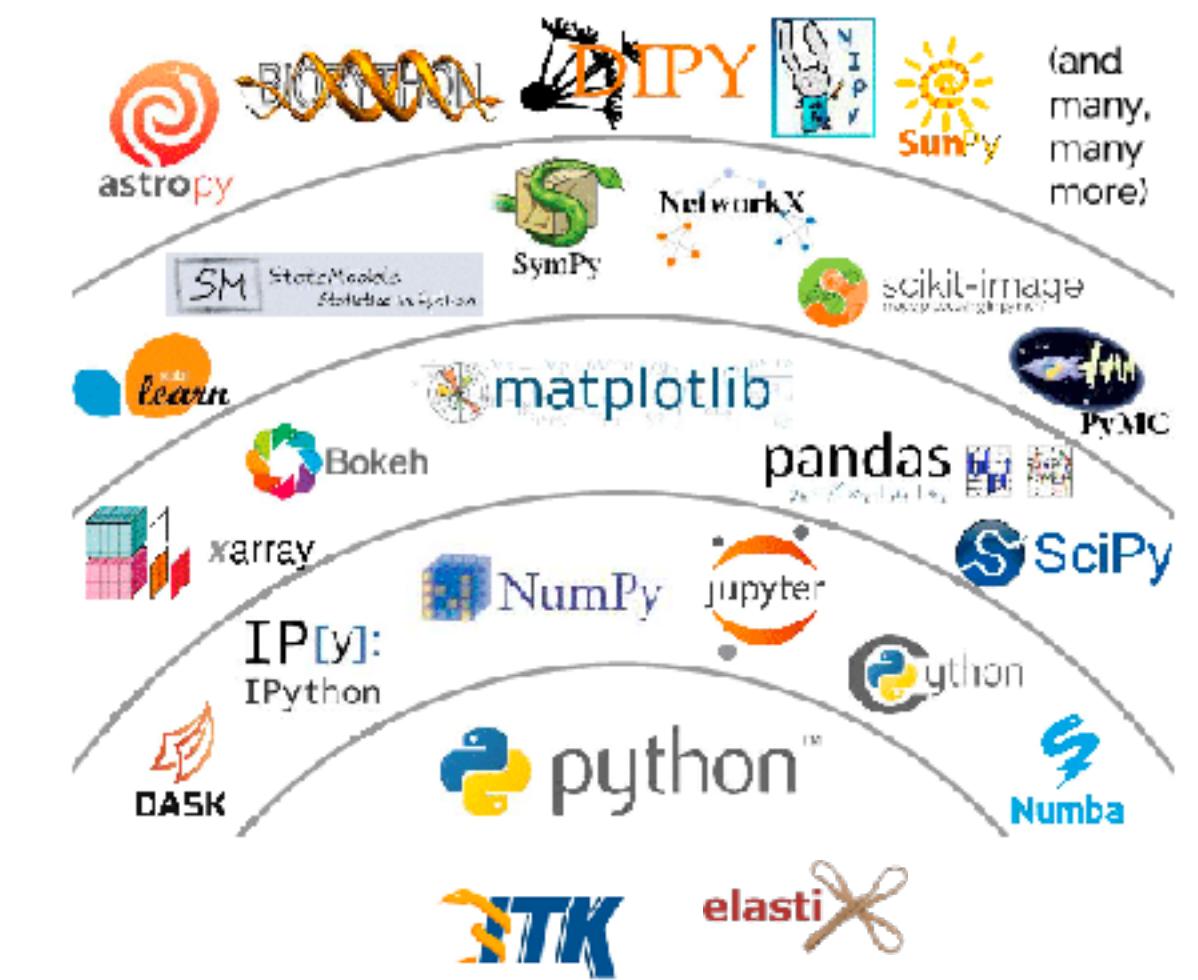
image-processing stitching  
image-registration image-fusion

Readme BSD-3-Clause license

- Registration / fusion / stitching
- Diverse algorithm implementations
- 2D and 3D

- Chunk-aware for large data processing
- Use as a library / programmatically
- napari visualization / GUI

## Scientific python stack



& code contributors



# Other open source stitching software

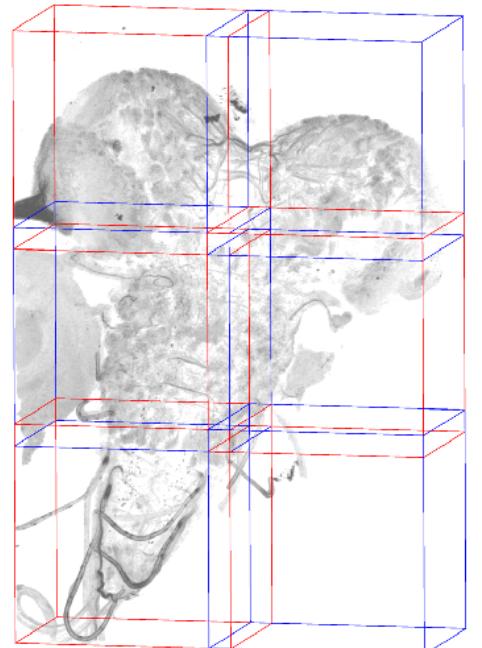
- BigStitcher
  - Mature Fiji plugin
  - 2D and 3D
  - Support for very large data
  - Great multi-view deconvolution
- Ashlar
  - Python-based command line tool
  - 2D stitching and cycle registration
- TeraStitcher
  - Support for large data
  - 2D and 3D

# Other open source stitching software

- BigStitcher
  - Mature Fiji plugin
  - 2D and 3D
  - Support for very large data
  - Great multi-view deconvolution
- Ashlar
  - Python-based command line tool
  - 2D stitching and cycle registration
- TeraStitcher
  - Support for large data
  - 2D and 3D

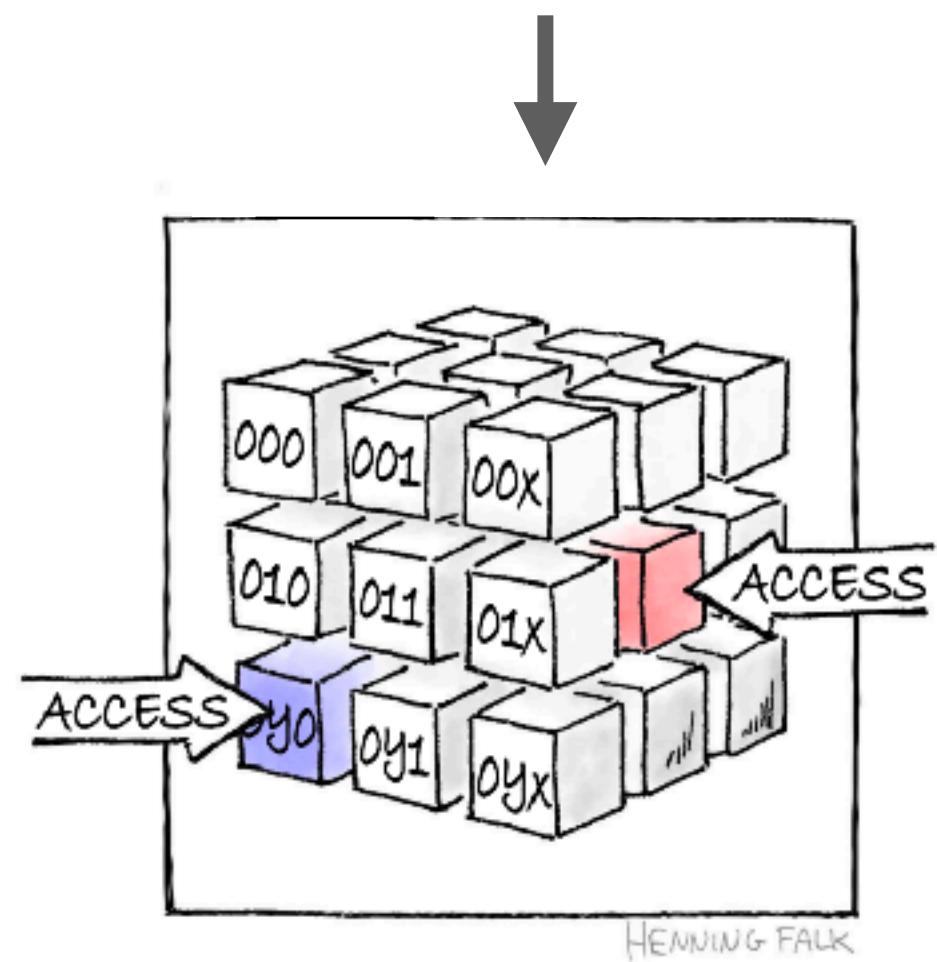
## Unique multiview-stitcher features:

- Stitching in python:
  - 2D and 3D
  - Large data support
- Extensible framework supporting custom functions available in the ecosystem for
  - Data loading
  - Registration
  - Fusion



# multiview-stitcher

Python toolbox for registering and fusing  
large multi-positioning image datasets

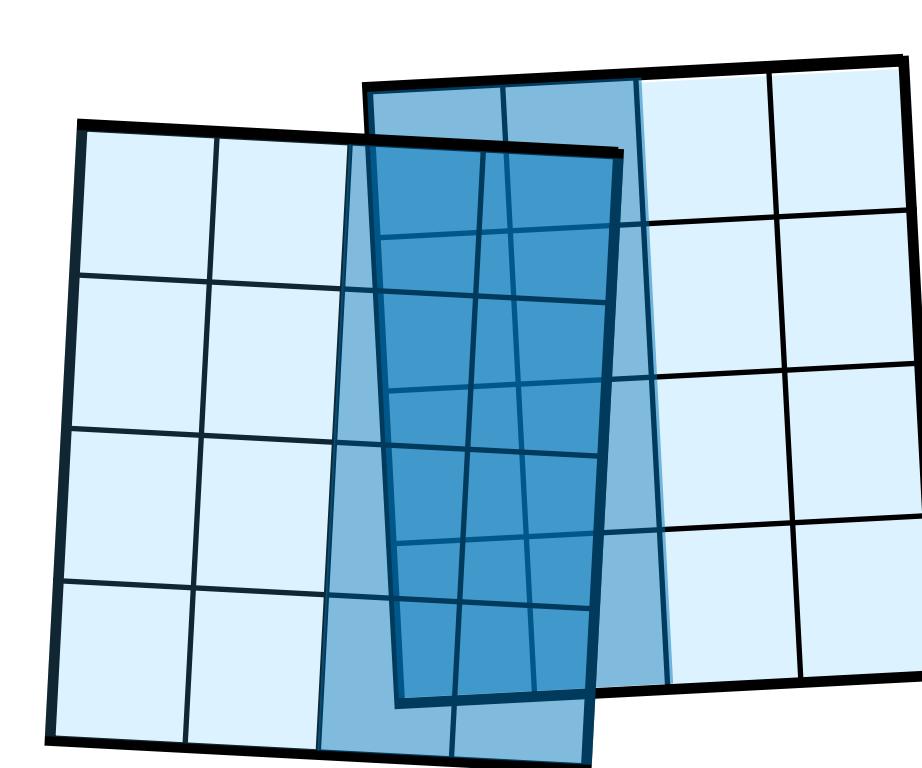


## OME-Zarr file conversion

- Chunked image data

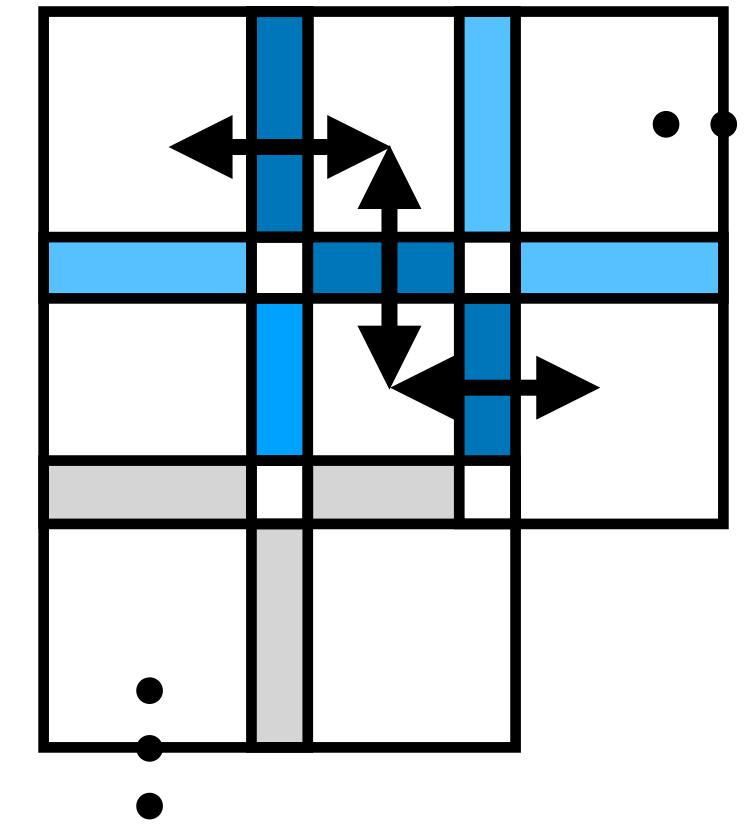
## Pairwise registration

- Load only tile overlap
- Pluggable registration functions



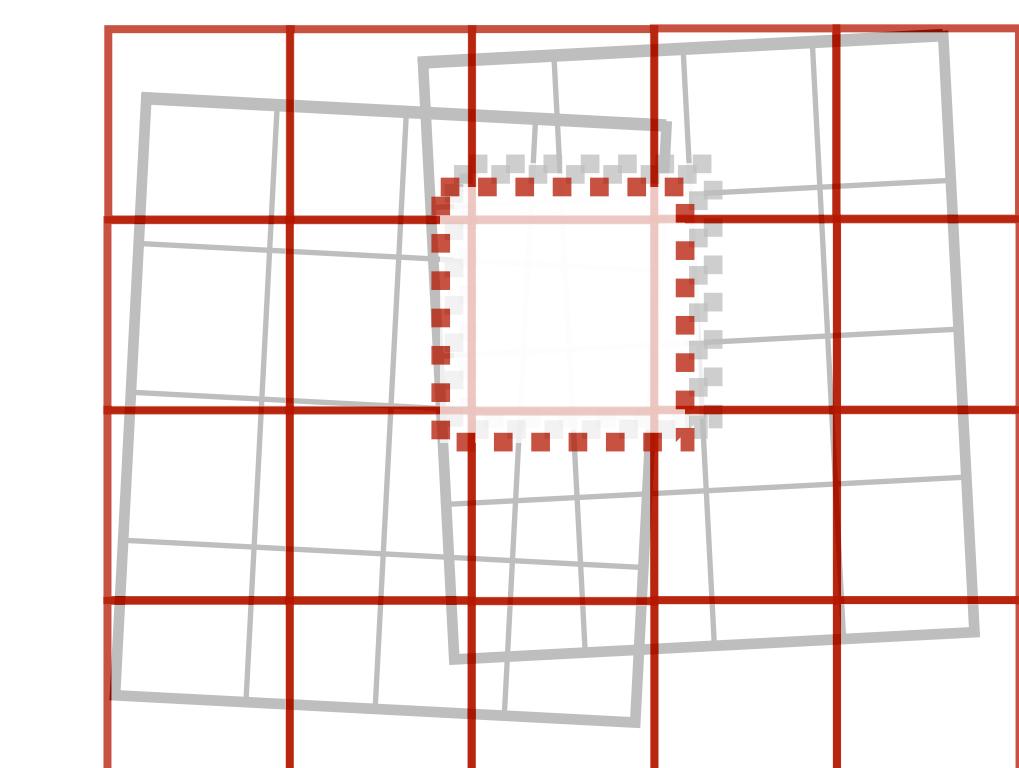
## Group-wise registration

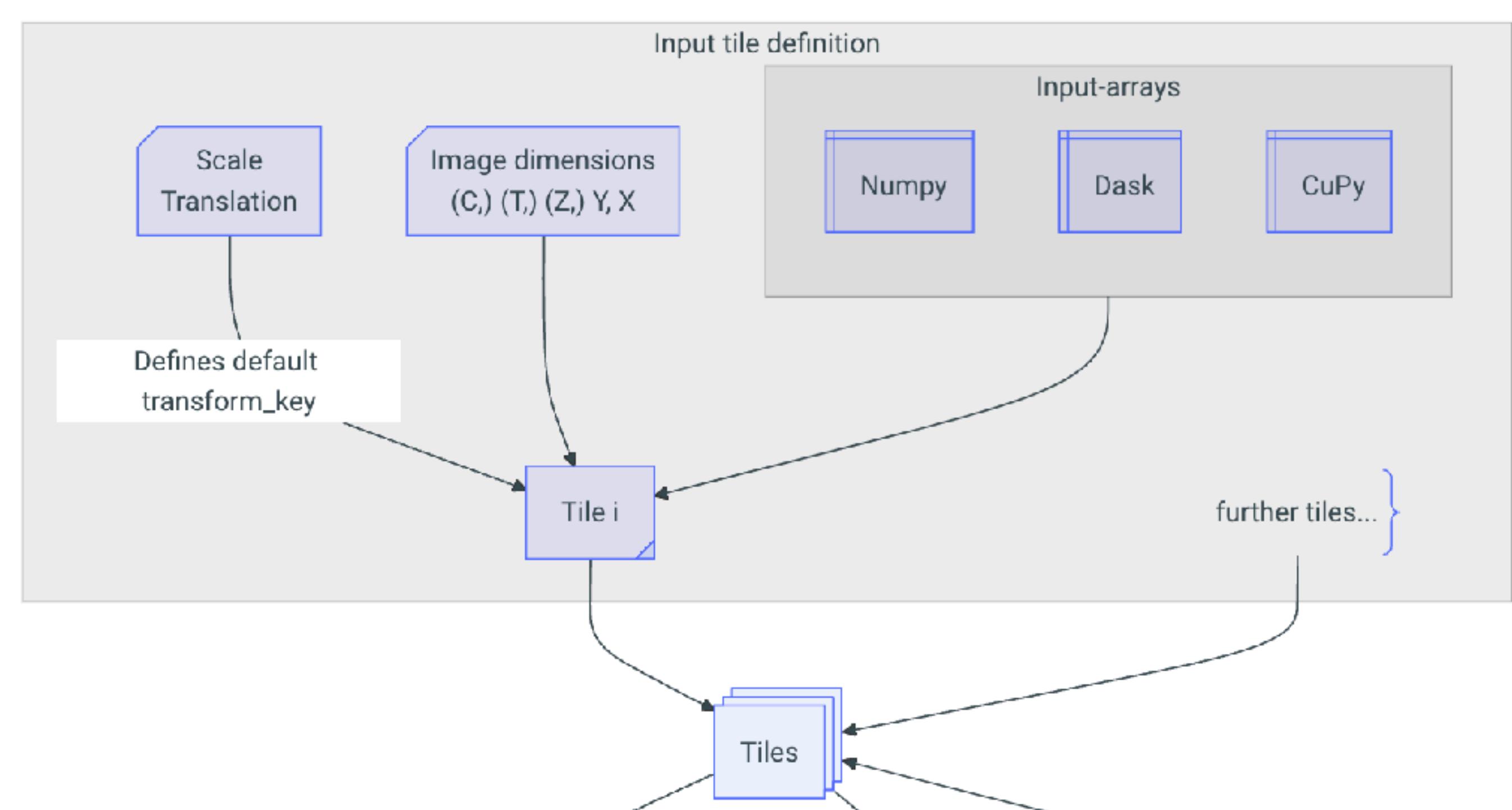
- View adjacency graph
- Global optimization



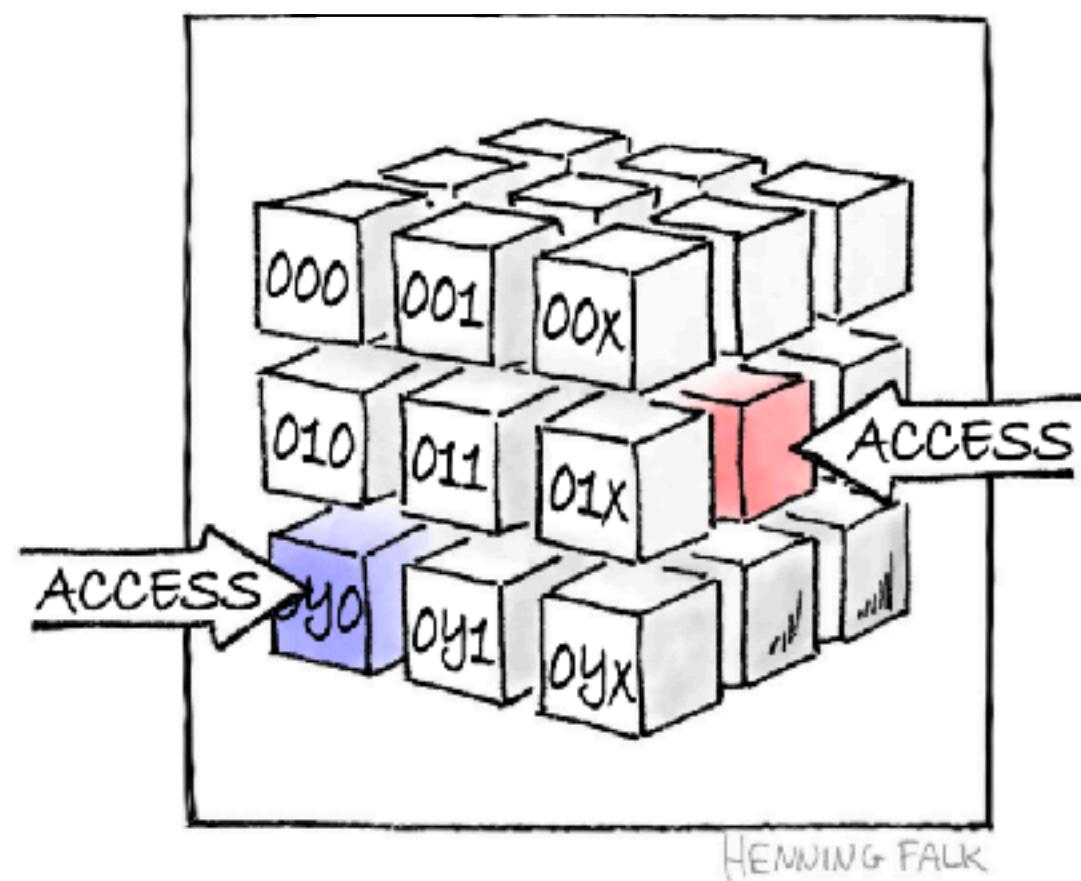
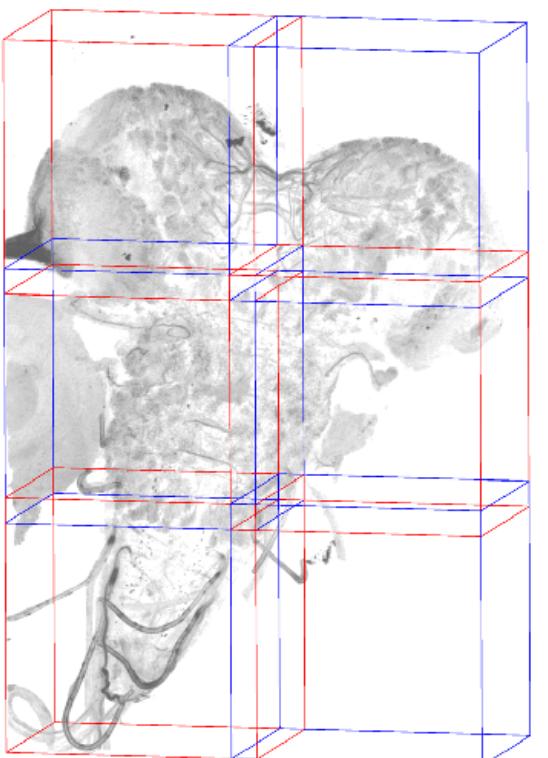
## Image Fusion

- Blending between tiles
- Multi-view deconvolution
- Pluggable fusion functions





# Stitching input



In case of OME-Zarr:

```
sim = ngff_utils.get_msim_from_ngff_zarr('tile1.zarr')
```

Code example

Input tile configuration

```
import numpy as np
import dask.array as da
from multiview_stitcher import msi_utils
from multiview_stitcher import si_utils

# input data can be any numpy-like array:
# numpy, dask, cupy, etc.
tile_arrays = [da.from_zarr('tile1.zarr'),
               da.from_zarr('tile2.zarr'),
               da.from_zarr('tile3.zarr')]

# provide metadata for the tile arrays
dims = ["c", "z", "y", "x"]
channels = ["DAPI", "GFP"]
pixel_spacing = \
    {"z": 2, "y": 0.5, "x": 0.5}
tile_translations = [
    {"z": 2.5, "y": -10, "x": 30},
    {"z": 2.5, "y": 30, "x": 10},
    {"z": 2.5, "y": 30, "x": 50},
]

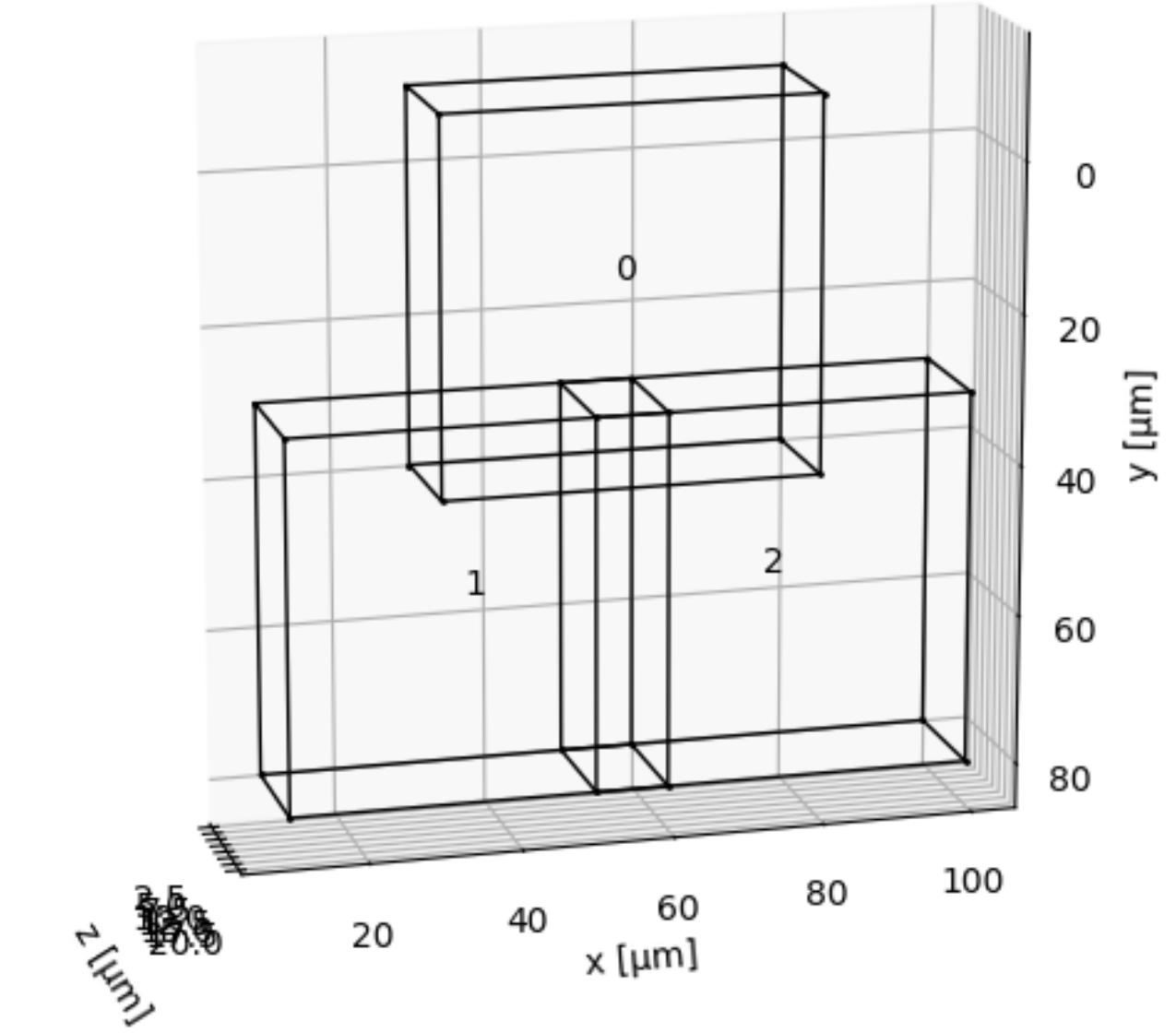
# build input for stitching
msims = []
for view_index in range(3):
    sim = si_utils.get_sim_from_array(
        tile_arrays[view_index],
        dims=dims,
        scale=pixel_spacing,
        translation=tile_translations[view_index],
        transform_key="stage_metadata",
        c_coords=channels,
    )
    msims.append(msi_utils.get_msim_from_sim(
        sim, scale_factors=[]))
```

Visualization of tile positions

```
from multiview_stitcher import vis_utils

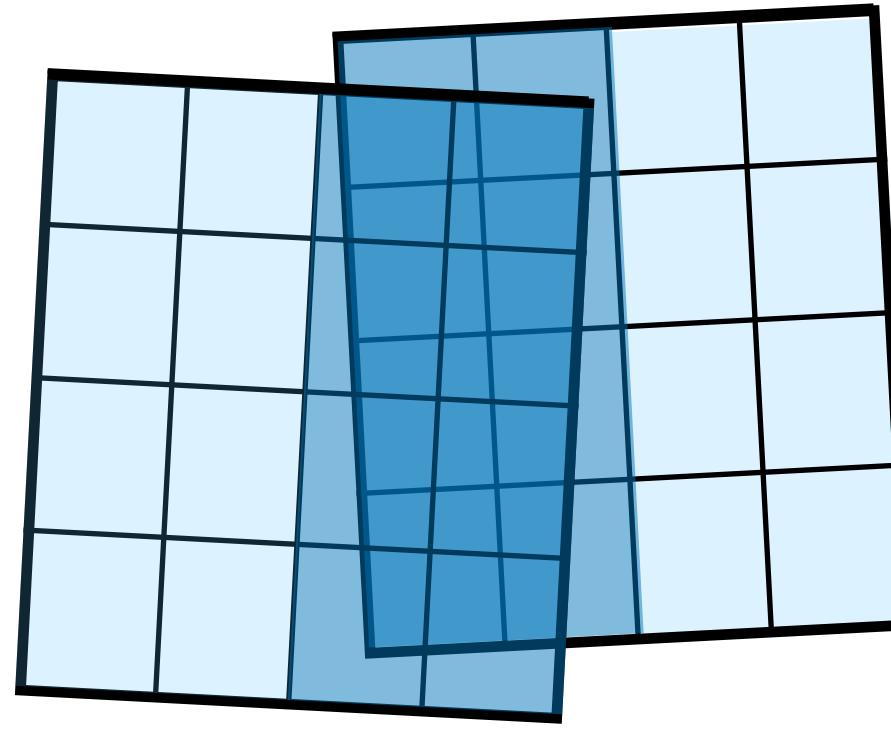
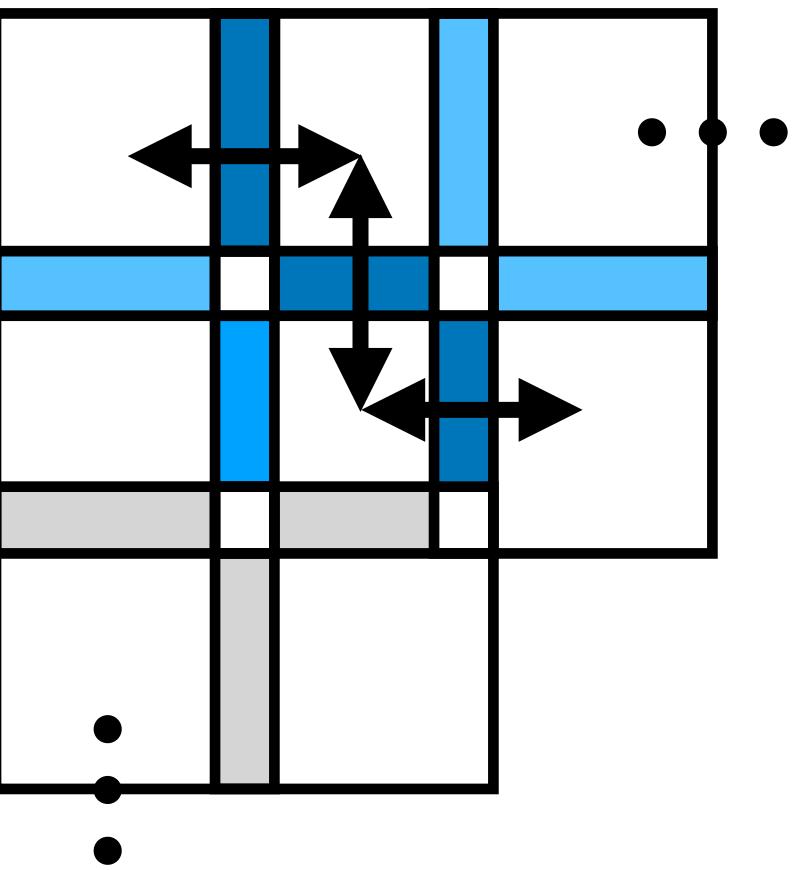
vis_utils.plot_positions(
    msims,
    transform_key='stage_metadata'
)
```

Input tile configuration



# Image registration

## Overview



 NetworkX  
Network Analysis in Python

**Geometry3D**

 dask

### Group-wise registration

- View adjacency graph
- Registration pair determination
- Global optimization
- Parallel execution using dask

### Pairwise registration

- Load only tile overlap
- Downsample (multiscale)
- Pluggable / custom functions
- Supports affine transforms

 scikit-image  
Image processing in python

 ANTsPy

 elastiX

# Image registration

## Code example

Registration entry point:

```
from multiview_stitcher import registration

params = registration.register(
    msims,
    reg_channel="DAPI", # channel to use for registration
    transform_key="stage_metadata", # starting transform key
    new_transform_key="registered", # key for registration results
)
```

# Image registration

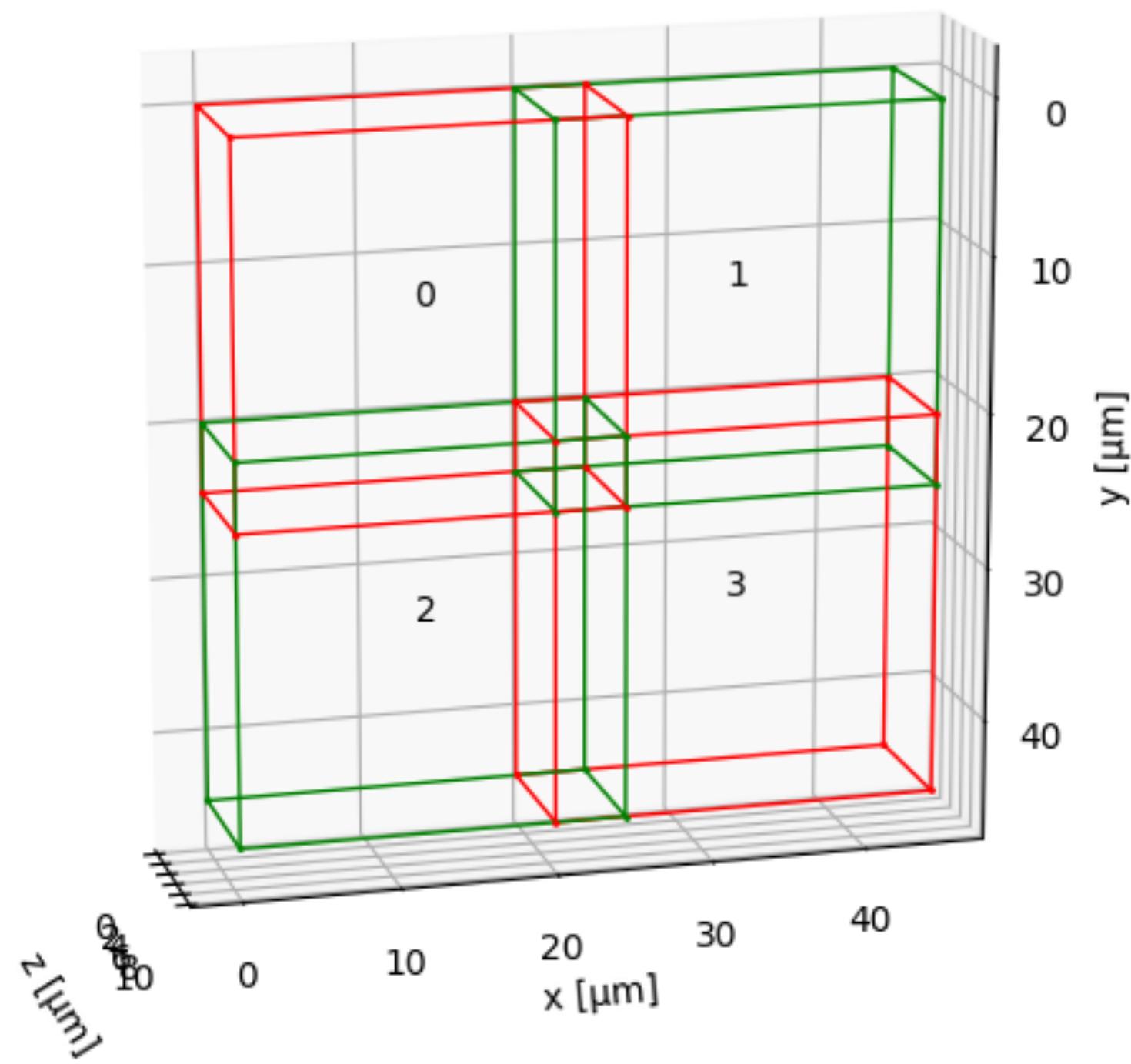
## Code example

Registration entry point:

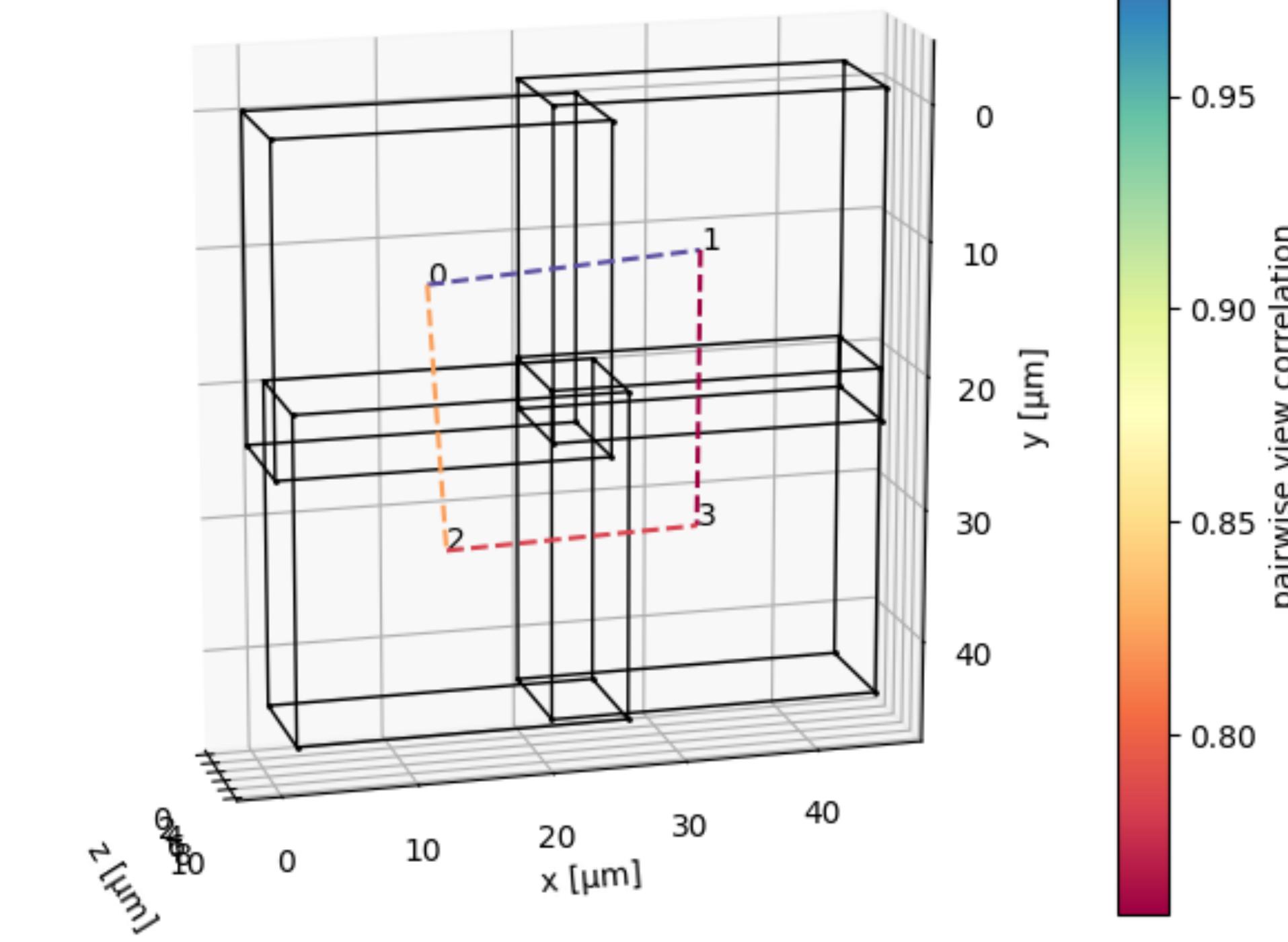
```
from multiview_stitcher import registration

params = registration.register(
    msims,
    reg_channel="DAPI", # channel to use for registration
    transform_key="stage_metadata", # starting transform key
    new_transform_key="registered", # key for registration results
)
```

Tile arrangement before registration

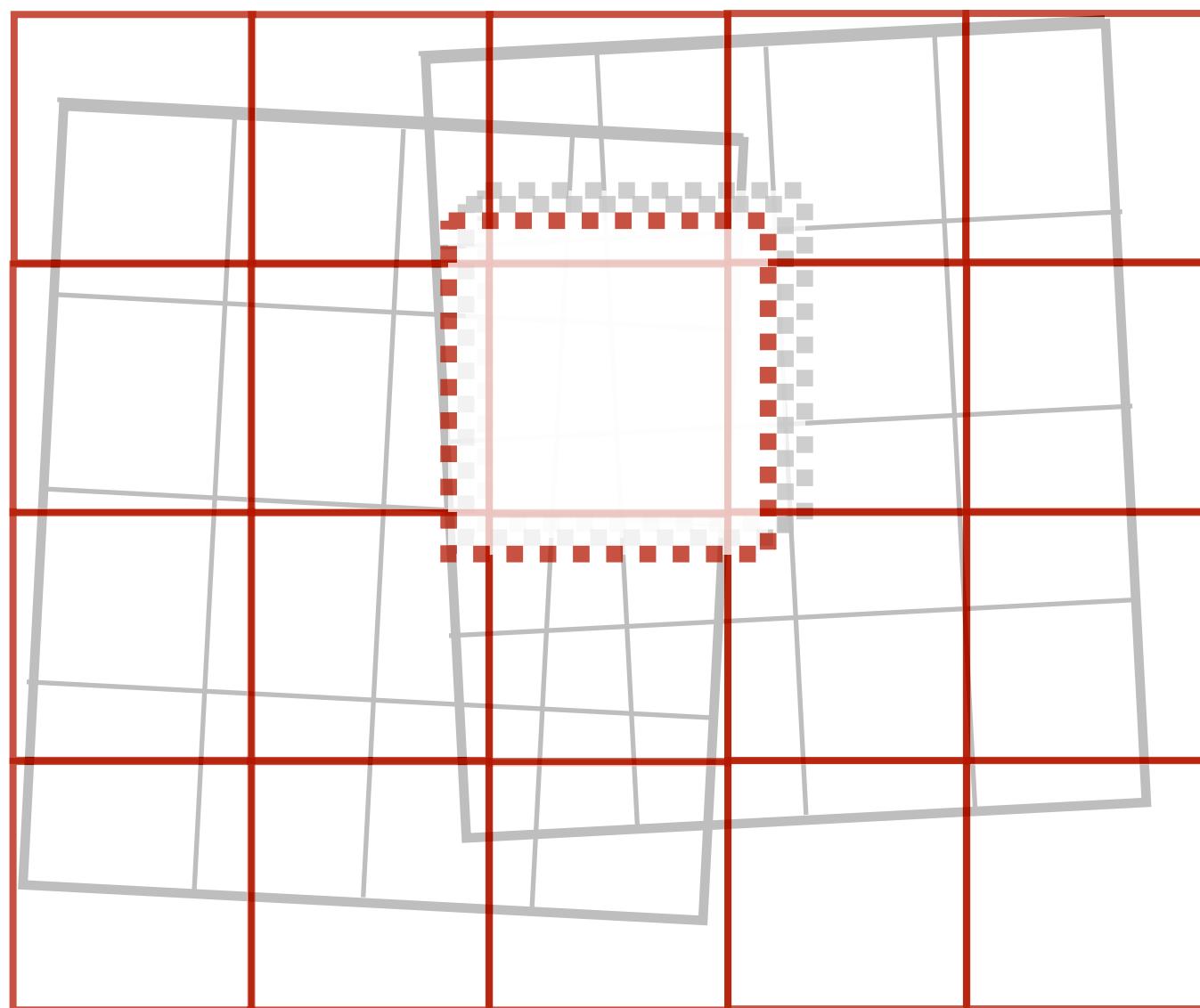


After registration / summary



# Image fusion

## Overview



- Input views / tiles (chunked)
- Fused output chunks
- Fusion input chunks



dask-image



CuPy



- Chunk-by-chunk for large data
- Transformation into output coordinate system
- Support for selected fusion modalities:
  - Blending fusion
  - Contrast-weighted fusion
  - Multi-view deconvolution (soon)
- Pluggable / custom functions

### Code example:

```
from multiview_stitcher import fusion

fused_sim = fusion.fuse(
    [msi_utils.get_sim_from_msim(msim) for msim in msims],
    transform_key="registered",
    output_spacing={"z": 1, "y": 0.5, "x": 0.5},
    output_chunksize={"z": 1, "y": 100, "x": 100},
    # fusion_func=fusion.weighted_average_fusion,
)

# get fused array as a dask array
fused_sim.data

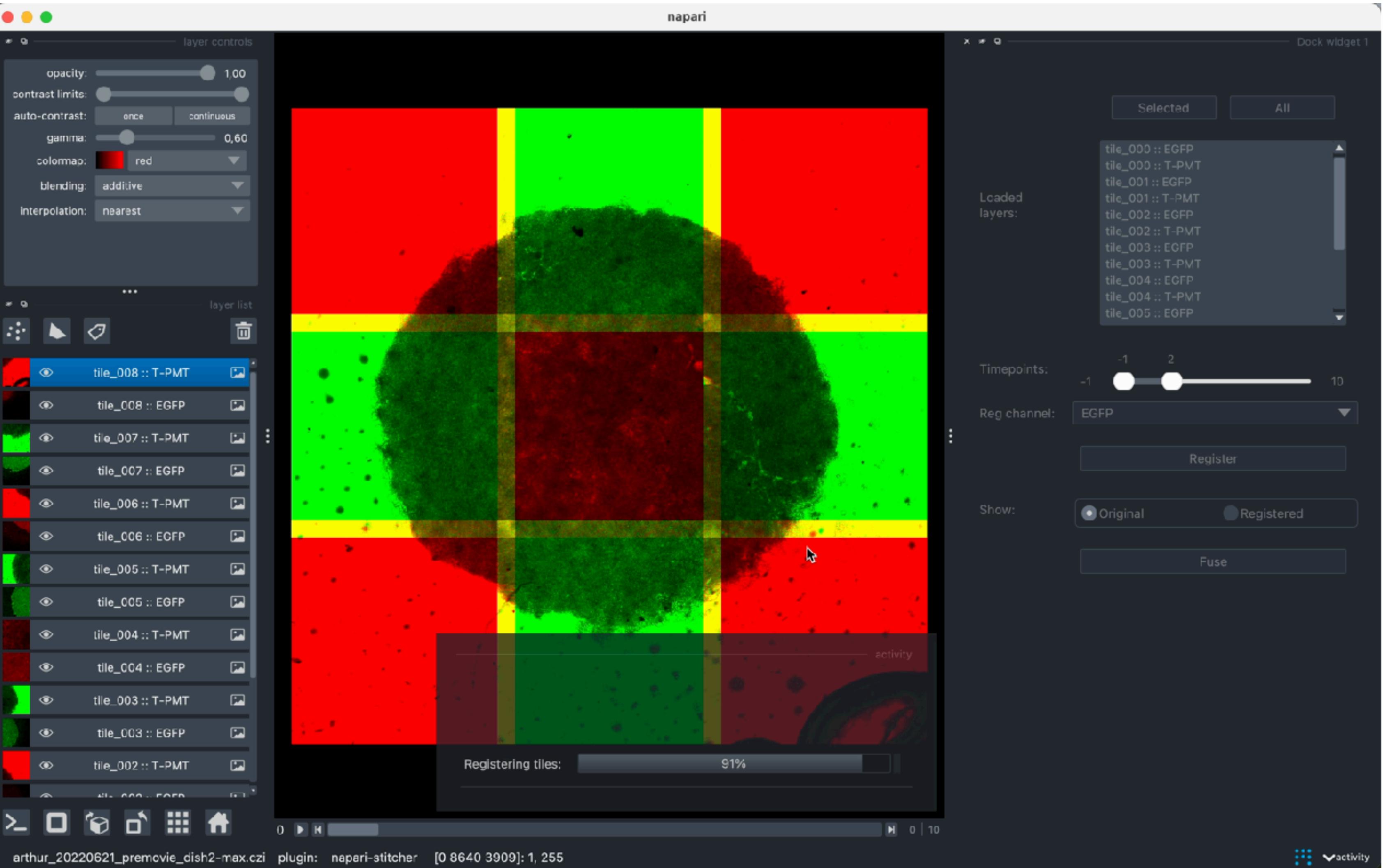
# get fused array as a numpy array
fused_sim.data.compute()
```

# Visualization and GUI

## napari-stitcher

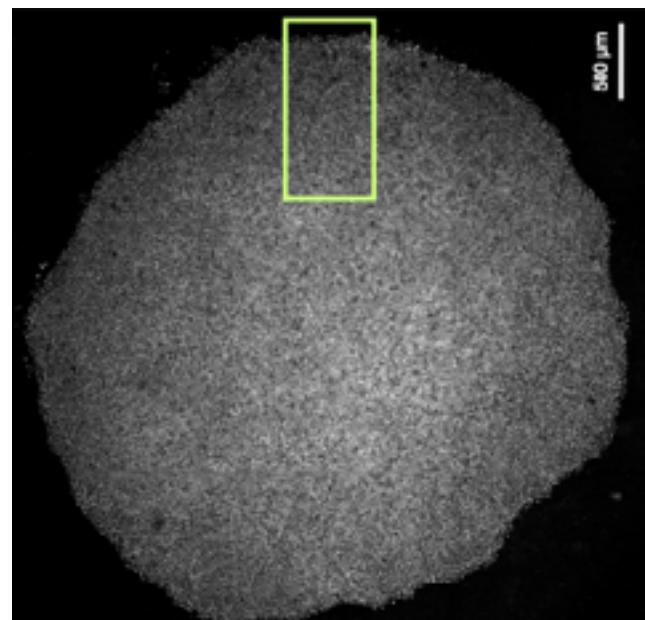
- Basic GUI
- Directly stitches napari layers

[github.com/multiview-stitcher/  
napari-stitcher](https://github.com/multiview-stitcher/napari-stitcher)

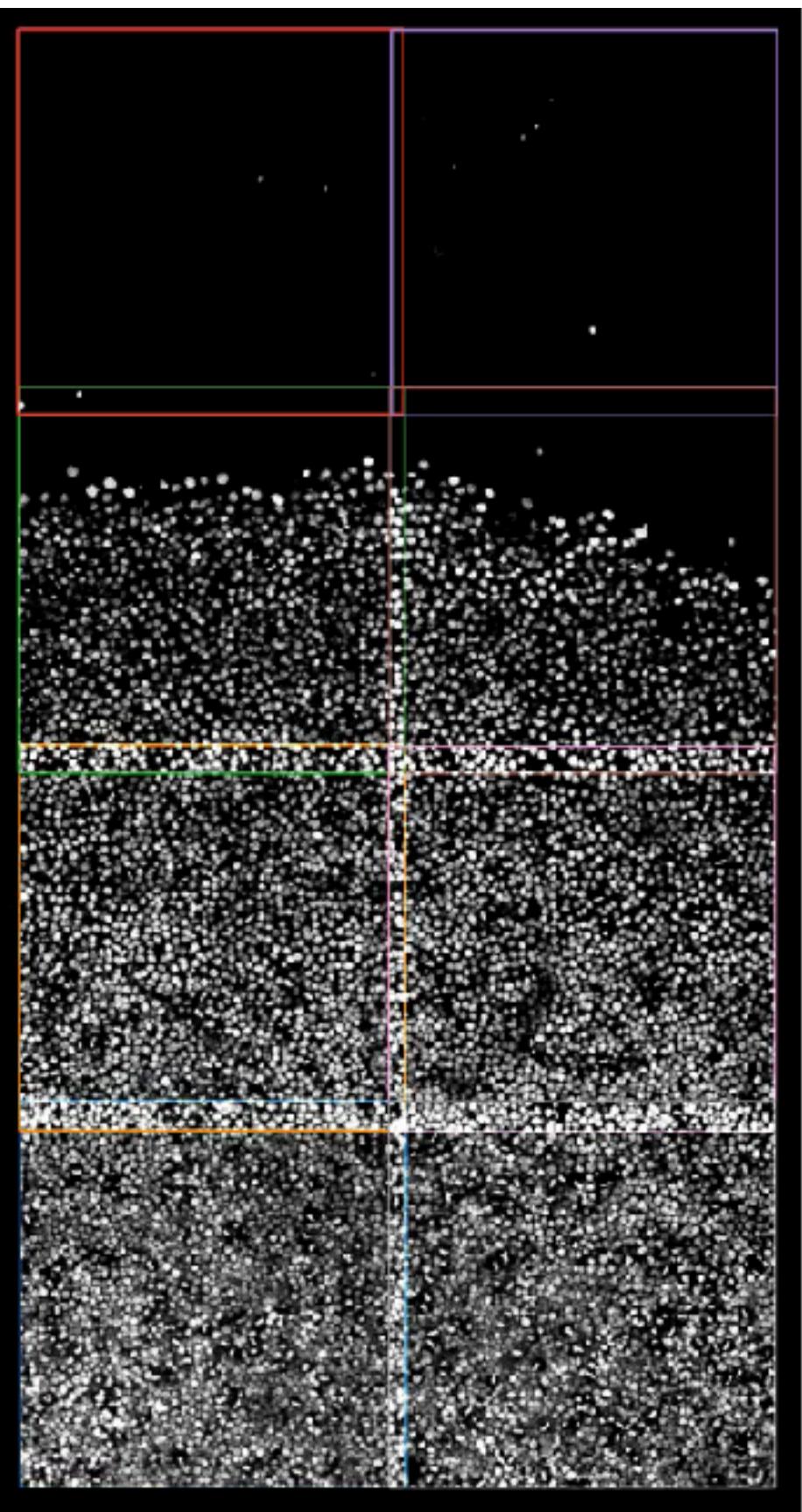


# Applications of multiview-stitcher

Multi-positioning developmental time lapses

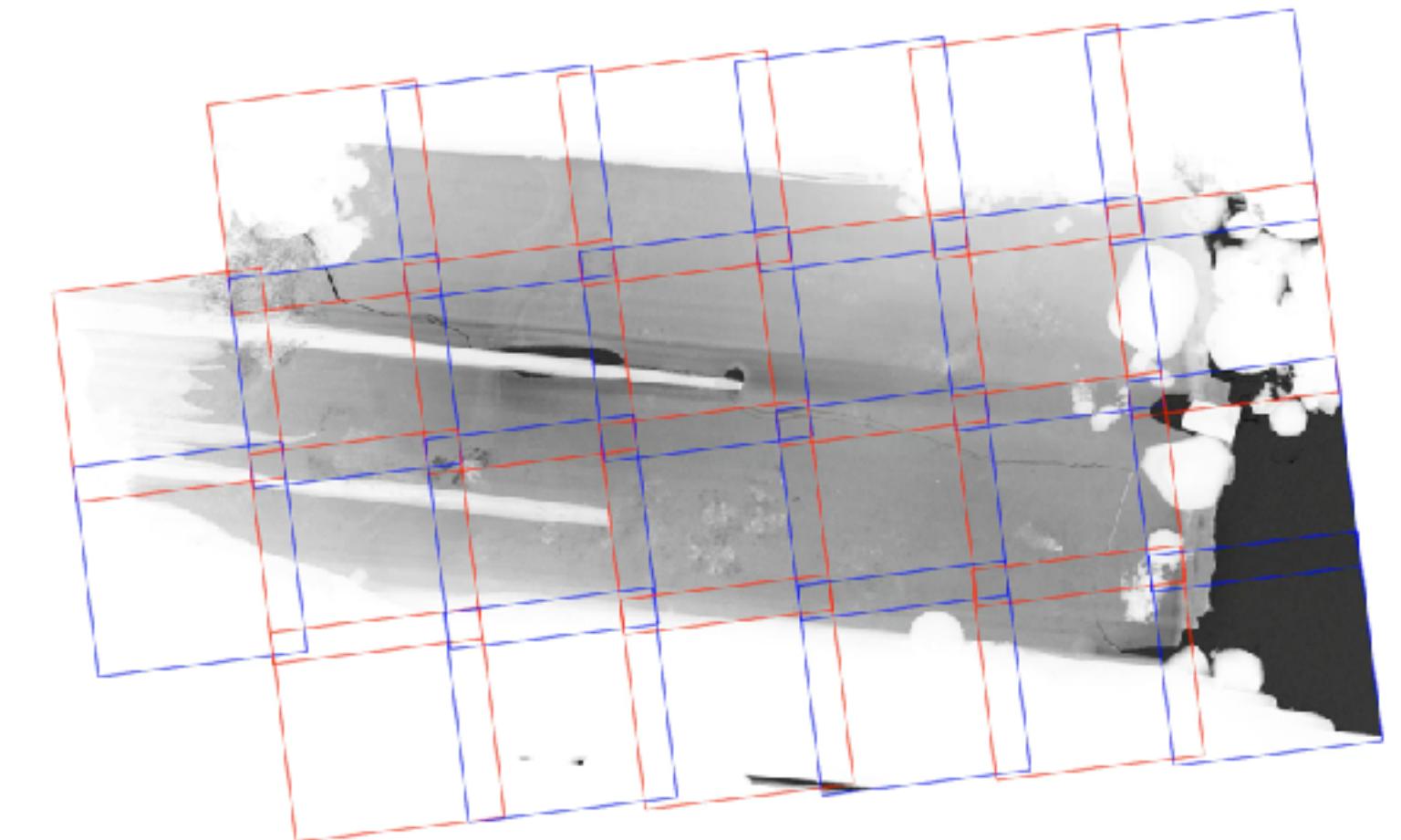
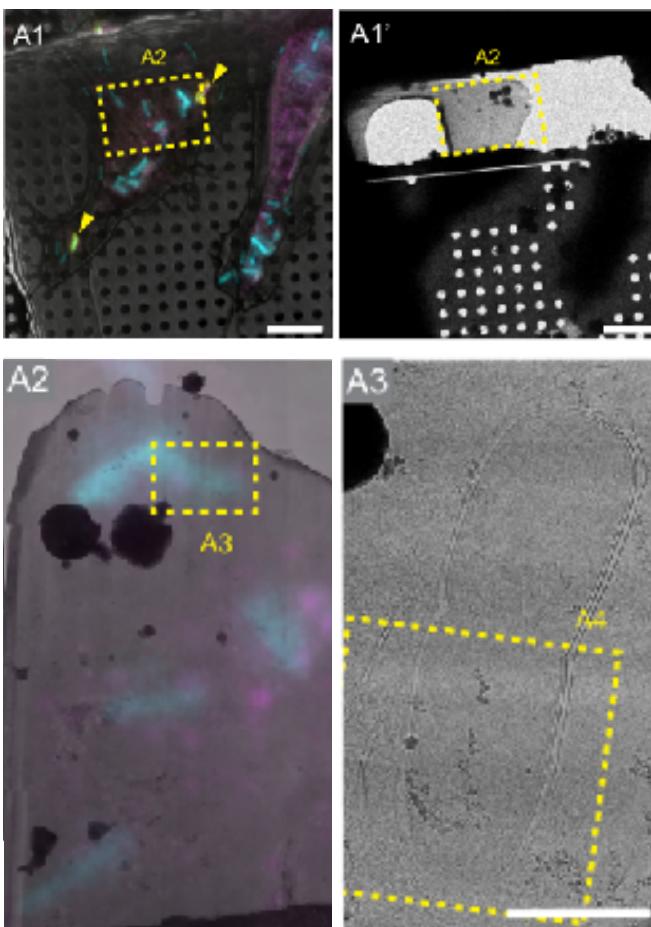


Quail embryo



Arthur Michaut  
Institut Pasteur

EM Montage stitching / CLEM registration



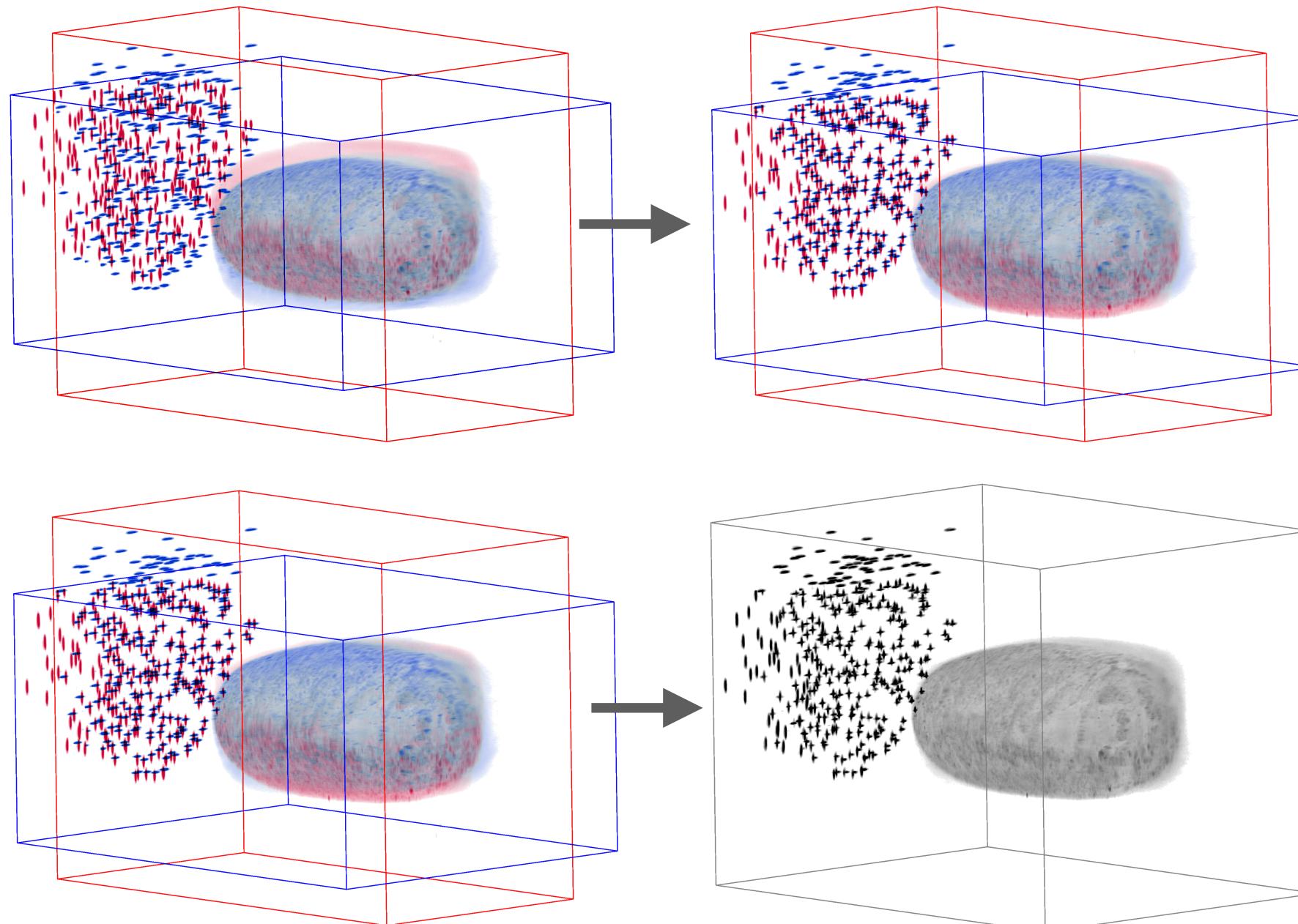
Léa Swistak  
Institut Pasteur



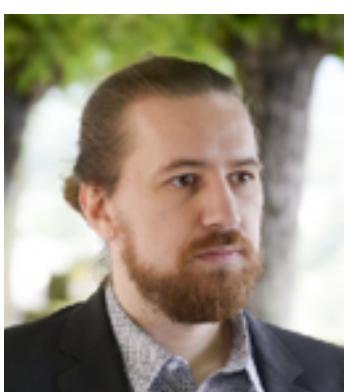
Jost Enninga  
Institut Pasteur

# Applications of multiview-stitcher

## Multi-view light sheet reconstruction

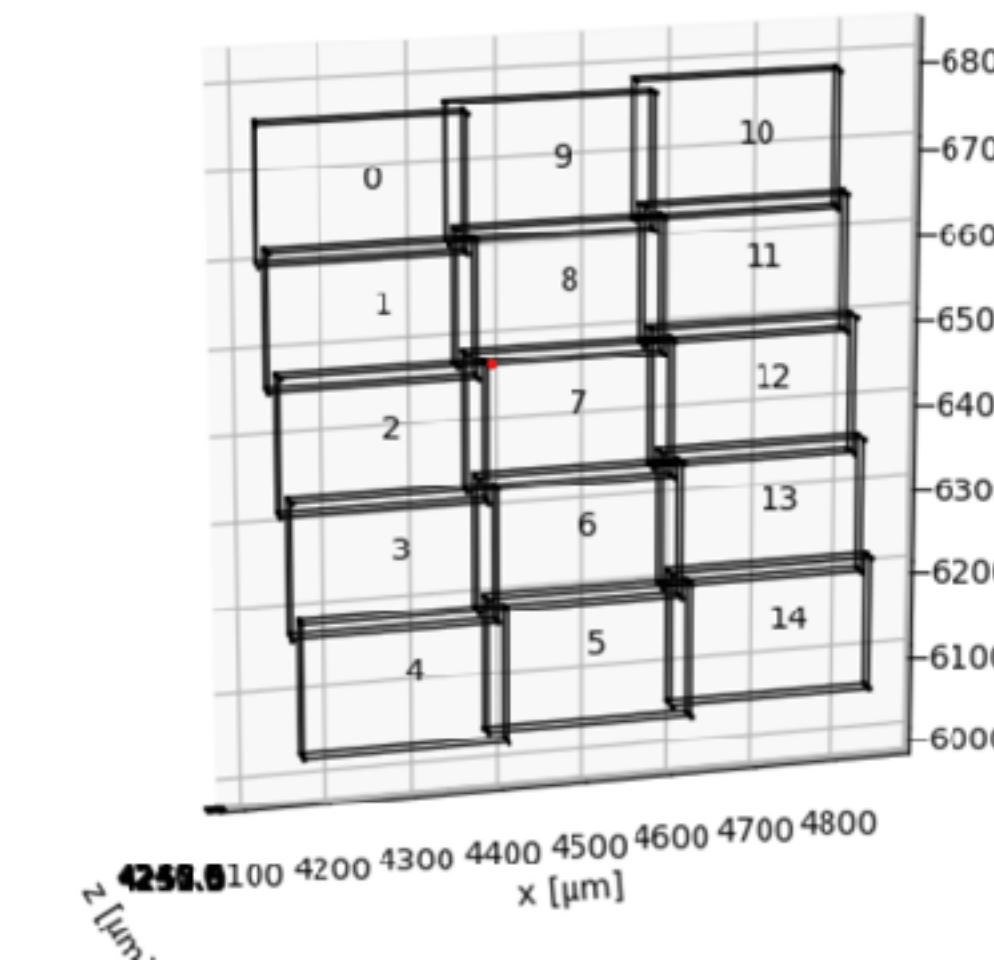


Artemiy  
Golden  
Alexander  
Wilhelmi  
University of Frankfurt



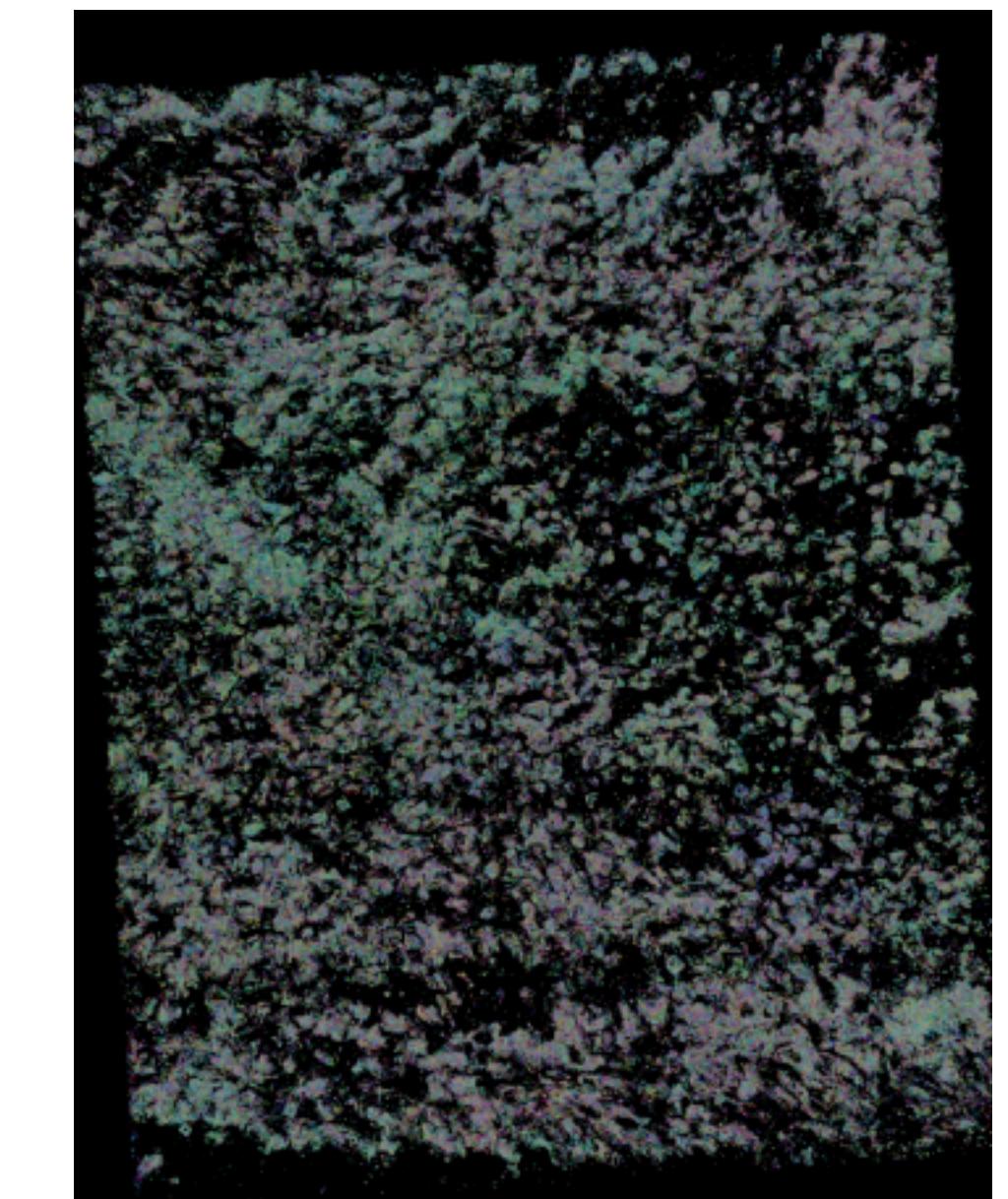
## Large FOV multiplexed light sheet imaging

### 3D Tile configuration



Doug Shepherd  
Arizona State University

### Decoded transcripts



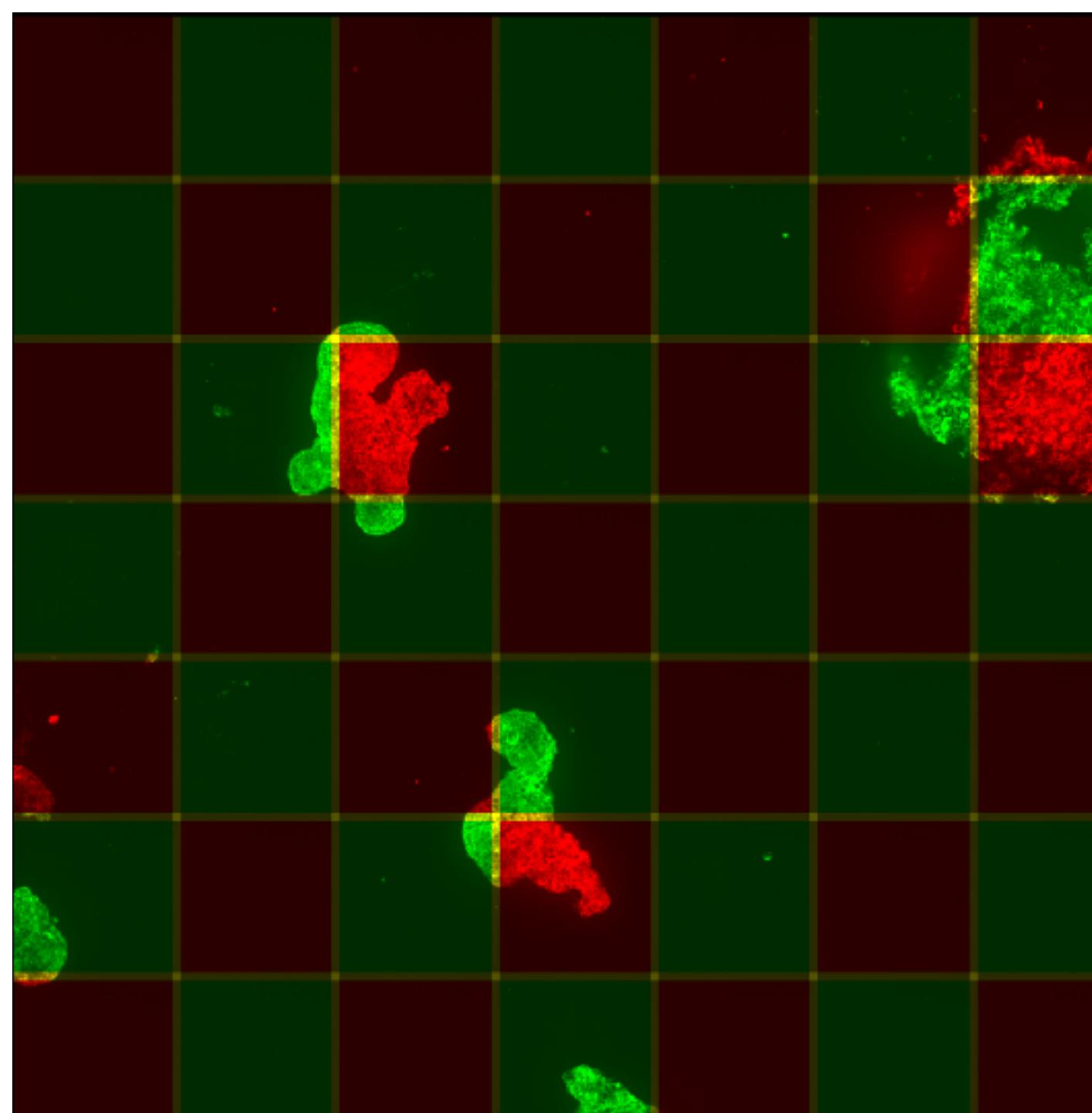
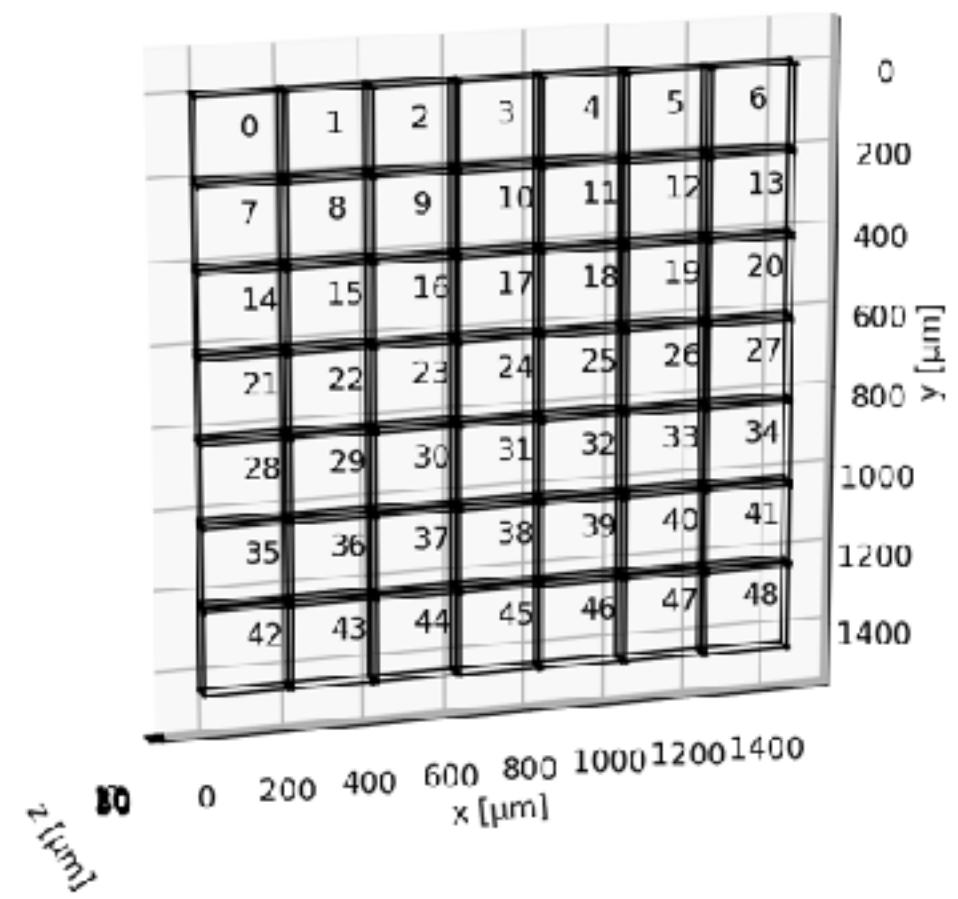


Nicole Repina  
Liberali Lab  
FMI Basel

Joel Luethi  
BioVisionCenter  
University of Zurich

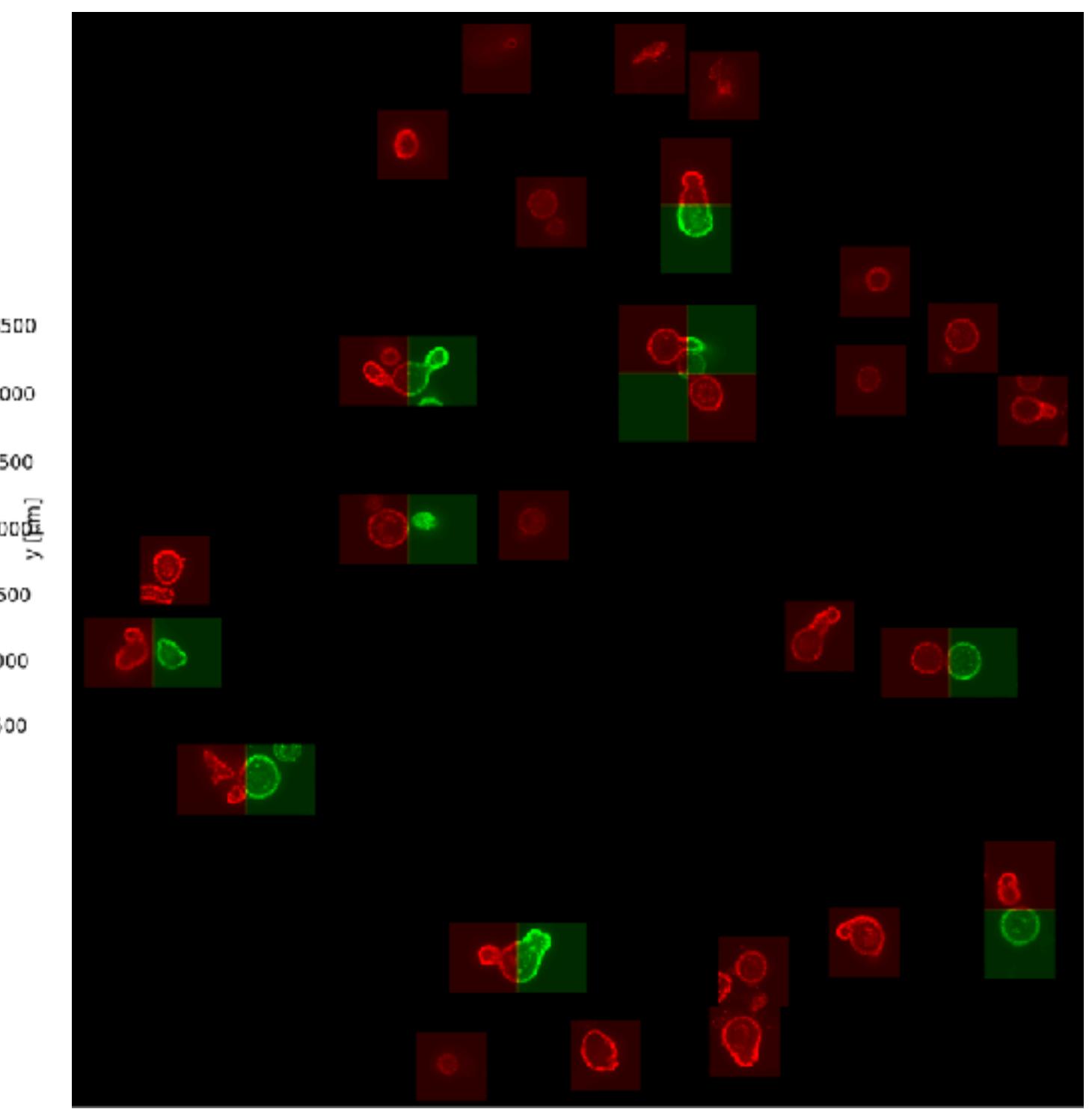
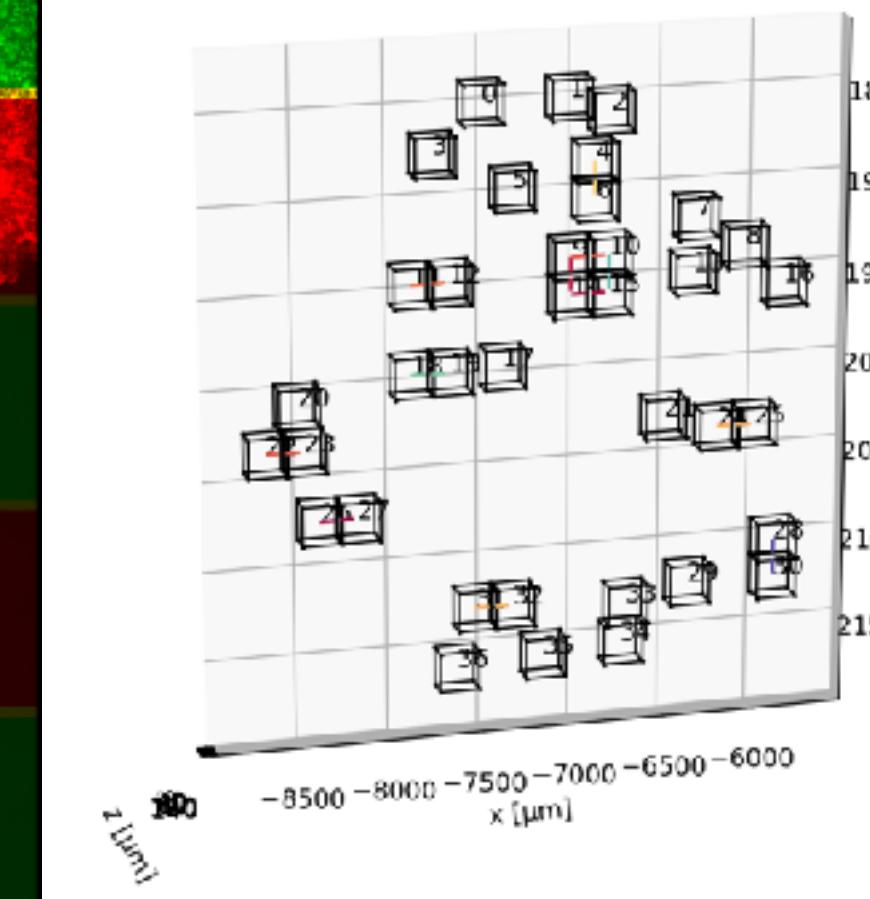
# Application: 3D HCS organoid imaging

Full well imaging



Max projection

“Search First” - targeted imaging



Central slice

Fractal task repo: <https://github.com/m-albert/fractal-ome-zarr-hcs-stitching>

# Workshop outline

## 1. Napari plugin demonstration

- 2D example
- 3D example



[github.com/m-albert/  
Virtual-I2K-2024-multiview-stitcher](https://github.com/m-albert/Virtual-I2K-2024-multiview-stitcher)

## 2. Python code example

- Loading 3D data
  - into memory
  - “lazily”
- Prepositioning tiles
- Registration: Translation and affine transforms
- Fusion: Linear blending and max projection
- Visualizing (intermediate) results