

Filtering and Segmentation Models for Computational Image Analysis

Alexandre Cunha

Center for Advanced Computing Research
Center for Integrative Study of Cell Regulation
California Institute of Technology

Computational Morphodynamics
KITP/UCSB - September 2009



Collaboration

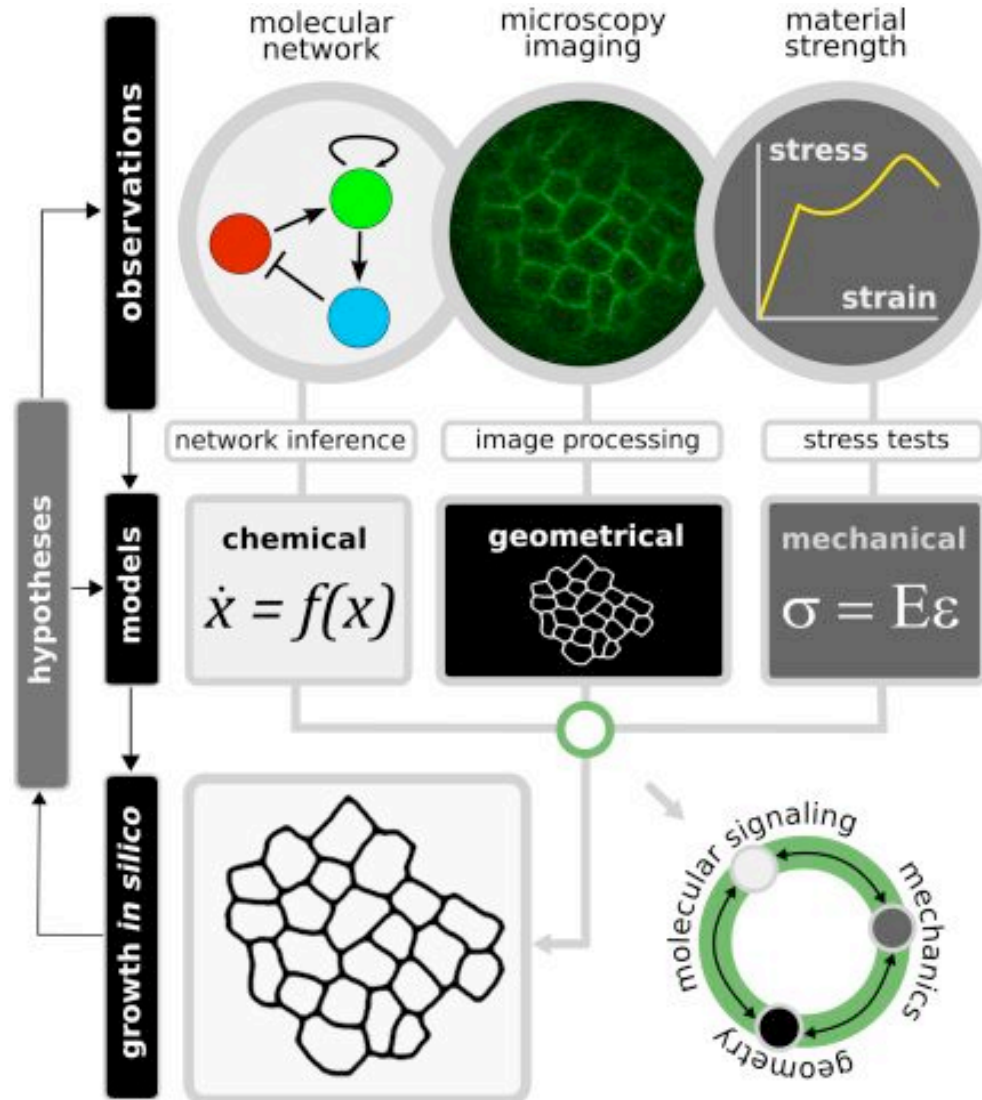
Work in collaboration with

- Jerome Darbon - Math, UCLA
- Grant Jensen - Structural Biologist (cryoEM), Caltech
- Morphodynamicists : Adrienne Roeder (sepals), Marcus Heisler, Paul Tarr, Cory Tobin (meristem), Vijay Chickarmane, Eric Mjolsness (UCI), Elliot Meyerowitz

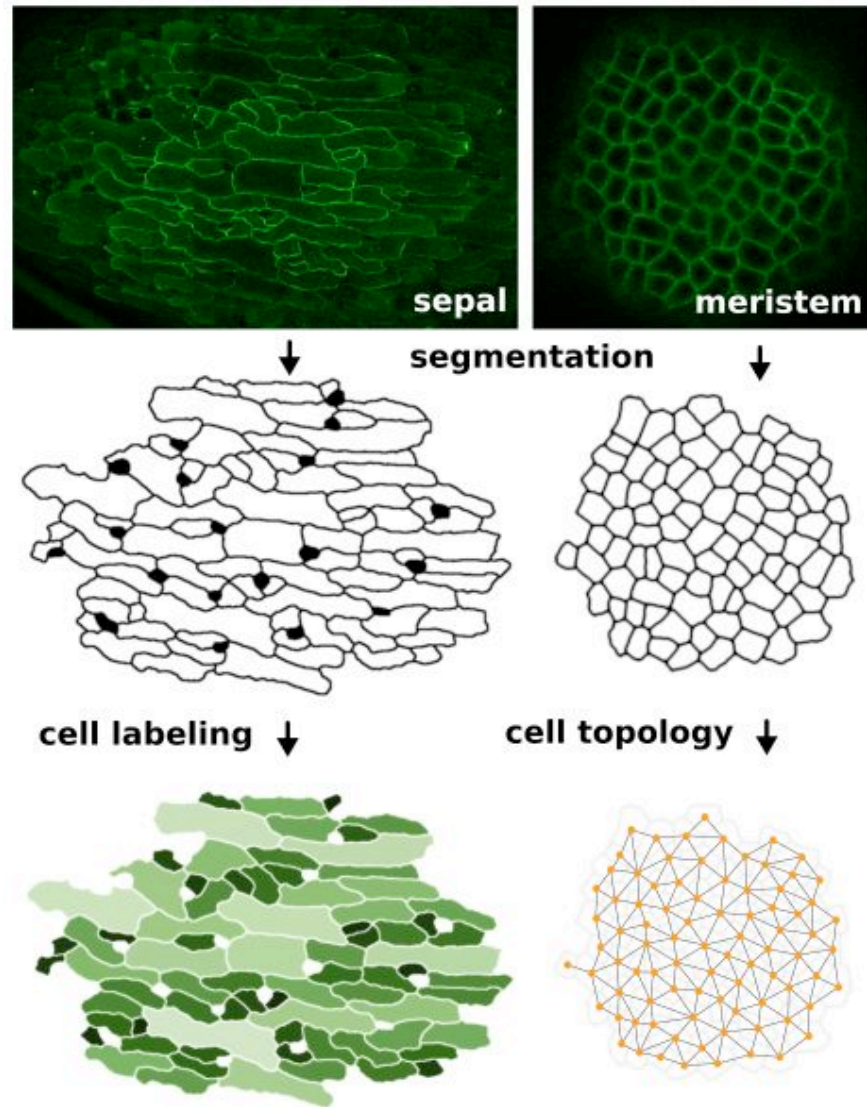
Partially supported by a gift grant from the Gordon and Betty Moore Foundation.



Morphodynamics and Imaging



Morphodynamics and Imaging



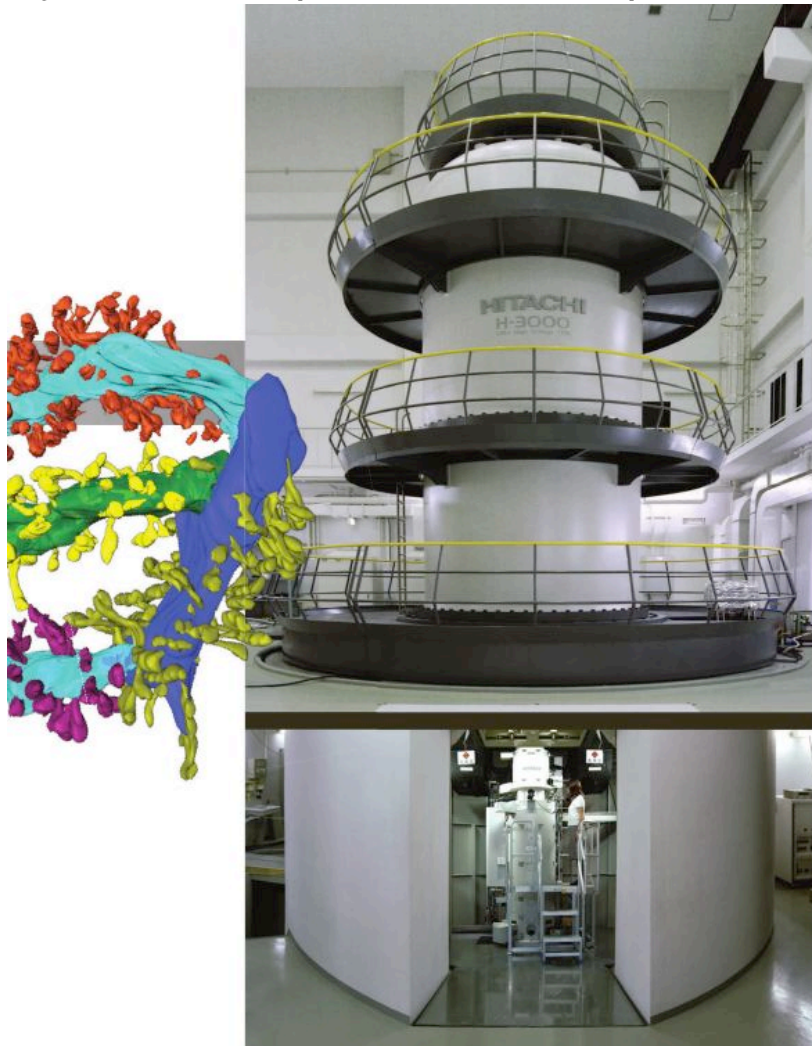
Outline

- 1 Morphodynamics
- 2 Content
- 3 Denoising
 - cryoEM
 - denoising model
- 4 Arabidopsis
 - examples
- 5 Segmentation
- 6 Math Morphology
 - Make it simple and human



Electron Cryo-Microscopy (cryoEM)

The Big and Bold: The electron cryo-microscope in Osaka, Japan.

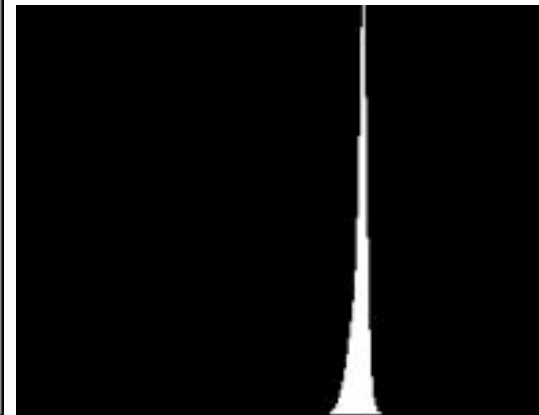
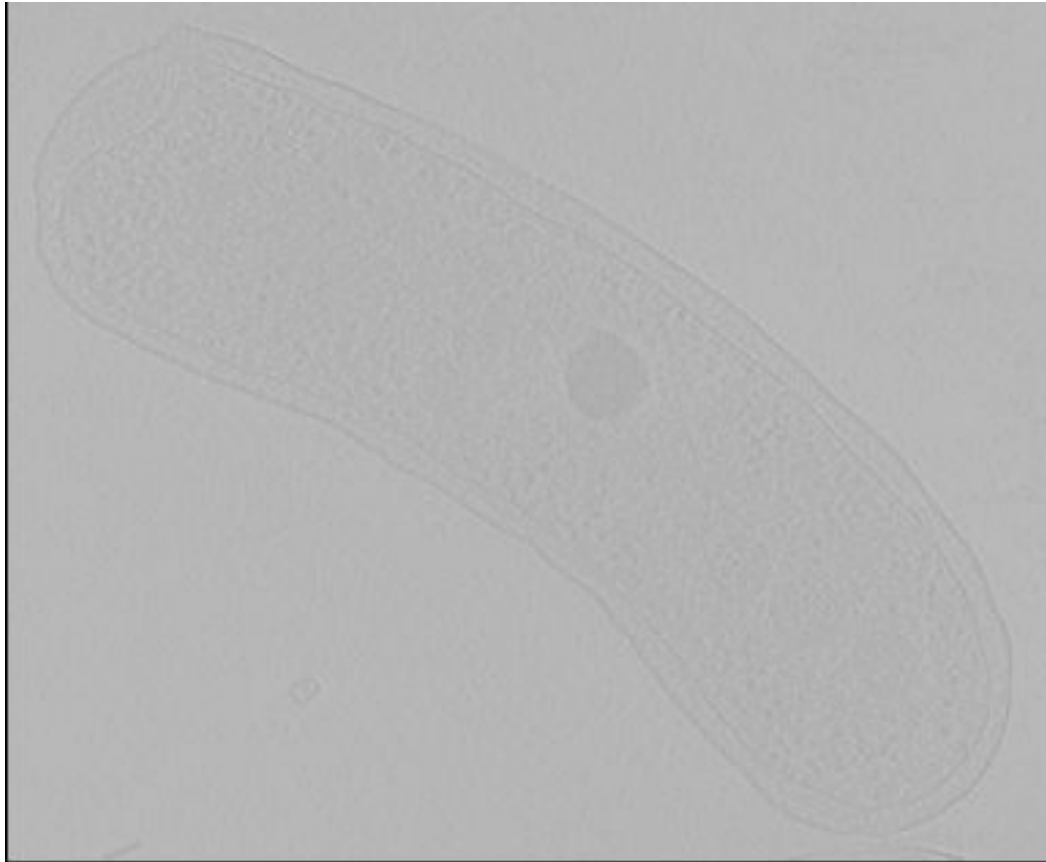


cryoEM is noisy

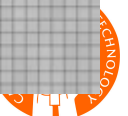
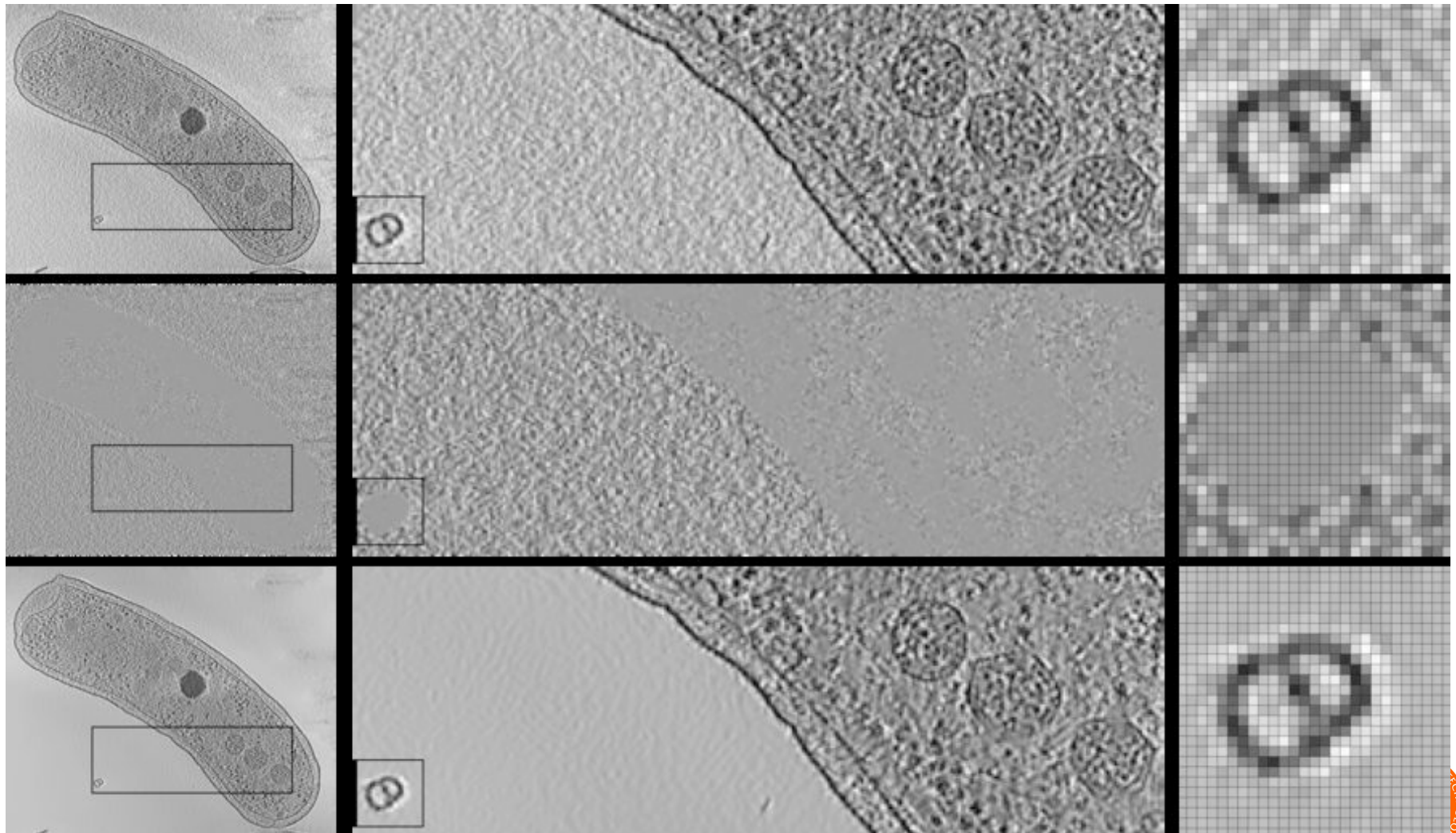
Images generated in cryoEM are typically low contrast, extremely noisy (SNR below 1), large in size (already at 8K x 8K) with cryo-tomograms having from million to billion voxels.



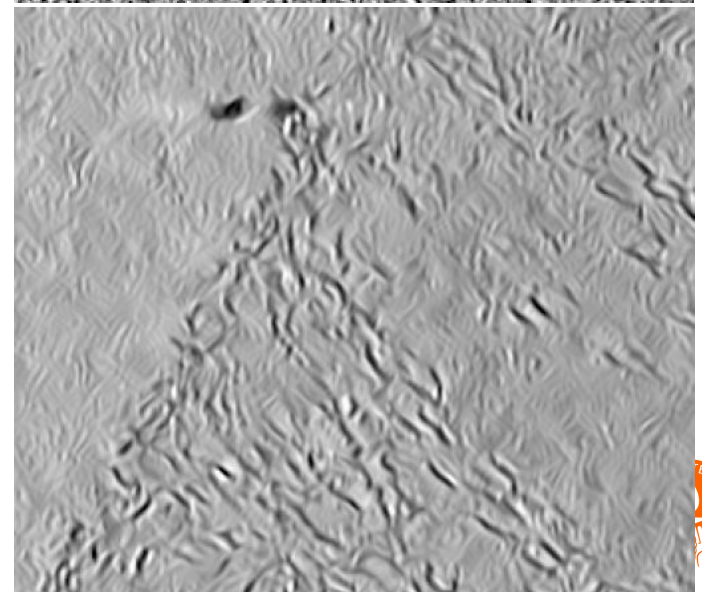
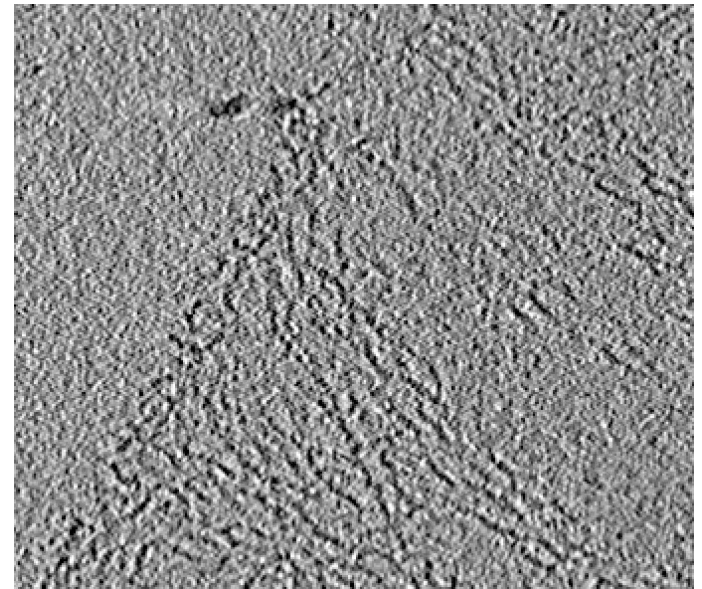
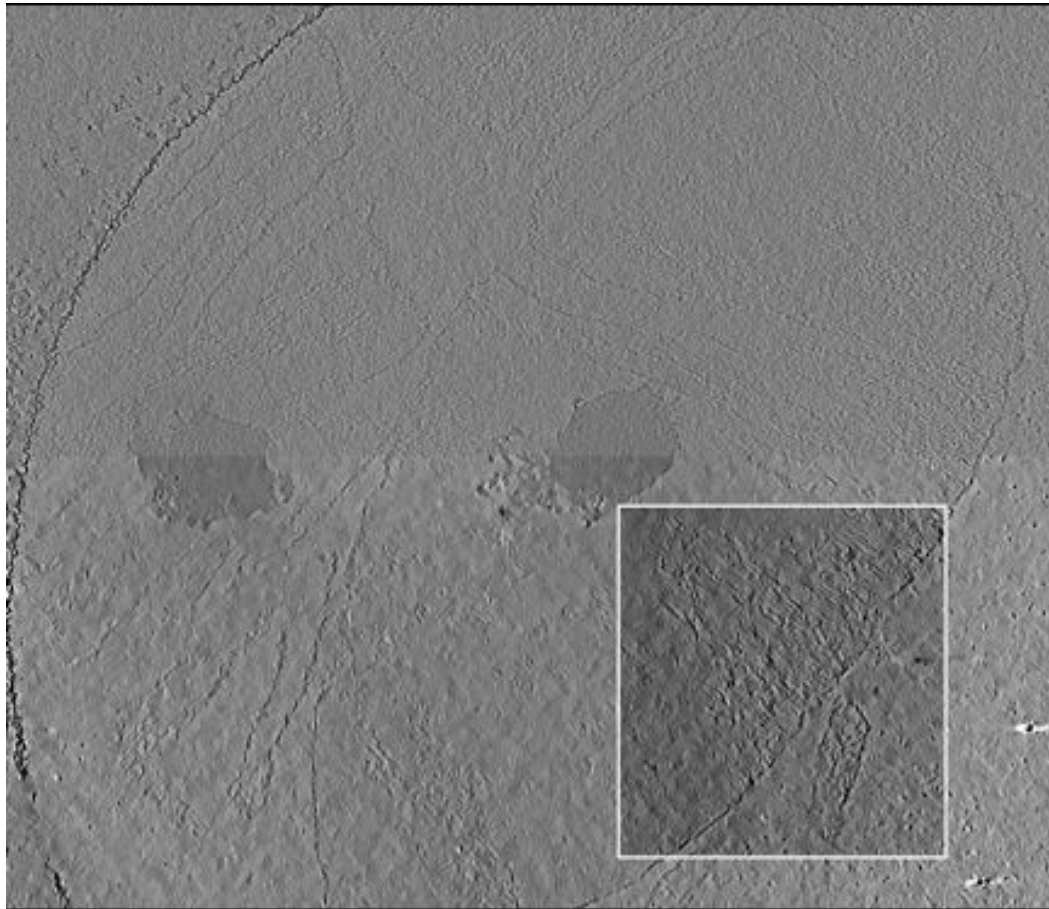
Typical cryoEM Image



Typical cryoEM Image

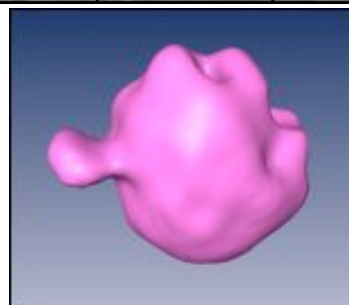
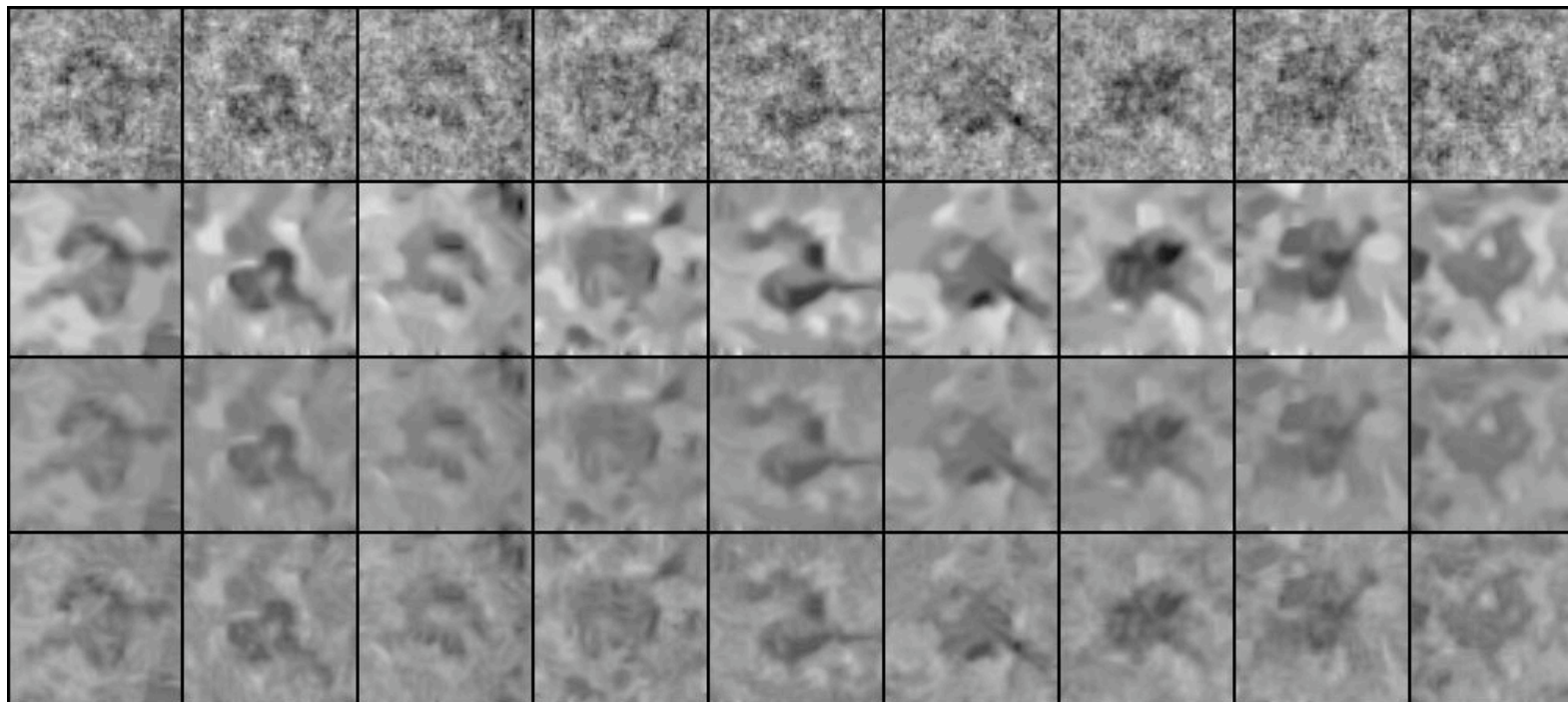


cryoEM - peptoglycan chain in *C. crescentus*



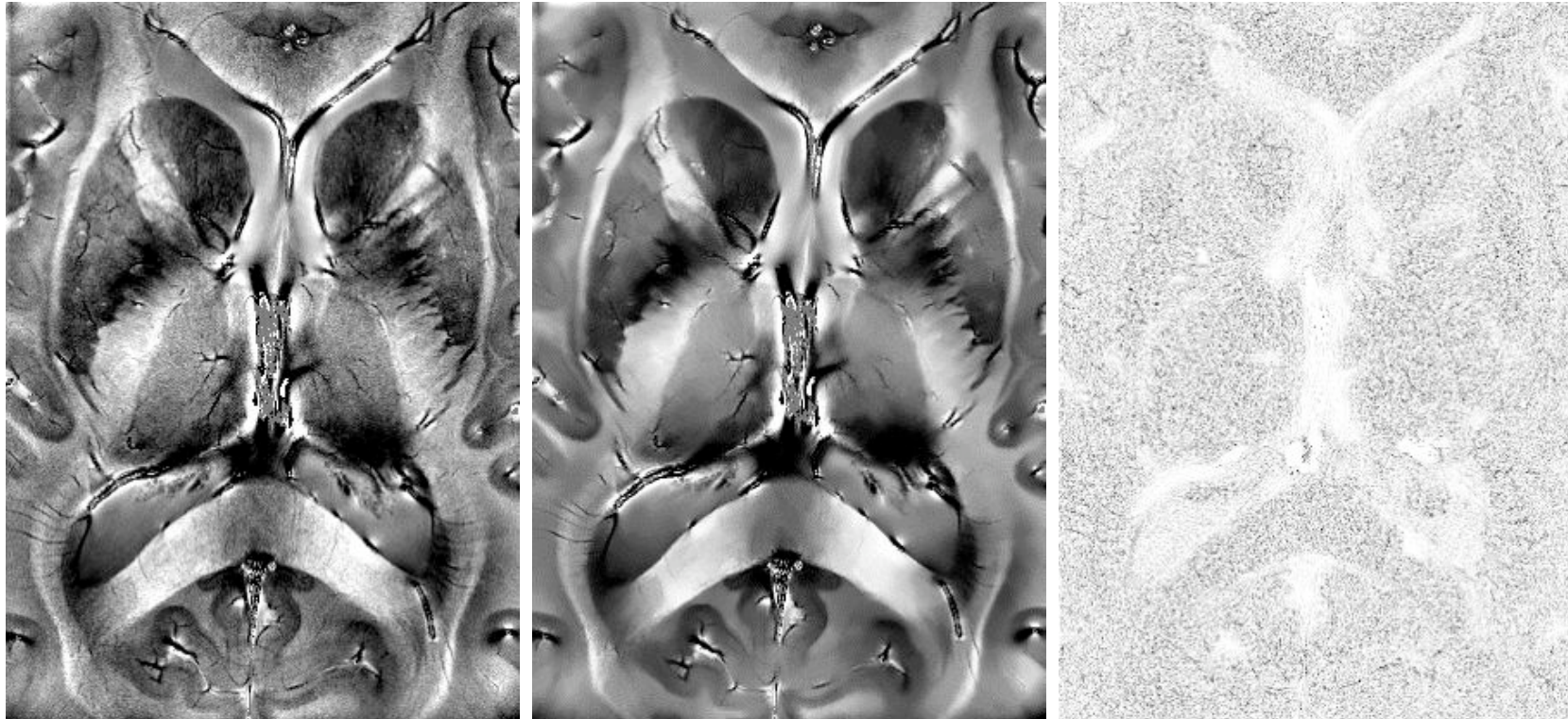
Fast Nonlocal Means Denoising of cryoEM Images, A. Cunha, J. Darbon. G.J. Jensen, Asia-Pacific Congress on Electron Tomography, Brisbane, Australia, 2009.

cryoEM - Single Particle Analysis



MRI

Denoising of a high resolution MRI image (acquired at 7.0 Tesla):



Efficient and Robust Restoration of High Resolution MRI, A. Cunha, J. Darbon, 5th European Congress on Computational Methods in Applied Sciences and Engineering, Venice, Italy, 2008.

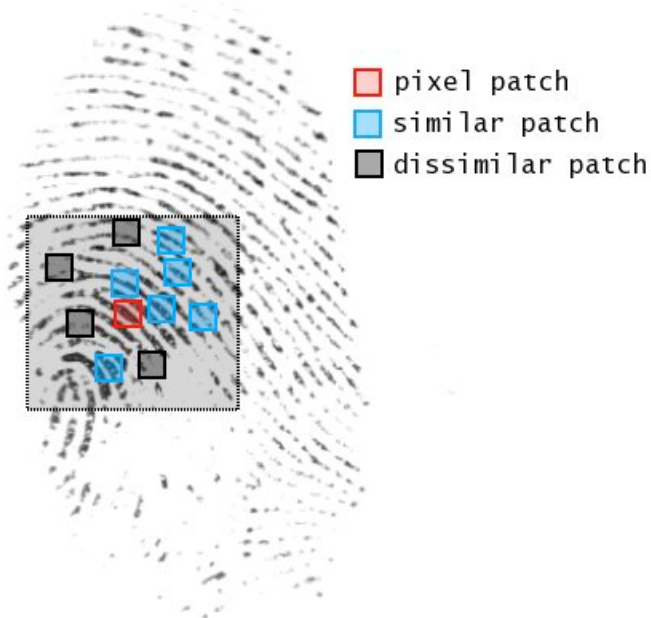


Nonlocal Means

Nonlocal Means

The nonlocal means approach is a neighborhood filtering scheme where similarity between patches around pixels is used to restore pixel values.

A review of image denoising algorithms, with a new one, A. Buades, B. Coll, J.M. Morel, SIAM Multiscale Model. Simul. 4(2), 2005.



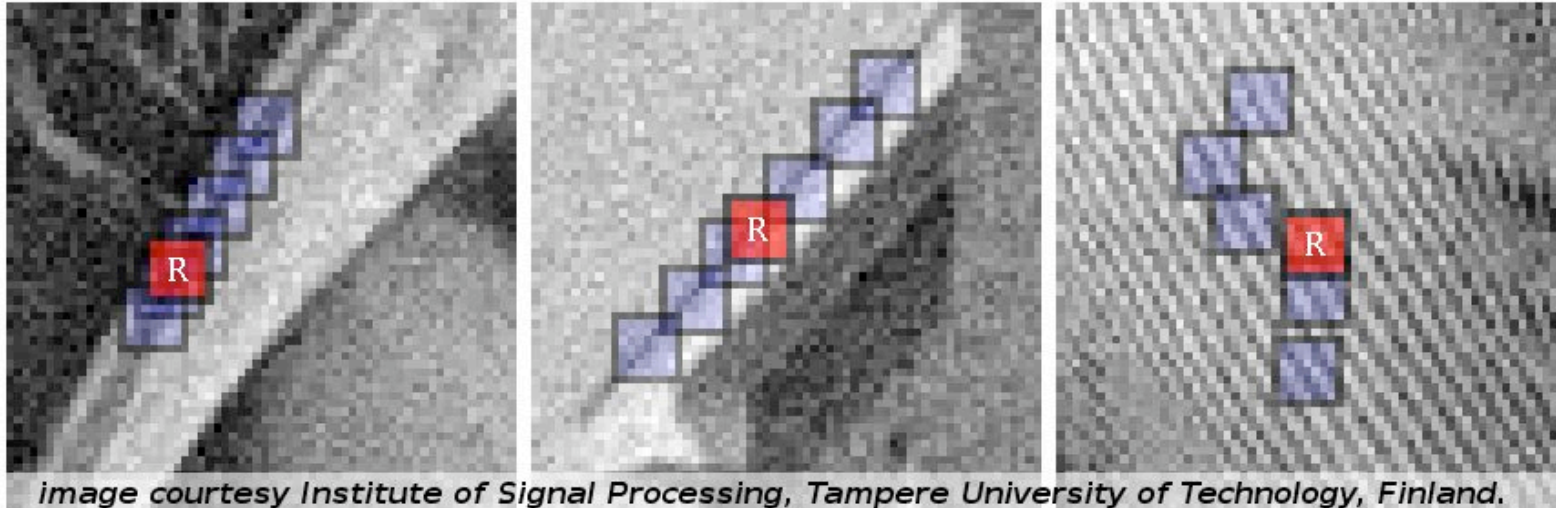
nonlocal means as a weighted average

When restoring the value u_i of pixel i nonlocal means considers the contribution of pixels j belonging to its neighborhood N_i where similarities w_{ij} between patches centered at i and j are taking into account:

$$u_i = \sum_{j \in N_i} \frac{w_{ij}}{\sum_k w_{ik}} v_j, \quad \forall i \in \Omega$$

The method restores images including textures while keeping the image geometry intact.

Nonlocal Means



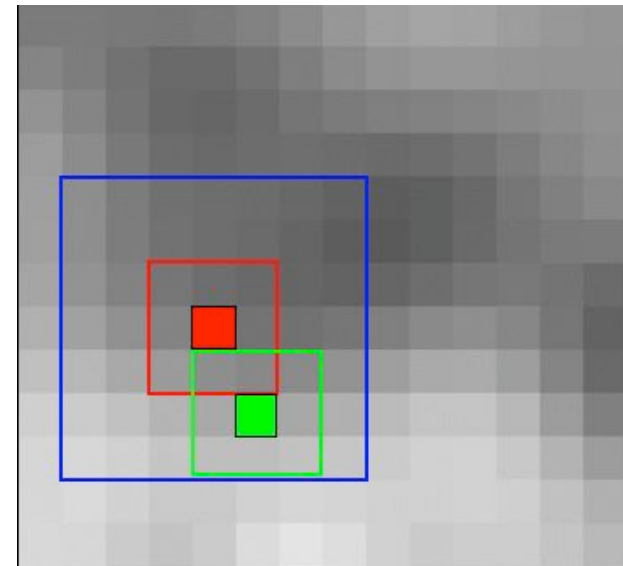
$$u_i = \sum_{j \in N_i} \frac{w_{ij}}{\sum_k w_{ik}} v_j$$

$$w_{ij} = e^{-d_{ij}^2 / h^2}$$

$$d_{ij} = \|v_{P_i} - v_{P_j}\|_{\sigma, L_p}^p, \quad p = 1, 2$$

parameters are then $h, |N_i|, |P_i|$

Computing the weights for every pixel can be quite expensive when using a standard sliding window approach.



Rewriting of Nonlocal Means

Fast Nonlocal Filtering Applied to Electron Cryomicroscopy, J. Darbon, A. Cunha, T.F. Chan, S. Osher, G.J. Jensen, IEEE ISBI 2008, pp. 1331-1334.

We depart from the sliding window approach and resort to more efficient strategies to deliver a fast algorithm and implementation in contemporary shared memory computer architectures. Some features of our implementation are:

- Instead of sliding images use shifted images to obtain differences in place
- In the tradeoff between memory and speed we favor speed (for every image we store another 4 extra auxiliary images to do fast computations)
- Partition the input image into as many computer cores as possible (domain decomposition)
- Vectorization of all operations (optimize cache locality)
- Use SIMD parallel instructions for single precision floats (AMD and Intel chipsets)



Rewriting of Nonlocal Means

Fast Nonlocal Filtering Applied to Electron Cryomicroscopy, J. Darbon, A. Cunha, T.F. Chan, S. Osher, G.J. Jensen, IEEE ISBI 2008, pp. 1331-1334.

We depart from the sliding window approach and resort to more efficient strategies to deliver a fast algorithm and implementation in contemporary shared memory computer architectures. Some features of our implementation are:

- Instead of sliding images use shifted images to obtain differences in place
- In the tradeoff between memory and speed we favor speed (for every image we store another 4 extra auxiliar images to do fast computations)
- Partition the input image into as many computer cores as possible (domain decomposition)
- Vectorization of all operations (optimize cache locality)
- Use SIMD parallel instructions for single precision floats (AMD and Intel chipsets)



Rewriting of Nonlocal Means

Fast Nonlocal Filtering Applied to Electron Cryomicroscopy, J. Darbon, A. Cunha, T.F. Chan, S. Osher, G.J. Jensen, IEEE ISBI 2008, pp. 1331-1334.

We depart from the sliding window approach and resort to more efficient strategies to deliver a fast algorithm and implementation in contemporary shared memory computer architectures. Some features of our implementation are:

- Instead of sliding images use shifted images to obtain differences in place
- In the tradeoff between memory and speed we favor speed (for every image we store another 4 extra auxiliar images to do fast computations)
- Partition the input image into as many computer cores as possible (domain decomposition)
- Vectorization of all operations (optimize cache locality)
- Use SIMD parallel instructions for single precision floats (AMD and Intel chipsets)



Rewriting of Nonlocal Means

Fast Nonlocal Filtering Applied to Electron Cryomicroscopy, J. Darbon, A. Cunha, T.F. Chan, S. Osher, G.J. Jensen, IEEE ISBI 2008, pp. 1331-1334.

We depart from the sliding window approach and resort to more efficient strategies to deliver a fast algorithm and implementation in contemporary shared memory computer architectures. Some features of our implementation are:

- Instead of sliding images use shifted images to obtain differences in place
- In the tradeoff between memory and speed we favor speed (for every image we store another 4 extra auxiliar images to do fast computations)
- Partition the input image into as many computer cores as possible (domain decomposition)
- Vectorization of all operations (optimize cache locality)
- Use SIMD parallel instructions for single precision floats (AMD and Intel chipsets)



Rewriting of Nonlocal Means

Fast Nonlocal Filtering Applied to Electron Cryomicroscopy, J. Darbon, A. Cunha, T.F. Chan, S. Osher, G.J. Jensen, IEEE ISBI 2008, pp. 1331-1334.

We depart from the sliding window approach and resort to more efficient strategies to deliver a fast algorithm and implementation in contemporary shared memory computer architectures. Some features of our implementation are:

- Instead of sliding images use shifted images to obtain differences in place
- In the tradeoff between memory and speed we favor speed (for every image we store another 4 extra auxiliary images to do fast computations)
- Partition the input image into as many computer cores as possible (domain decomposition)
- Vectorization of all operations (optimize cache locality)
- Use SIMD parallel instructions for single precision floats (AMD and Intel chipsets)



Rewriting of Nonlocal Means

Fast Nonlocal Filtering Applied to Electron Cryomicroscopy, J. Darbon, A. Cunha, T.F. Chan, S. Osher, G.J. Jensen, IEEE ISBI 2008, pp. 1331-1334.

We depart from the sliding window approach and resort to more efficient strategies to deliver a fast algorithm and implementation in contemporary shared memory computer architectures. Some features of our implementation are:

- Instead of sliding images use shifted images to obtain differences in place
- In the tradeoff between memory and speed we favor speed (for every image we store another 4 extra auxiliary images to do fast computations)
- Partition the input image into as many computer cores as possible (domain decomposition)
- Vectorization of all operations (optimize cache locality)
- Use SIMD parallel instructions for single precision floats (AMD and Intel chipsets)



Rewriting of Nonlocal Means

Fast Nonlocal Filtering Applied to Electron Cryomicroscopy, J. Darbon, A. Cunha, T.F. Chan, S. Osher, G.J. Jensen, IEEE ISBI 2008, pp. 1331-1334.

We depart from the sliding window approach and resort to more efficient strategies to deliver a fast algorithm and implementation in contemporary shared memory computer architectures. Some features of our implementation are:

- Instead of sliding images use shifted images to obtain differences in place
- In the tradeoff between memory and speed we favor speed (for every image we store another 4 extra auxiliary images to do fast computations)
- Partition the input image into as many computer cores as possible (domain decomposition)
- Vectorization of all operations (optimize cache locality)
- Use SIMD parallel instructions for single precision floats (AMD and Intel chipsets)



Timings

Timings, not included I/O, are averaged after 40 runs in the same image on a AMD x64, dual core, 2.8GHz platform. We achieve linear scalability.

Timings for filtering images of different sizes					
image size	256^2	512^2	1024^2	2048^2	4096^2
time (s)	0.07	0.35	1.54	6.05	26.32
scalability	–	5.0	4.4	3.9	4.4



Timings

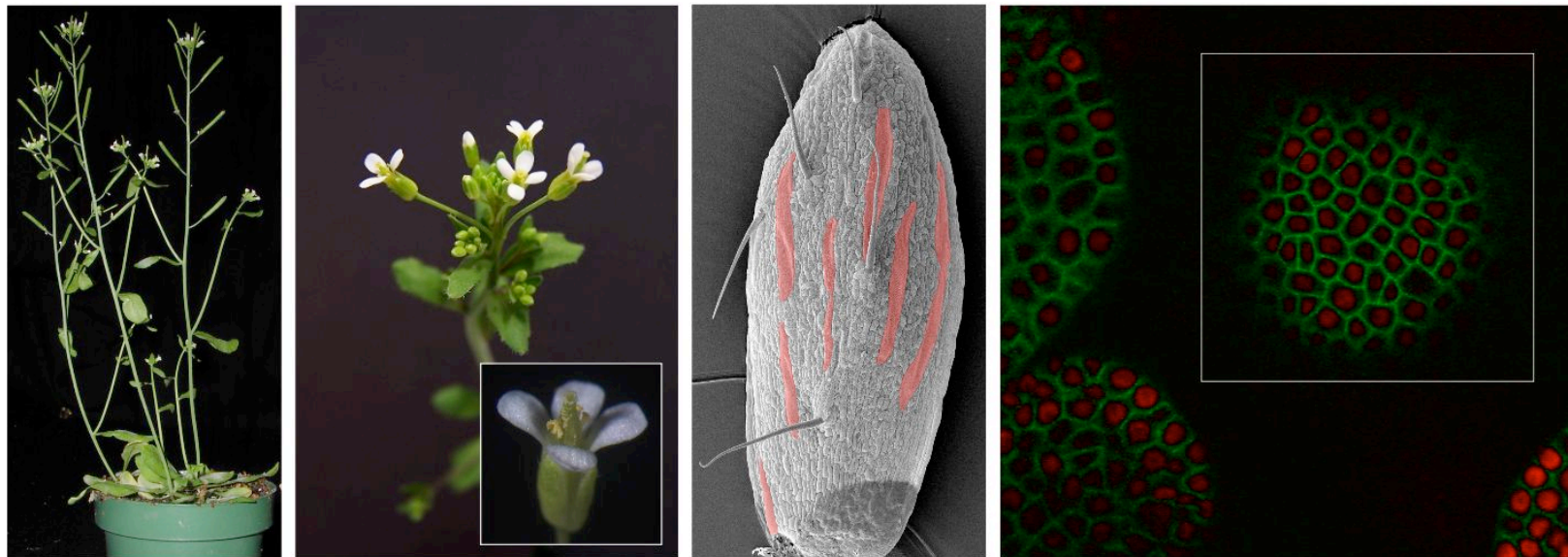
Timings, not included I/O, are averaged after 40 runs in the same image on a AMD x64, dual core, 2.8GHz platform. We achieve linear scalability.

Scalability results for a 8192 x 8192 cryoimage

version	cores	time(s)	speedup	efficiency(%)
serial	1	7,071.42	1	100
parallel	1	313.12	23	2,300
	2	163.59	43	2,150
	4	102.21	69	1,725
	8	66.83	106	1,325
	16	64.15	110	688



Arabidopsis thaliana



From left to right: A pot with *Arabidopsis* grown in the lab at Caltech. A close up of the sepals is shown next with some giant cells marked pink in a sepal in the 3rd column. Giant cells undergo endoreduplication where their DNA content duplicates but they do not divide thus growing much larger than other cells. Measuring cell size distribution and cell network topology helps in the investigation of a fundamental question in biology: how a pattern of different cell types develops from a field of relatively uniform cells. The patterning of giant cells clearly involves control of cell division or growth.



Denoising of meristem

Runtimes for filtering using 2 cpu cores

image size	256 ²	512 ²	1024 ²	2048 ²	4096 ²
time (s)	0.07	0.35	1.54	6.05	26.32
scalability	–	5.0	4.4	3.9	4.4

Images increase by a factor of 4.0 and time by an average of 4.4, showing linear scalability.

fNLMEANS implements the following algorithm. Let $v(x)$ be a noisy image defined on a grid \mathcal{G} . For each pixel $x \in \mathcal{G}$ the nonlocal means method produces a restored pixel $u(x)$ using a *weighted average*

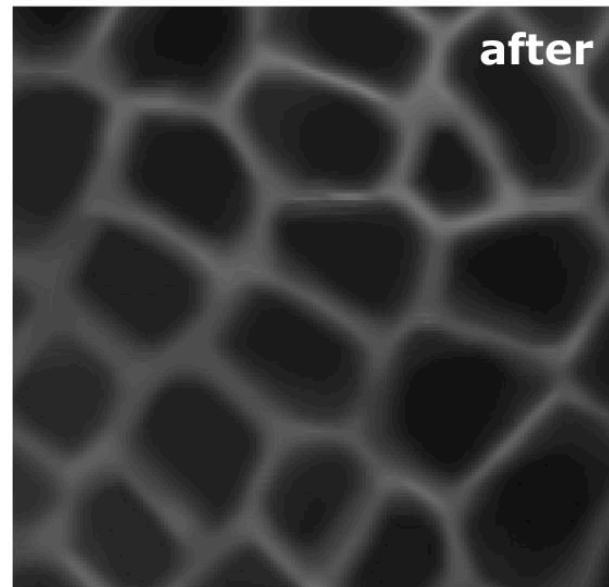
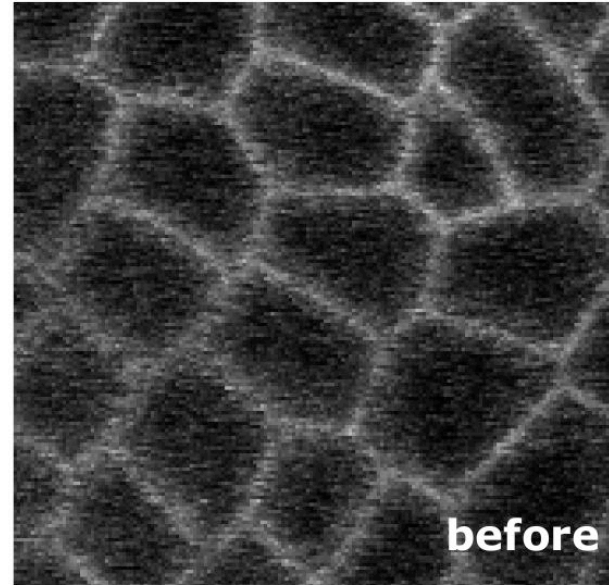
$$u(x) = \sum_{y \in \mathcal{N}_x} \frac{w(x, y)}{\sum_{z \neq x} w(x, z)} v(y)$$

where $w : \mathcal{G}^2 \rightarrow \mathbb{R}$ is the weight function between sites x and y , for y in the neighborhood $\mathcal{N}_x \subset \mathcal{G}$ of x , given in our new formulation by

$$w(x, y) = \frac{1}{1 + h^2 d^2(x, y)}.$$

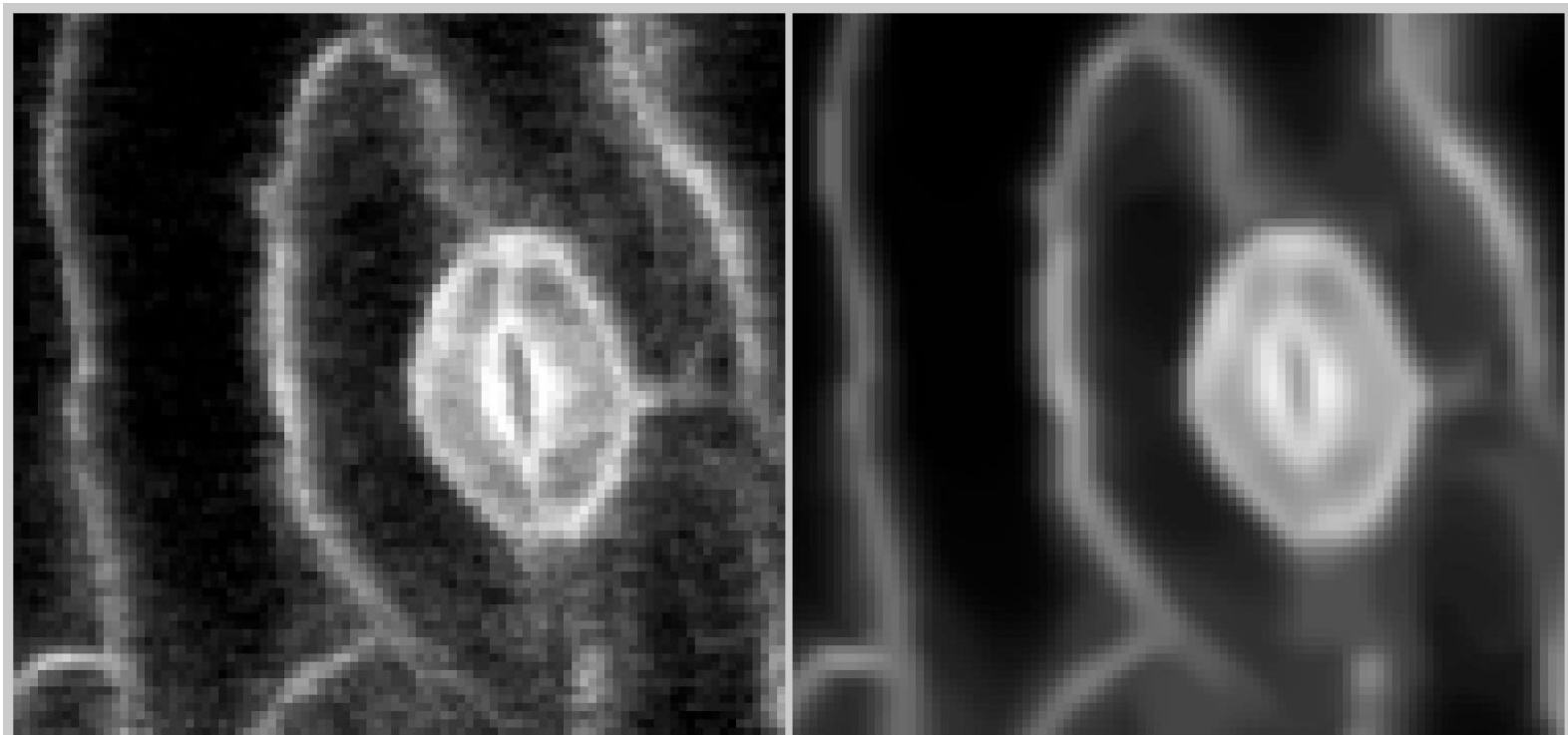
$h \in \mathbb{R}$ is a filtering intensity parameter and $d^2(x, y)$ is the square of the L_p norm ($p = 1, 2$) of the *similarity distance* between square patches \mathcal{P}_x and \mathcal{P}_y centered, respectively, at x and y . If $v_x = v(\mathcal{P}_x)$ is the vector of pixel values in the square \mathcal{P}_x , then

$$d^2 = \|v_x - v_y\|_p^2 = \left(\sum_{i=1}^{|\mathcal{P}_x|} |v_{i_x} - v_{i_y}|^p \right)^{2/p}$$



fNLMEANS: Fast Nonlocal Means

And example of the robust denoising method. We apply denoising in images before further processing. It does help in segmentation and it is a major contributor to the edge localization method we use to segment cells in sepals and meristem.



ACTIVE: Active Contours Without Edges

ACTIVE is an implementation of the *Active Contours Without Edges* variational model for segmenting gray level images, introduced by Chan and Vese (IEEE TIP 10(2), 2001):

$$\min_{\phi} \rightarrow \mathcal{J}(\phi, c_1, c_2) = \int_{\Omega} (u - c_1)^2 H_{\varepsilon}(\phi) dx + \int_{\Omega} (u - c_2)^2 (1 - H_{\varepsilon}(\phi)) dx + \mu \int_{\Omega} |\nabla H_{\varepsilon}(\phi)| dx$$

where $H_{\varepsilon}(\phi) \in [0, 1]$ is the ε -mollified Heaviside of the level set function $\phi(x)$ and

$$c_1 = \frac{\int u H(\phi(x)) dx}{\int H(\phi(x)) dx} \quad (\text{average inside})$$

$$c_2 = \frac{\int u (1 - H(\phi(x))) dx}{\int (1 - H(\phi(x))) dx} \quad (\text{average outside})$$

This level set based model can be viewed as a rubber band (the interface where $\phi(x) = 0$) with stiffness μ separating an image into inside and outside regions having, respectively, average intensities c_1 and c_2 .



ACTIVE: Active Contours Without Edges

We cast the minimization problem as a geometric flow of the corresponding Euler-Lagrange PDE of the energy:

$$\frac{d\phi}{dt} = \delta_\varepsilon(\phi)[\mu\kappa(\phi) - (u - c_1)^2 + (u - c_2)^2]$$

where $\delta_\varepsilon(\phi)$ is the Dirac delta applied to ϕ and $\kappa(\phi)$ is the curvature of the rubber. We solve this PDE using a stable (CFL condition) finite difference scheme:

- Starting with an initial guess ϕ_0 , compute c_1^0, c_2^0
- Propagate ϕ according to the discretization of the above PDE,
 $\phi^k = f(\phi^{k-1}, c_1^{k-1}, c_2^{k-1}, \Delta t)$
- Recompute c_1, c_2 for the newly computed ϕ^k and repeat
- Convergence is achieved when $|c_1^k - c_1^{k-1}| < \epsilon$ (same for c_2) or a maximum number of iterations is achieved



ACTIVE: Active Contours Without Edges

We cast the minimization problem as a geometric flow of the corresponding Euler-Lagrange PDE of the energy:

$$\frac{d\phi}{dt} = \delta_\varepsilon(\phi)[\mu\kappa(\phi) - (u - c_1)^2 + (u - c_2)^2]$$

where $\delta_\varepsilon(\phi)$ is the Dirac delta applied to ϕ and $\kappa(\phi)$ is the curvature of the rubber. We solve this PDE using a stable (CFL condition) finite difference scheme:

- Starting with an initial guess ϕ_0 , compute c_1^0, c_2^0
- Propagate ϕ according to the discretization of the above PDE,
 $\phi^k = f(\phi^{k-1}, c_1^{k-1}, c_2^{k-1}, \Delta t)$
- Recompute c_1, c_2 for the newly computed ϕ^k and repeat
- Convergence is achieved when $|c_1^k - c_1^{k-1}| < \epsilon$ (same for c_2) or a maximum number of iterations is achieved



ACTIVE: Active Contours Without Edges

We cast the minimization problem as a geometric flow of the corresponding Euler-Lagrange PDE of the energy:

$$\frac{d\phi}{dt} = \delta_\varepsilon(\phi)[\mu\kappa(\phi) - (u - c_1)^2 + (u - c_2)^2]$$

where $\delta_\varepsilon(\phi)$ is the Dirac delta applied to ϕ and $\kappa(\phi)$ is the curvature of the rubber. We solve this PDE using a stable (CFL condition) finite difference scheme:

- Starting with an initial guess ϕ_0 , compute c_1^0, c_2^0
- Propagate ϕ according to the discretization of the above PDE,
 $\phi^k = f(\phi^{k-1}, c_1^{k-1}, c_2^{k-1}, \Delta t)$
- Recompute c_1, c_2 for the newly computed ϕ^k and repeat
- Convergence is achieved when $|c_1^k - c_1^{k-1}| < \epsilon$ (same for c_2) or a maximum number of iterations is achieved



ACTIVE: Active Contours Without Edges

We cast the minimization problem as a geometric flow of the corresponding Euler-Lagrange PDE of the energy:

$$\frac{d\phi}{dt} = \delta_\varepsilon(\phi)[\mu\kappa(\phi) - (u - c_1)^2 + (u - c_2)^2]$$

where $\delta_\varepsilon(\phi)$ is the Dirac delta applied to ϕ and $\kappa(\phi)$ is the curvature of the rubber. We solve this PDE using a stable (CFL condition) finite difference scheme:

- Starting with an initial guess ϕ_0 , compute c_1^0, c_2^0
- Propagate ϕ according to the discretization of the above PDE,
 $\phi^k = f(\phi^{k-1}, c_1^{k-1}, c_2^{k-1}, \Delta t)$
- Recompute c_1, c_2 for the newly computed ϕ^k and repeat
- Convergence is achieved when $|c_1^k - c_1^{k-1}| < \epsilon$ (same for c_2) or a maximum number of iterations is achieved



ACTIVE - Whole Sepal Segmentation

Example: segmenting a whole sepal (its gray level version):

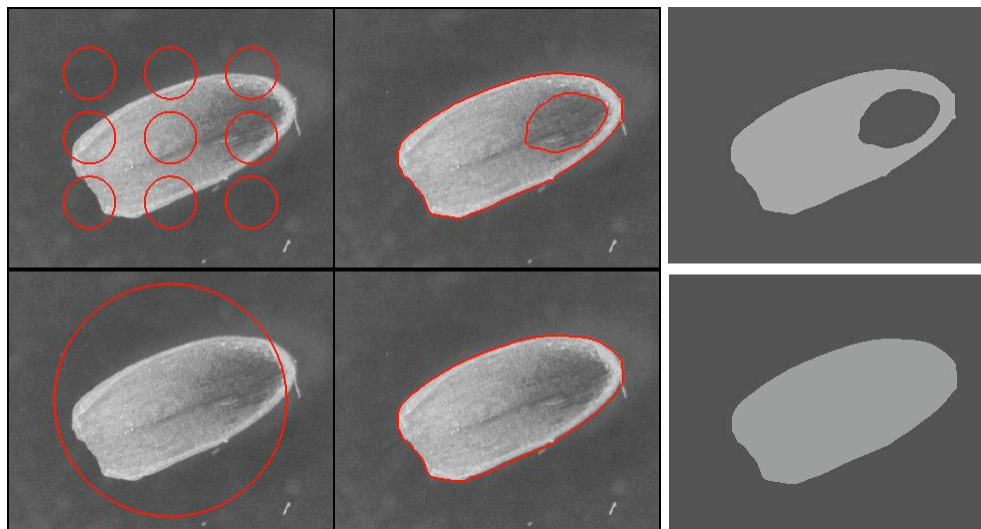
Screen

Screen



ACTIVE - Whole Sepal Segmentation

Achieving good results do depend on initial conditions: the initial zero level set interface (where do we start the rubber) and the rubber stiffness are major players.



- **top row:** starting from 6 small circles and a soft rubber ($\mu = 0.2$) leads to disjoint regions (a correct result as the inner and most outer regions have similar average intensities - see 3rd column)
- **bottom row:** using a single initial circle with a stronger rubber ($\mu = 0.5$) gives the desired result.



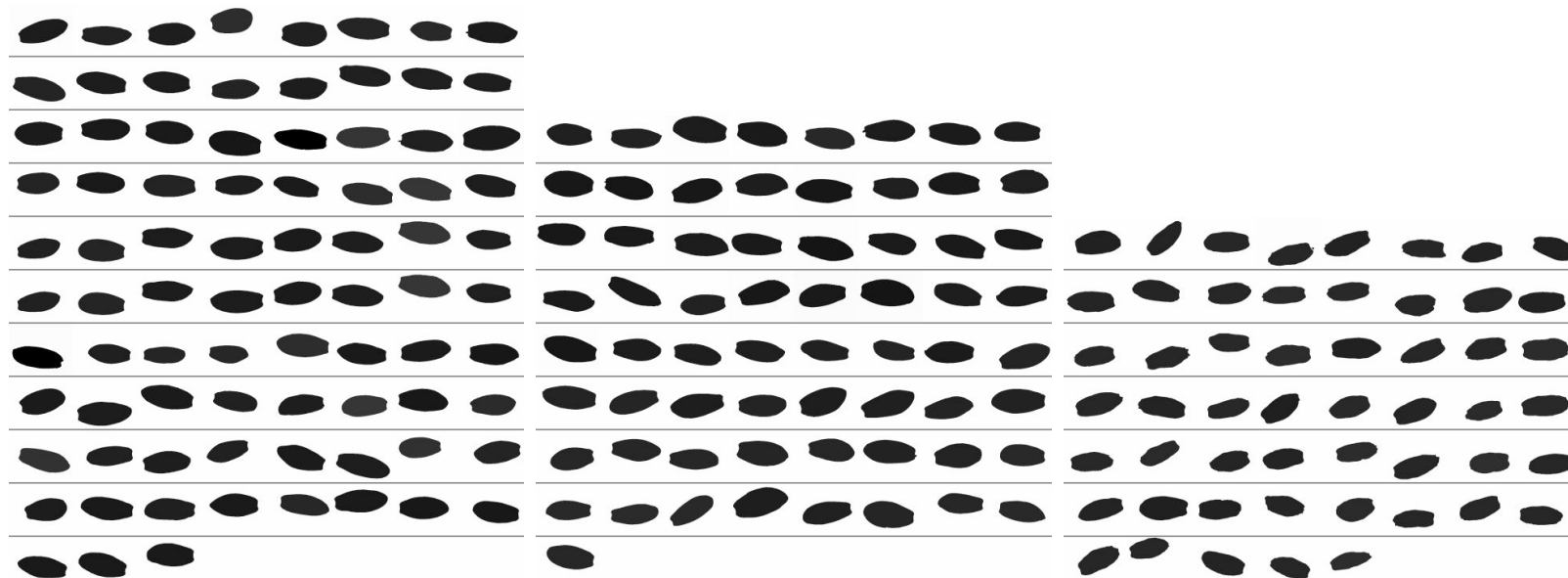
ACTIVE - Whole Sepal Segmentation

By knowing the limitations of the method the biologist could offer a simple alternative to increase the success rate of the algorithm: make sepals more homogeneous by staining them with a stronger marker:



ACTIVE - Whole Sepal Segmentation

We then achieve a fully automatic segmentation with 100% success rate. Below are automatically segmented masks for 3 different sets of sepals (mutants and wild):



ACTIVE - Segmenting Sepal Cells

ACTIVE segmentation of cells in a sepal. We experimentally confirmed the benefits of denoising before segmenting with the Chan & Vese model even though the model itself is capable of segmenting noisy patterns.

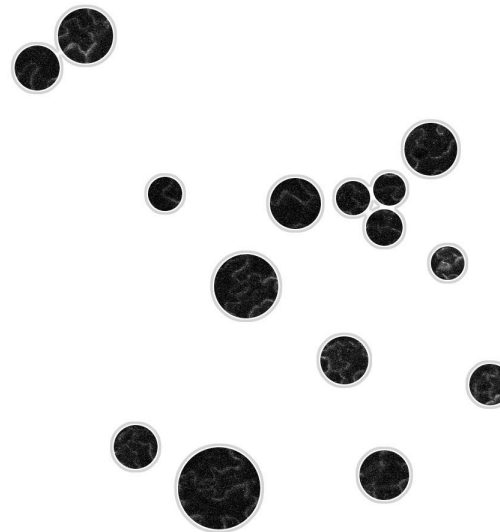
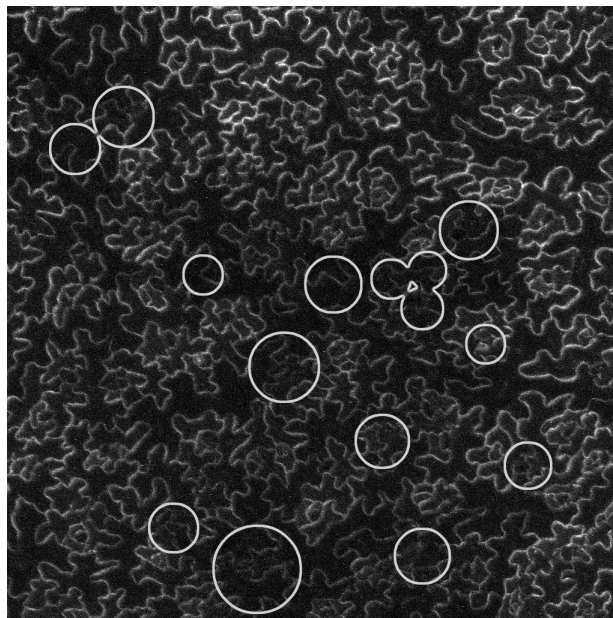
Screen

Screen



Poor quality images

The karma of poor quality images is not going to disappear. We most often have to deal with images acquired in less than perfect conditions, containing different, non-uniform patterns of defects. *We sometimes can't afford discarding high value images, taken of plants in vivo.*



We aim to offer the user/biologist means to fix errors either due to the algorithms or due to poor image quality. This is in line with the principles of the contemporary *Human Computation* paradigm.





Human Computation


Human Computation : because images can be quite complex and computer vision solutions to interpret them are limited.

Location:





Word Verification: Type the characters you see in the picture below.




Letters are not case-sensitive



Type the two words:

    stop spam. read books.

Scanned type → **This aged portion of society were distinguished from**

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

OCR reads as → "niis aged pntkm at society were distinguished frow."

reCAPTCHA: Human-Based Character Recognition via Web Security Measures, L. von Ahn, B. Maurer, C. McMillen, D. Abraham, M. Blum, Science 321, Sep. 2008.



SEMSE^h

Computational Morphogenesis. Tools for Simulating Cell Growth and Division in Arabidopsis thaliana, A. Cunha, A. Roeder, M. Heisler, E. Mjolsness, E. Meyerowitz, Bioimage Informatics Conference, HHMI, April 2009.

SEMSE^h stands for *Segmentation Made Simple and Human*. It is our attempt to (re)introduce the human factor to deal with problems computationally intractable. We can't expect to write robust algorithms to detect and fix general defects (missing information) in poor quality images. We ask the human to do it instead.

Advantages:

- + Empower the end user to intervene in the delivery of expected results
- + Provide solutions in a fraction of time as compared to usual approaches (lengthy algorithm + code development)
- + Control the quality of results without distorting data
- + Provide feedback that might leverage algorithm development

Disadvantages:

- - Manual intervention/editing only feasible for small data sets
- - Disagreement between different users of what is correct



SEMSE^h

Computational Morphogenesis. Tools for Simulating Cell Growth and Division in Arabidopsis thaliana, A. Cunha, A. Roeder, M. Heisler, E. Mjolsness, E. Meyerowitz, Bioimage Informatics Conference, HHMI, April 2009.

SEMSE^h stands for *Segmentation Made Simple and Human*. It is our attempt to (re)introduce the human factor to deal with problems computationally intractable. We can't expect to write robust algorithms to detect and fix general defects (missing information) in poor quality images. We ask the human to do it instead.

Advantages:

- + Empower the end user to intervene in the delivery of expected results
- + Provide solutions in a fraction of time as compared to usual approaches (lengthy algorithm + code development)
- + Control the quality of results without distorting data
- + Provide feedback that might leverage algorithm development

Disadvantages:

- - Manual intervention/editing only feasible for small data sets
- - Disagreement between different users of what is correct



SEMSE^h

Computational Morphogenesis. Tools for Simulating Cell Growth and Division in Arabidopsis thaliana, A. Cunha, A. Roeder, M. Heisler, E. Mjolsness, E. Meyerowitz, Bioimage Informatics Conference, HHMI, April 2009.

SEMSE^h stands for *Segmentation Made Simple and Human*. It is our attempt to (re)introduce the human factor to deal with problems computationally intractable. We can't expect to write robust algorithms to detect and fix general defects (missing information) in poor quality images. We ask the human to do it instead.

Advantages:

- + Empower the end user to intervene in the delivery of expected results
- + Provide solutions in a fraction of time as compared to usual approaches (lengthy algorithm + code development)
- + Control the quality of results without distorting data
- + Provide feedback that might leverage algorithm development

Disadvantages:

- - Manual intervention/editing only feasible for small data sets
- - Disagreement between different users of what is correct



SEMSE^h

Computational Morphogenesis. Tools for Simulating Cell Growth and Division in Arabidopsis thaliana, A. Cunha, A. Roeder, M. Heisler, E. Mjolsness, E. Meyerowitz, Bioimage Informatics Conference, HHMI, April 2009.

SEMSE^h stands for *Segmentation Made Simple and Human*. It is our attempt to (re)introduce the human factor to deal with problems computationally intractable. We can't expect to write robust algorithms to detect and fix general defects (missing information) in poor quality images. We ask the human to do it instead.

Advantages:

- + Empower the end user to intervene in the delivery of expected results
- + Provide solutions in a fraction of time as compared to usual approaches (lengthy algorithm + code development)
- + Control the quality of results without distorting data
- + Provide feedback that might leverage algorithm development

Disadvantages:

- - Manual intervention/editing only feasible for small data sets
- - Disagreement between different users of what is correct



SEMSE^h

Computational Morphogenesis. Tools for Simulating Cell Growth and Division in Arabidopsis thaliana, A. Cunha, A. Roeder, M. Heisler, E. Mjolsness, E. Meyerowitz, Bioimage Informatics Conference, HHMI, April 2009.

SEMSE^h stands for *Segmentation Made Simple and Human*. It is our attempt to (re)introduce the human factor to deal with problems computationally intractable. We can't expect to write robust algorithms to detect and fix general defects (missing information) in poor quality images. We ask the human to do it instead.

Advantages:

- + Empower the end user to intervene in the delivery of expected results
- + Provide solutions in a fraction of time as compared to usual approaches (lengthy algorithm + code development)
- + Control the quality of results without distorting data
- + Provide feedback that might leverage algorithm development

Disadvantages:

- - Manual intervention/editing only feasible for small data sets
- - Disagreement between different users of what is correct



SEMSE^h

Computational Morphogenesis. Tools for Simulating Cell Growth and Division in Arabidopsis thaliana, A. Cunha, A. Roeder, M. Heisler, E. Mjolsness, E. Meyerowitz, Bioimage Informatics Conference, HHMI, April 2009.

SEMSE^h stands for *Segmentation Made Simple and Human*. It is our attempt to (re)introduce the human factor to deal with problems computationally intractable. We can't expect to write robust algorithms to detect and fix general defects (missing information) in poor quality images. We ask the human to do it instead.

Advantages:

- + Empower the end user to intervene in the delivery of expected results
- + Provide solutions in a fraction of time as compared to usual approaches (lengthy algorithm + code development)
- + Control the quality of results without distorting data
- + Provide feedback that might leverage algorithm development

Disadvantages:

- - Manual intervention/editing only feasible for small data sets
- - Disagreement between different users of what is correct



SEMSE^h and Mathematical Morphology

We realize SEMSE^h through the combination of our robust filter and the fast and elegant solutions provided by *mathematical morphology operators*.

Steps we adopt in segmenting with math morphology:

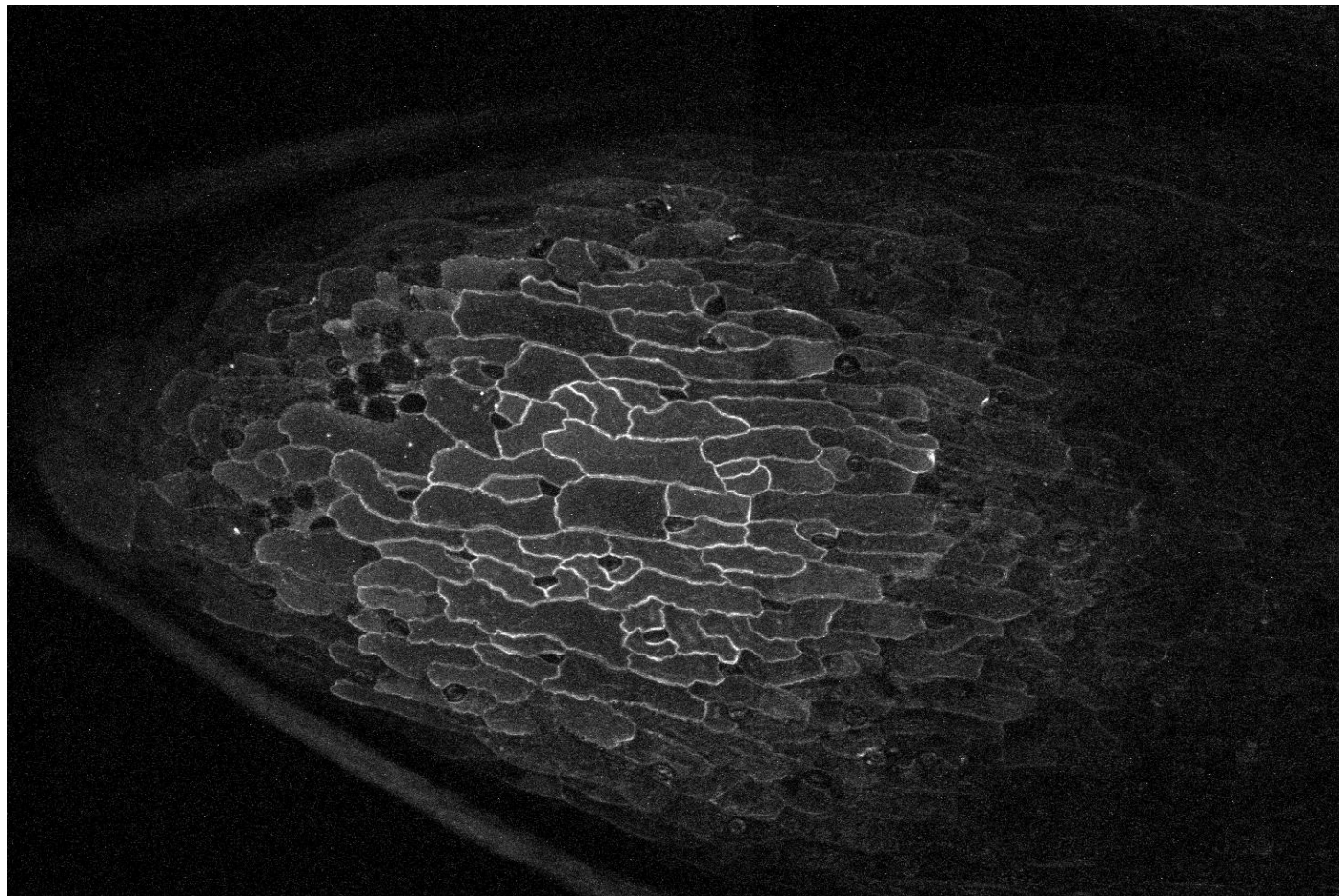
- Denoise input image
- If necessary, use high boost filtering to accentuate edges
- For sepals, compute Prewitt edges
- Then apply the following sequence of operations: threshold, majority filling (fill holes), thinning (generate one pixel wide lines), pruning (remove buds and open lines), cleaning (remove isolated single pixels)
- If results are OK, done! Otherwise, let user intervene to improve results

User intervention is *not on the original image*, but on the thick edges generated prior to thinning. User has to fill or create holes in a binary image. In our experience, **less than 3% of the pixels in the edges are manually edited to obtain good quality results.**



SEMSE^h and Mathematical Morphology

A typical confocal, maximum intensity projection image of a sepal:



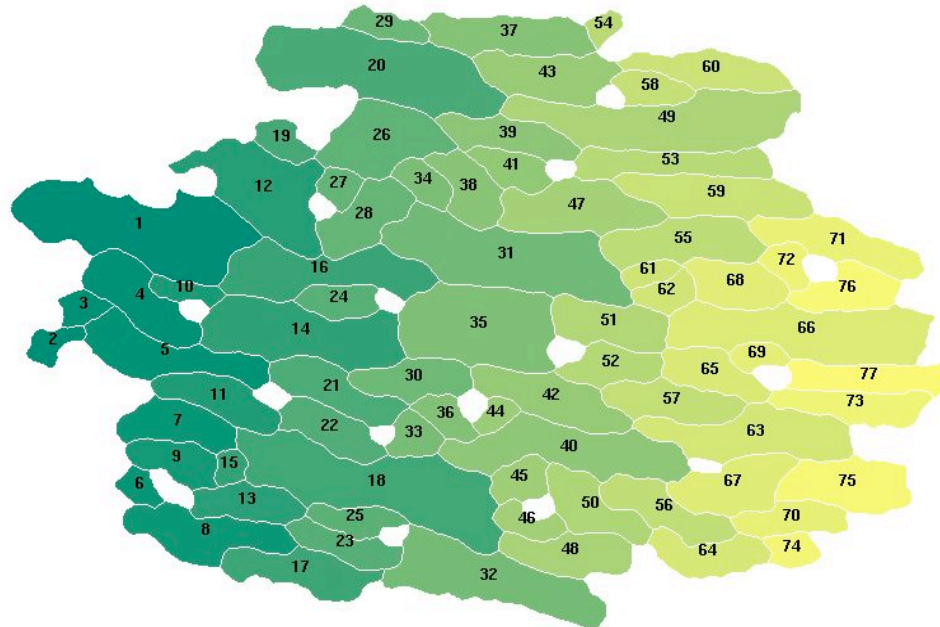
SEMSE^h and Mathematical Morphology

Segmentation of a sepal and the connectivity information of cells provided as a table (after running a connected components search):

```

0 35 : 1 2 3 4 5 6 7 8 9 11 12 17 19 20 26 29 32 37 48 49 50 53 54 59 60 63 64 66 70 71 73 74 75 76 77
1 5 : 0 4 10 12 16
2 3 : 0 3 5
3 4 : 0 2 4 5
4 6 : 0 1 3 5 10 14
5 7 : 0 2 3 4 11 14 21
6 3 : 0 8 9
7 5 : 0 9 11 15 18
8 5 : 0 6 13 17 23
9 5 : 0 6 7 13 15
10 4 : 1 4 14 16
11 5 : 0 5 7 18 22
12 7 : 0 1 16 19 26 27 28
13 6 : 8 9 15 18 23 25
14 8 : 4 5 10 16 21 24 30 35
15 4 : 7 9 13 18
16 8 : 1 10 12 14 24 28 31 35
17 4 : 0 8 23 32
18 12 : 7 11 13 15 22 25 32 33 40 45 46 48
19 3 : 0 12 26
20 7 : 0 26 29 37 39 43 49
21 6 : 5 14 22 30 33 36
22 4 : 11 18 21 33
23 5 : 8 13 17 25 32
24 2 : 14 16
25 3 : 13 18 23
26 9 : 0 12 19 20 27 28 34 38 39
27 3 : 12 26 28
28 6 : 12 16 26 27 31 34
29 3 : 0 20 37
30 5 : 14 21 35 36 42
31 10 : 16 28 34 35 38 47 51 55 61 62
32 5 : 0 17 18 23 48
33 5 : 18 21 22 36 40
34 4 : 26 28 31 38
35 7 : 14 16 30 31 42 51 52
36 5 : 21 30 33 40 44
37 5 : 0 20 29 43 54
38 6 : 26 31 34 39 41 47
39 6 : 20 26 38 41 49 53

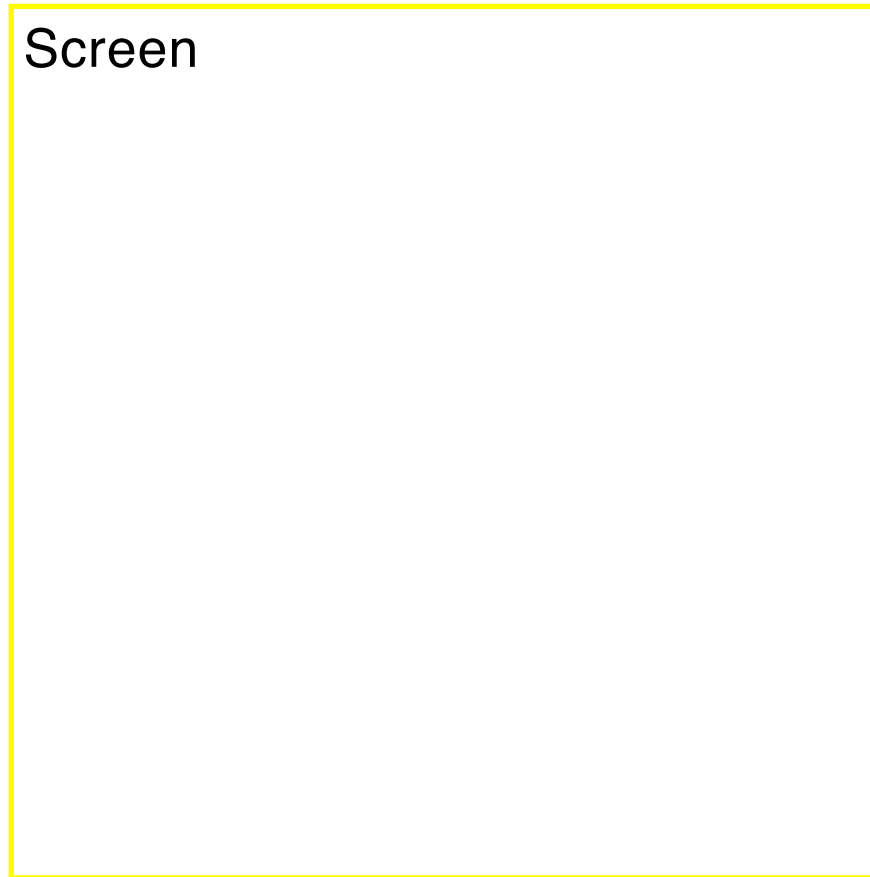
```



Denosing of meristem

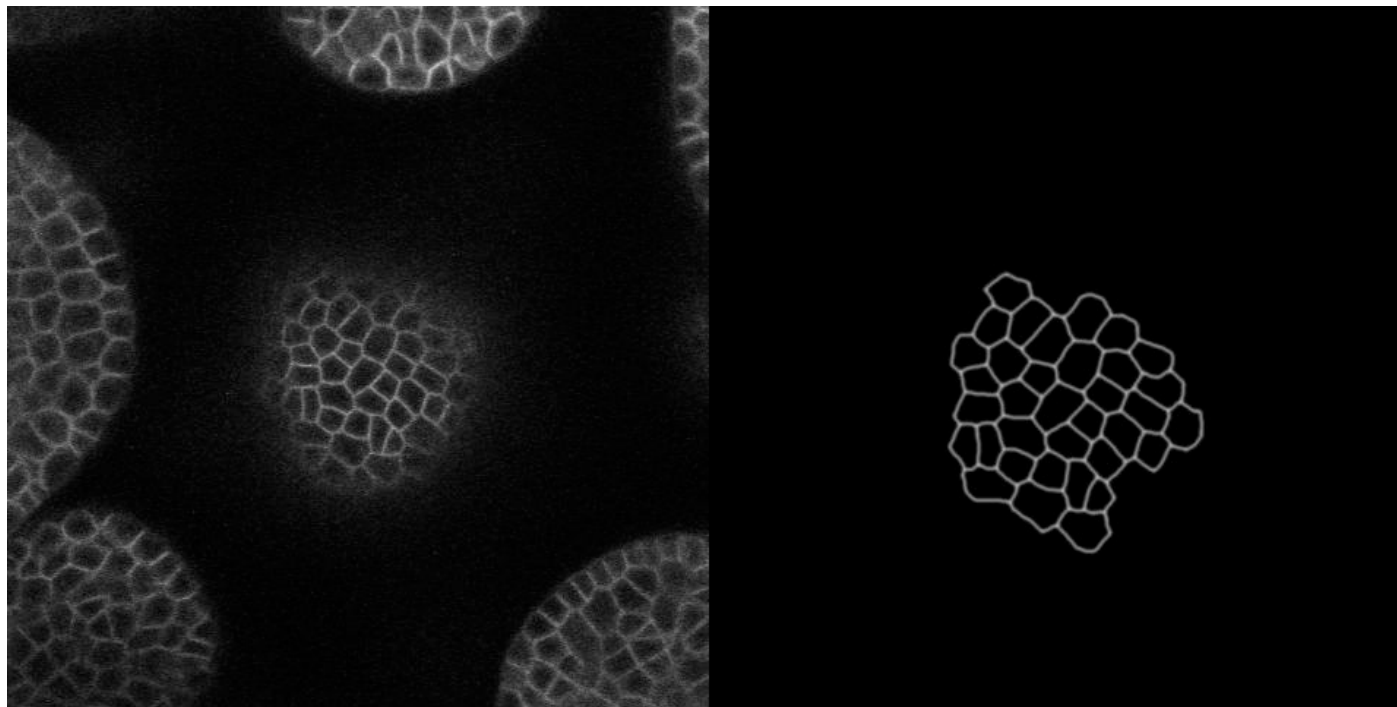
A confocal microscopy z-stack of a meristem; cell walls are shown in green and nuclei region in red:

Screen



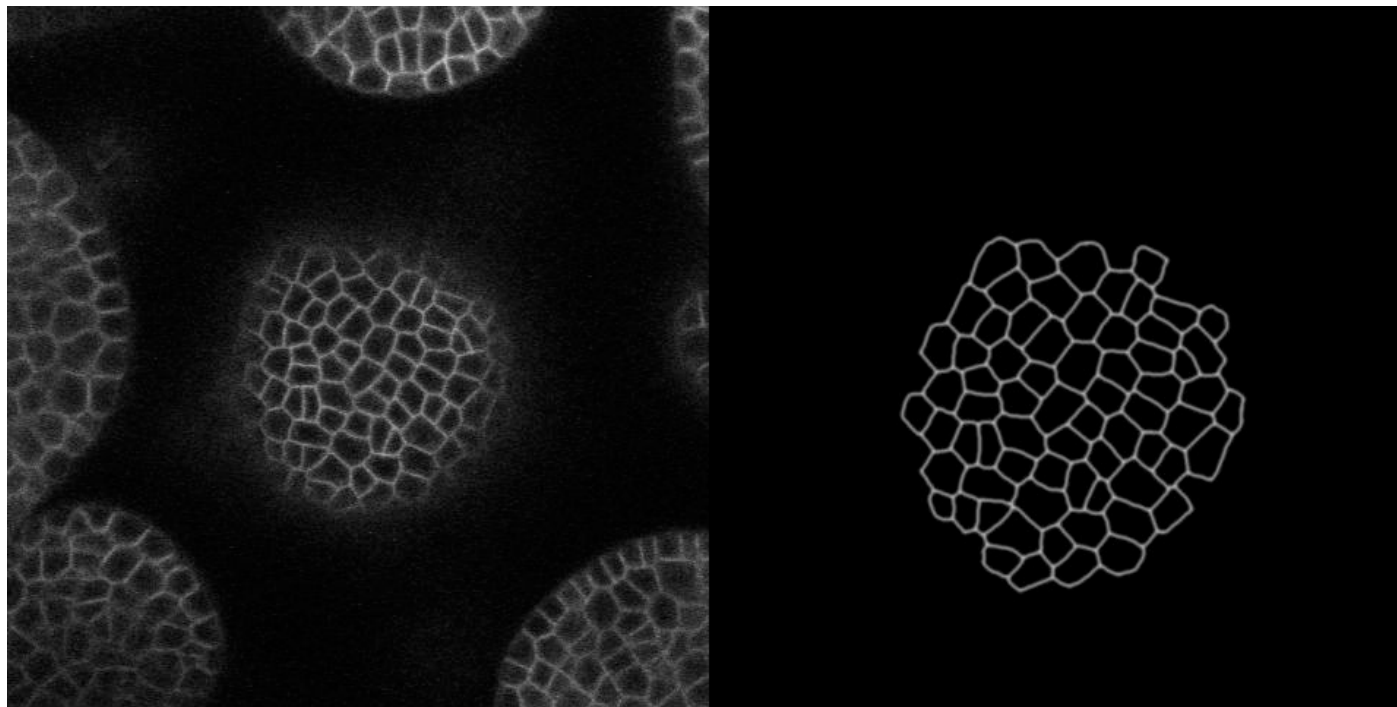
Segmentation of meristem layers

Segmenting slice #4 of cofocal image of meristem:



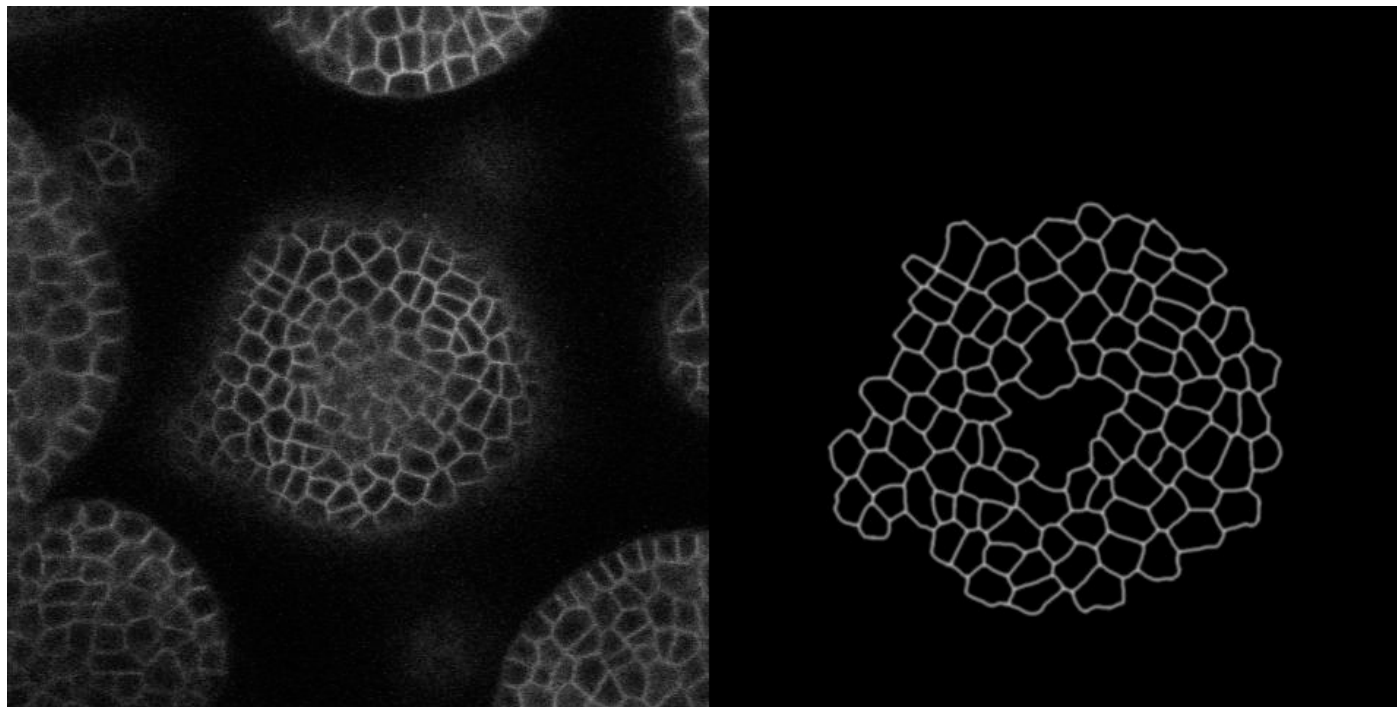
Segmentation of meristem layers

Segmenting slice #6 of cofocal image of meristem:



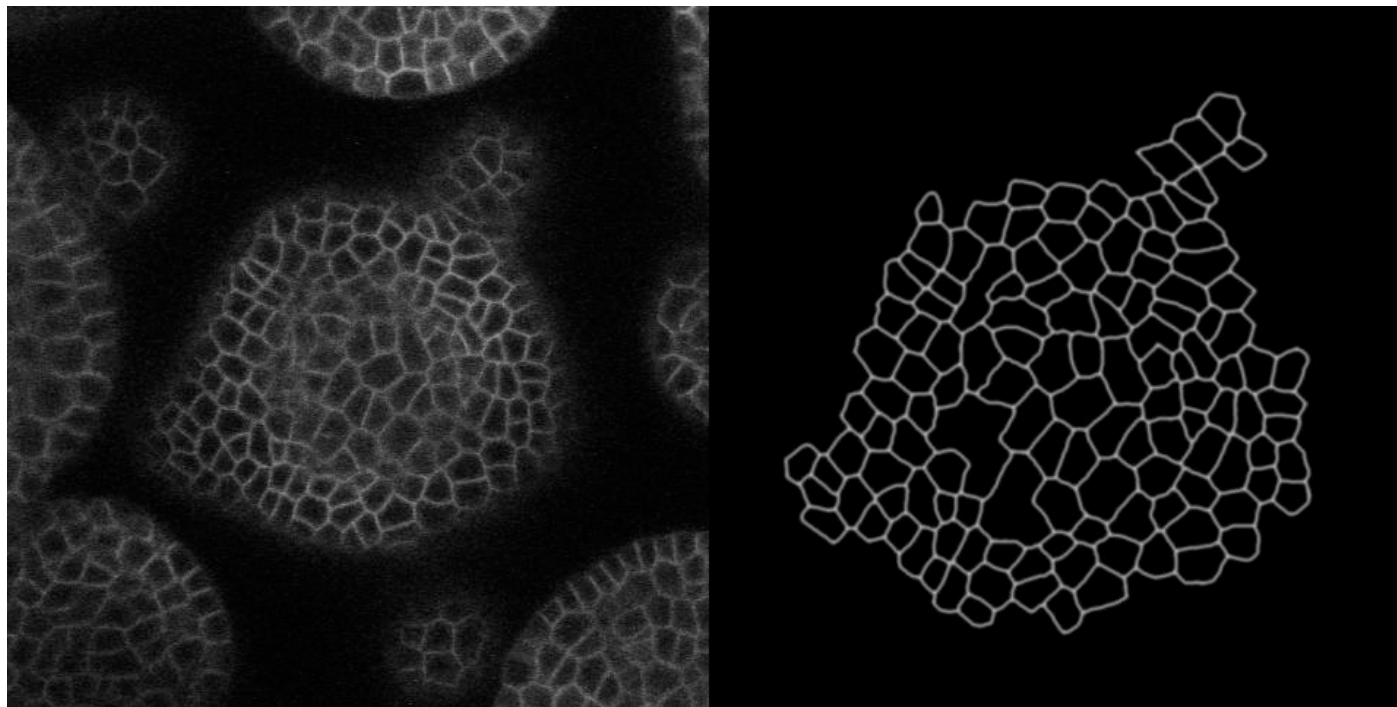
Segmentation of meristem layers

Segmenting slice #8 of cofocal image of meristem:



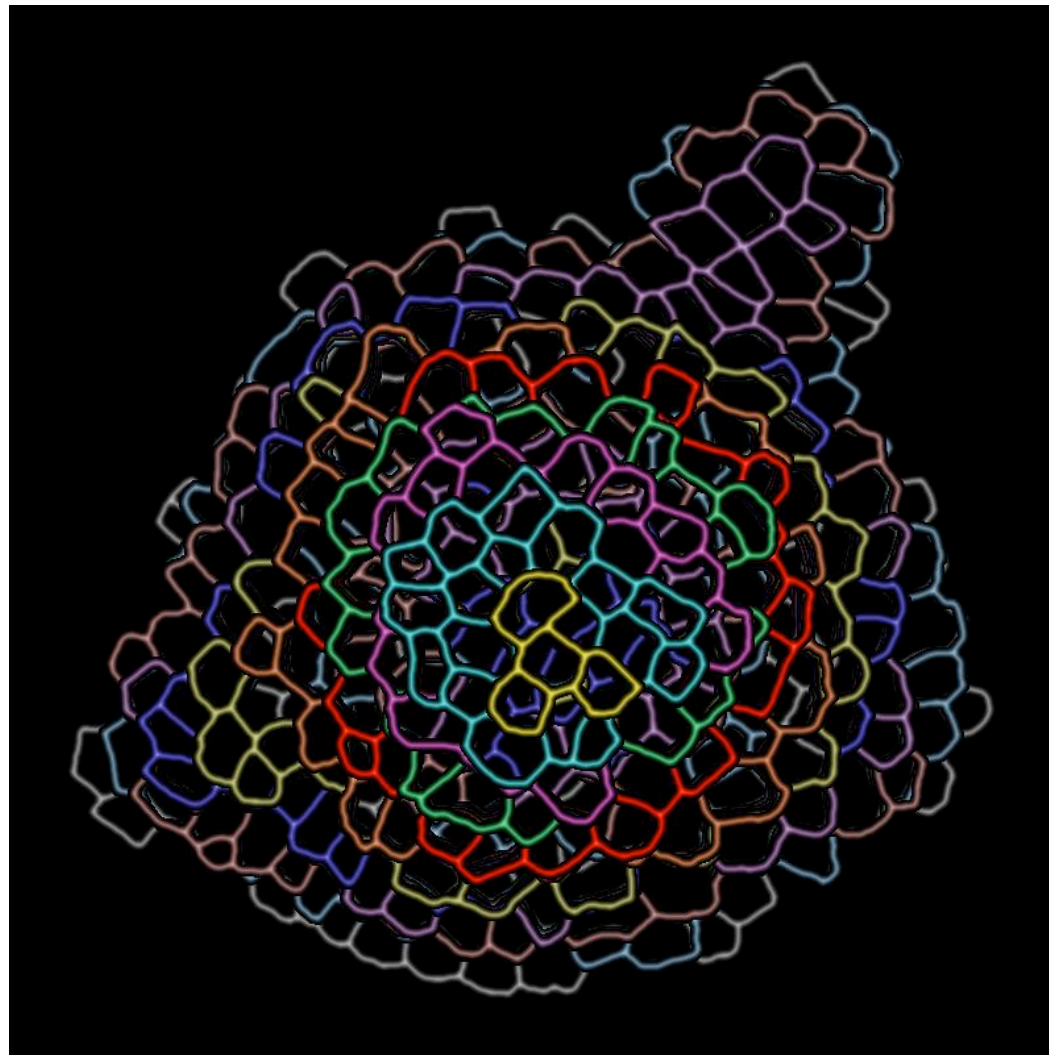
Segmentation of meristem layers

Segmenting slice #10 of cofocal image of meristem:



Segmentation of meristem layers

Superposition of all segmented slices:



Morphology - Junction detection

Junctions are singular points where 3 or more cell walls meet. They are crucial marks in the identification of the topology of non-manifold surface meshes created to represent cell walls.

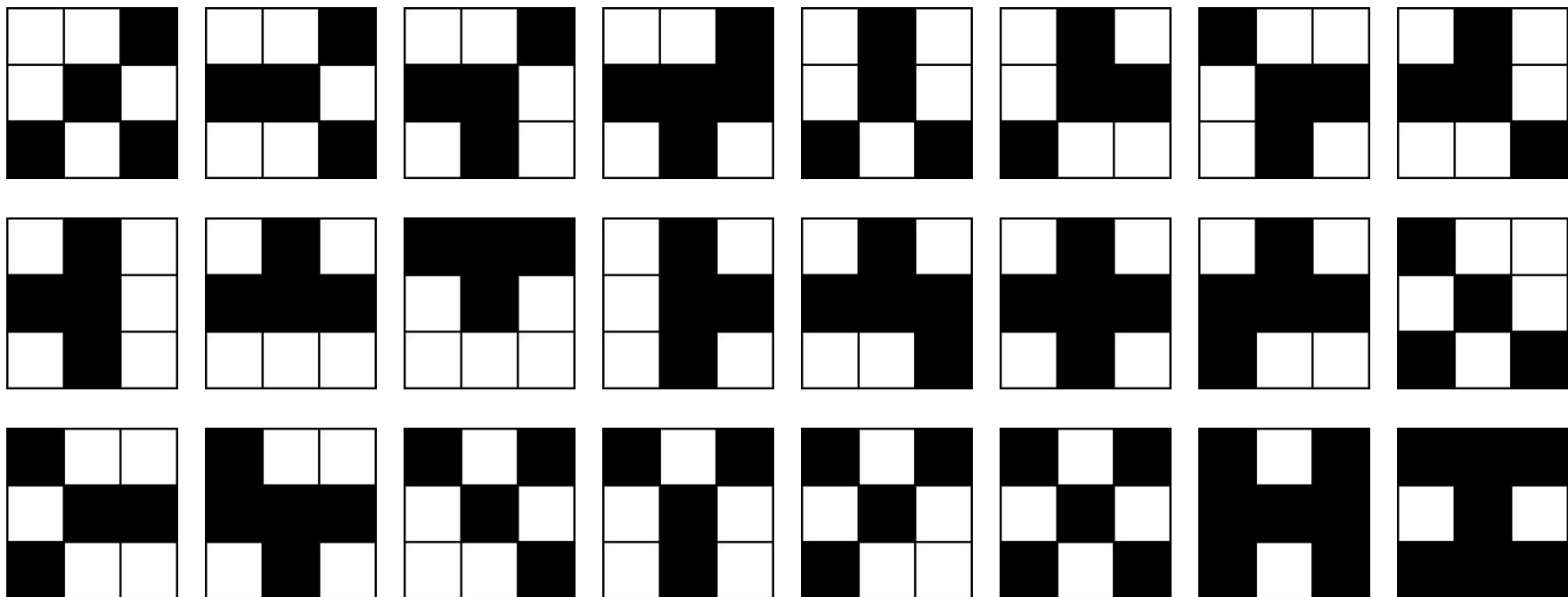
We use **template matching** to detect junctions in our *one pixel wide* segmented cell walls:

- Let $W = \{p | p \in \text{cell wall}\}$ be the set of pixels belonging to cell walls;
- Let $t_p = (d_0 d_1 \dots d_7) \quad \forall p \in W$ be the 3x3 neighborhood patch (minus p) centered around wall pixel p . Note that t_p is an ordered 8 bit word with binary entries $d_i \in \{0, 1\}_{i=0}^7$;
- Let $T = \{t_i | t_i = (b_0 b_1 \dots b_7)^i, i = 1 \dots 24, b_i \in \{0, 1\}\}$ be the set of binary words t_i representing the possible 3x3 *junction templates* (minus center pixel) encountered in one pixel wide lines (there are only 24 of them, $|T| = 24$);
- A wall pixel p is a junction iff $t_p \in T$



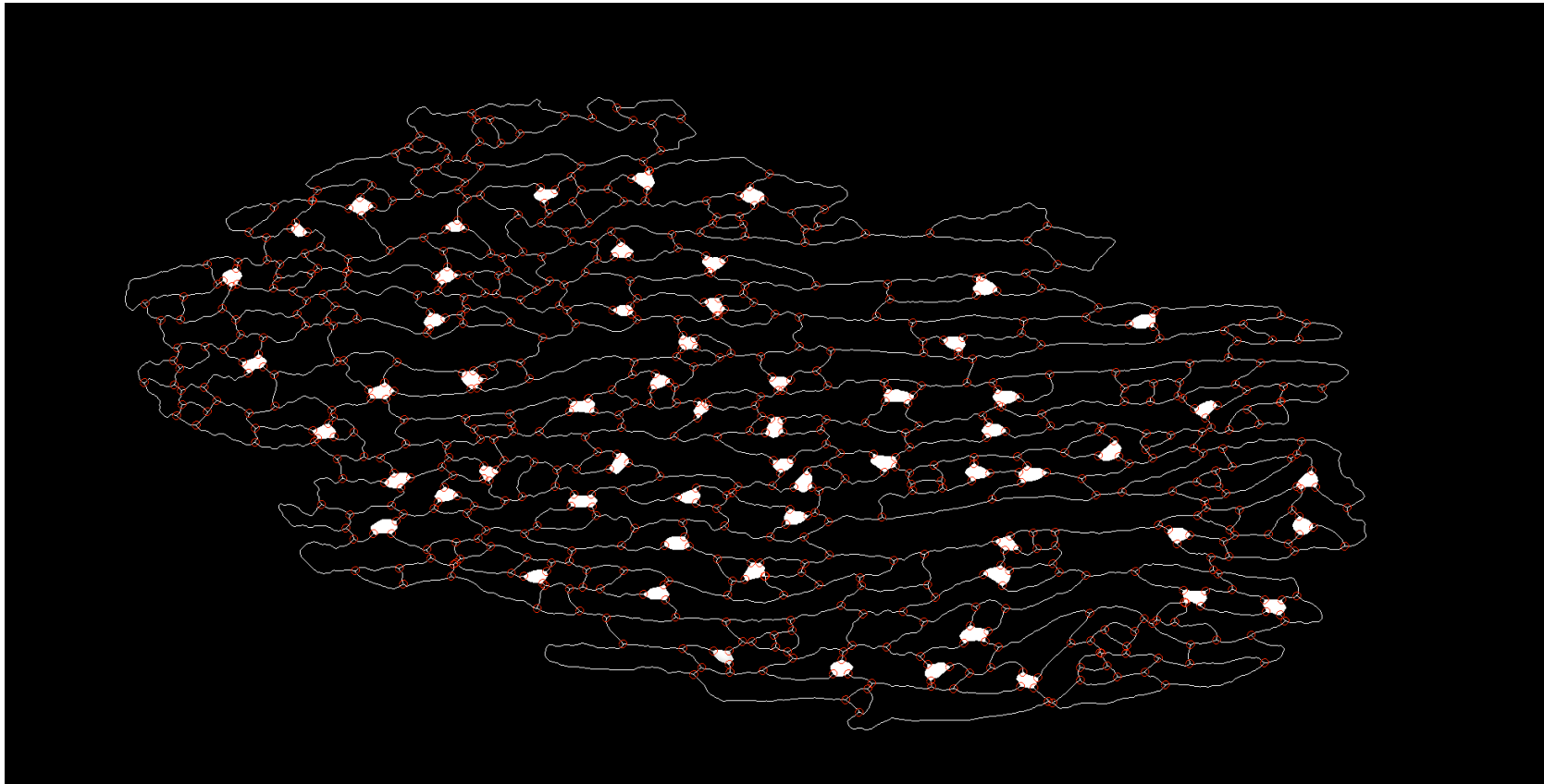
Morphology - Junction detection

The 24 junction templates, each represented by a 8 bit word (1 byte per template)



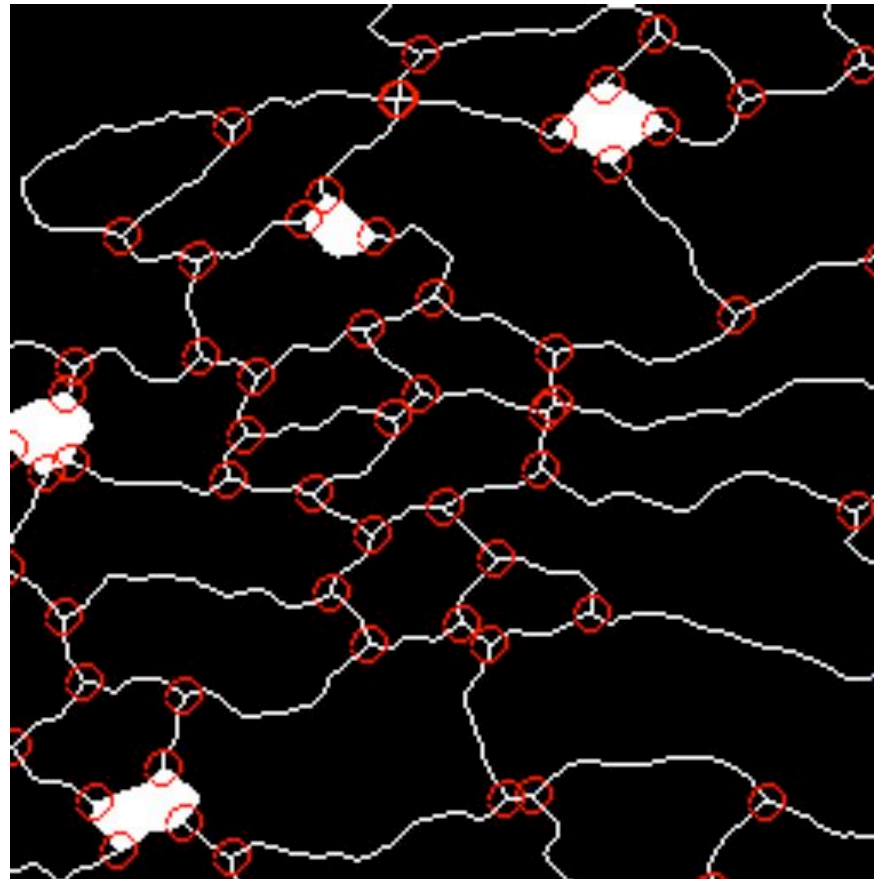
Morphology - Junction detection

Junctions are shown as centers of the red circles in the image



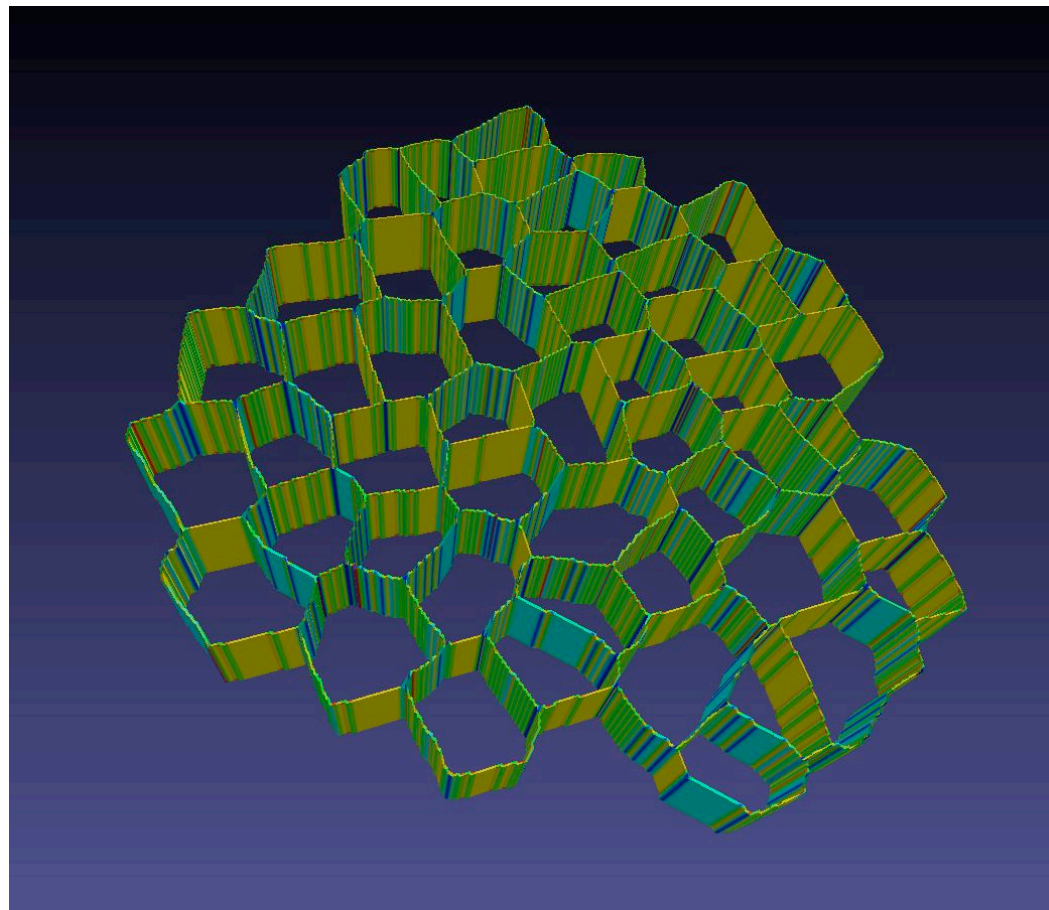
Morphology - Junction detection

Zooming in in the previous slide picture:



Non-manifold mesh

We stack a few copies of a same segmented slice to form a L1 layer of cells in the meristem. While this does not give the precise geometry it is a good approximation as compared to fully polygonal models describing the meristem.



The end

Thank you!

