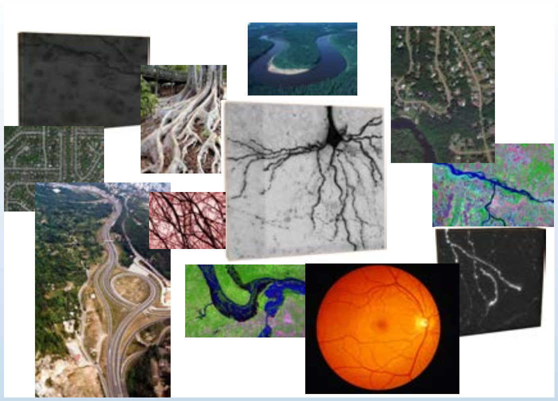
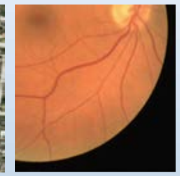
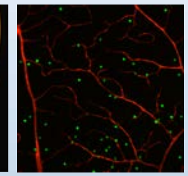
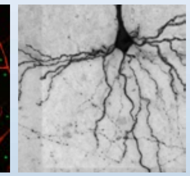
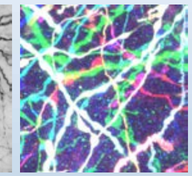


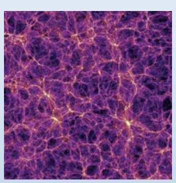
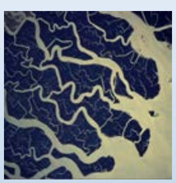
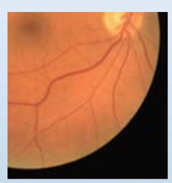
Using Machine Learning Techniques to Reconstruct Complex Curvilinear Structures

P. Fua
EPFL IC-CVLab

Complex Curvilinear Structures are Ubiquitous

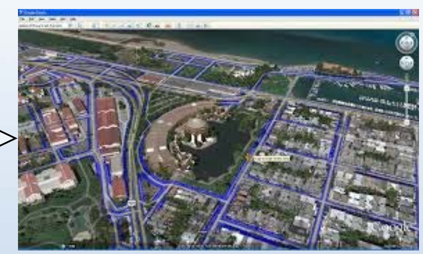
~ 500 m	~ 1 cm	~ 1 mm	~ 200 microns	~ 25 microns
				
Road Networks (AERIAL)	Retinal Vasculature (FUNDUS CAMERA)	Vessels and Neurons (CONFOCAL)	Neurons (BRIGHTFIELD)	Dendrites and Axons (BRAINBOW)

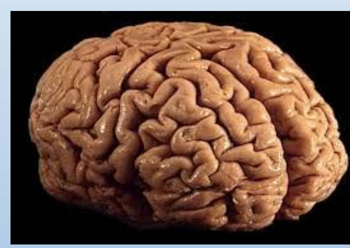
Complex Curvilinear Structures are Ubiquitous

~ 1e22 km !	~ 25 km		~ 500 m	~ 1 cm
		…		
Cosmic Web (SIMULATION)	Delta and Rivers (AERIAL)		Road Networks (AERIAL)	Retinal Vasculature (FUNDUS CAMERA)

… and often contain loops.

Google Earth ..

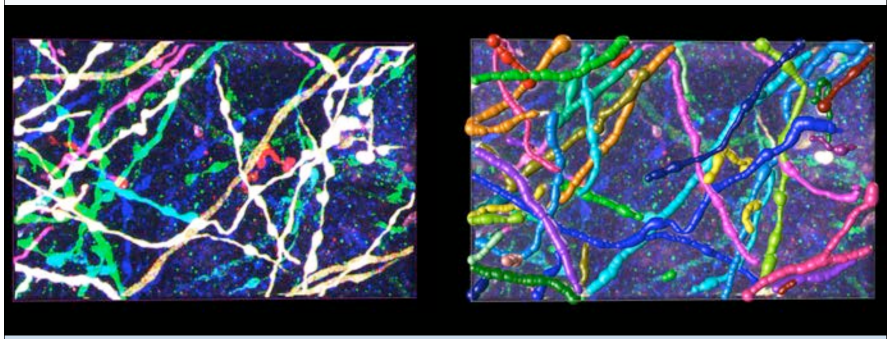




???

.. and Brain

Curse of Dimensionality

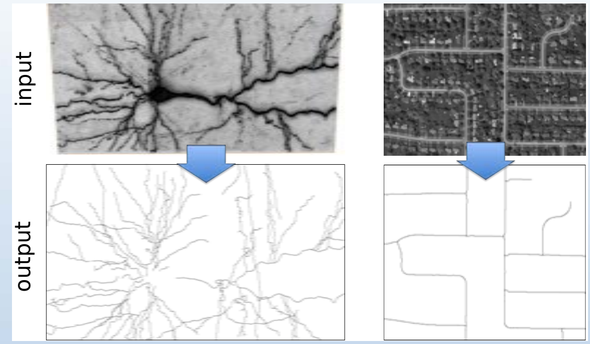


Brainbow 500x300x50 stack (Courtesy J. Lichtman) | Automated Reconstruction

- A 10 cm^3 cube or brain matter would produce a 100000x100000x100000 image at this resolution.
- Some hand tracing is possible but takes weeks and months.

—> Automation is required.

Delineation in 2D and 3D

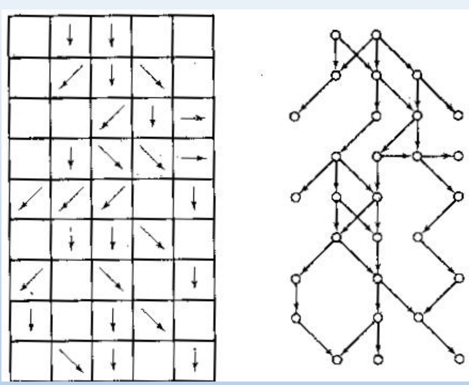


A very old topic in Computer Vision but full automation remains elusive:

- Images are very noisy.
- The structures are often complex.

Delineation in the 1980s

Image modeled as graph

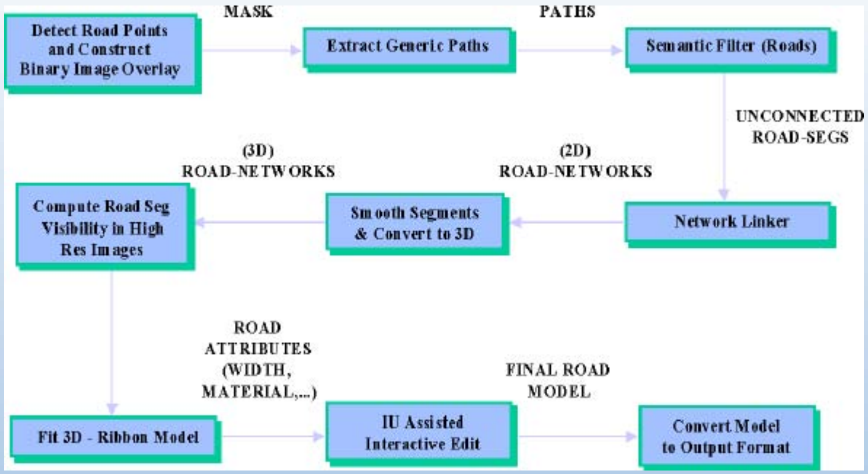


—> Minimun Spanning Tree can be computed in O(Nlog(N)).

Processing Steps

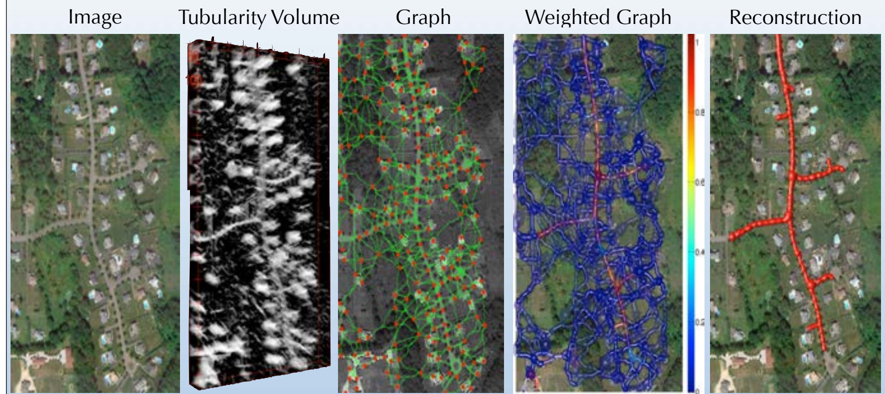
- Find centerlines.
- Build spanning tree.
- Retain road-like branches.
- Compute width.

Algorithm of Many Boxes



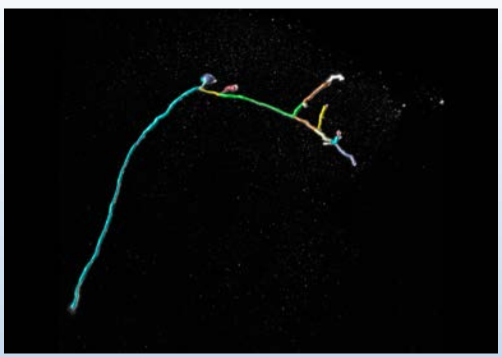
Fischler & Heller, 1998.

Algorithm Revisited (2D)



Turtetken, CVPR'13

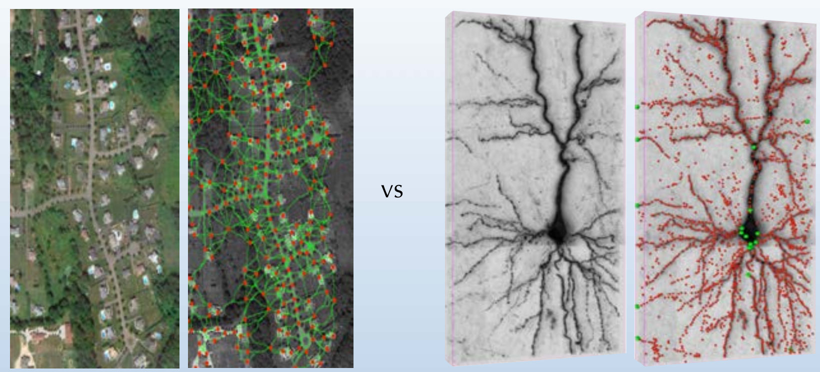
Algorithm Revisited (3D)



Processing steps:

1. Compute the tubularity of individual voxels.
2. Select voxels that locally maximize this measure.
3. Build a *tubular* graph by linking these voxels.
4. Find an optimal subgraph within the graph.

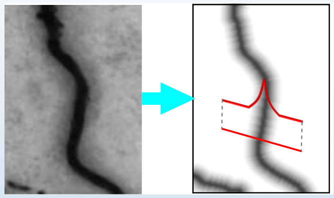
Machine Learning to the Rescue



- What does it mean to be "tubular"?
- How can you tell a good path from bad one?
- How do you design an algorithm that works in many different cases?

—> Learn the appearance from a suitable training database!

Finding the Centerlines



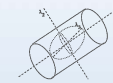
An ideal tubularity measures should:

- Have a strong response at the centre of the ribbons or tubes.
- This response should be maximal at the centre and decrease away from it so that non-maximum suppression works well.
- There should ***no*** response at the tube boundaries to avoid confusing the algorithm further down the processing chain.

Hand-Designed Tubularity Measures

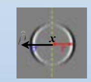
- Hessian Based Approaches:
  - Frangi 98
  - Sato 98

$$\nabla^2 I(\mathbf{x}) = \begin{bmatrix} I_{xx}(\mathbf{x}) & I_{xy}(\mathbf{x}) & I_{xz}(\mathbf{x}) \\ I_{yx}(\mathbf{x}) & I_{yy}(\mathbf{x}) & I_{yz}(\mathbf{x}) \\ I_{zx}(\mathbf{x}) & I_{zy}(\mathbf{x}) & I_{zz}(\mathbf{x}) \end{bmatrix}$$

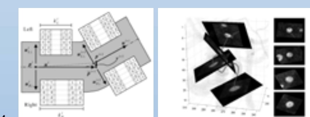


- Optimized Steerable Filters:
  - Mejering 04
  - Jacobs 04
  - Aguet 05
  - German 09

- Oriented Flux Based Approaches:
  - Law 08, 10
  - Turetken 13

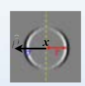


- Deformable Templates:
  - Al-Kofahi 03
  - Tyrrell 07

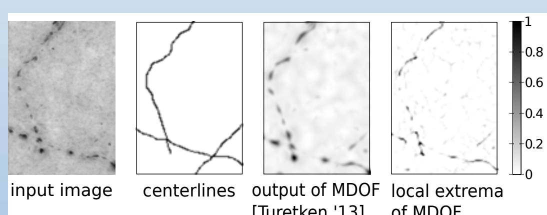


Oriented Flux

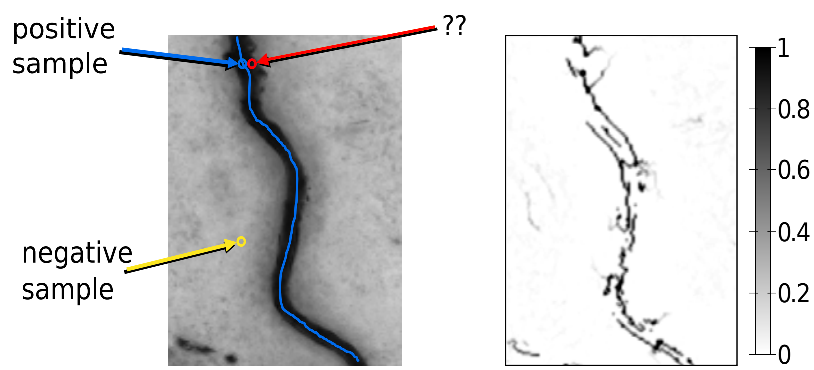
$$f(\boldsymbol{x}; r, \hat{\rho}) = \frac{1}{4\pi r^2} \int_{\partial B_r} \Big((\boldsymbol{v}(\boldsymbol{x} + \boldsymbol{A}) \cdot \hat{\rho})\hat{\rho} \Big) \cdot \hat{n}\, dA \quad s(\boldsymbol{x}; r, \hat{\rho}) = \frac{1}{4\pi r^2} \int_{\partial B_r} \Big(\boldsymbol{v}(\boldsymbol{x} + \boldsymbol{A}) \cdot \hat{\rho} \Big)\, dA$$



- Can be computed by convolving the images with appropriately designed filters.
- Works very well when the tubes are relatively regular but performance decreases as the irregularity increases.



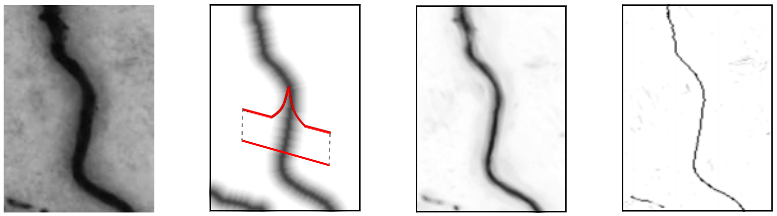
Centerline Detection as a Classification Problem



Input image Non-maxima suppression

- Classification is not the right approach to deal with pixels near the centerline, i.e. hard negatives.
- What other formalism could we use?

Centerline Detection as a **Regression** Problem



Input image | Desired regressor output | Actual output | After non-maxima suppression

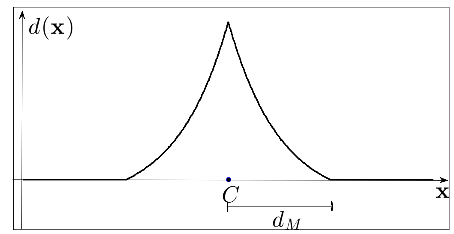
- Regression guarantees that non-maximum suppression will do the right thing.

Sironi, CVPR'14

Ideal Response

- C = set of centerline pixels
- $D_C(\mathbf{x})$ = distance transform to centerlines
- Learn $\mathbf{d}(\mathbf{x})$ = exponential transformation of $D_C(\mathbf{x})$:

$$\mathbf{d}(\mathbf{x}) = \begin{cases} e^{a\left(1-\frac{D_C(\mathbf{x})}{d_M}\right)} - 1 & \text{if } D_C(\mathbf{x}) < d_M \\ 0 & \text{otherwise,} \end{cases}$$



The function we want to learn (1D example)

Gradient Boost

- Learn $\varphi(f(\mathbf{x})) \approx \mathbf{d}(\mathbf{x})$ using GradientBoost:
  - Given feature vector $f_i = f(\mathbf{x}_i)$ and $y_i = \mathbf{d}(\mathbf{x}_i)$
  - Iteratively build

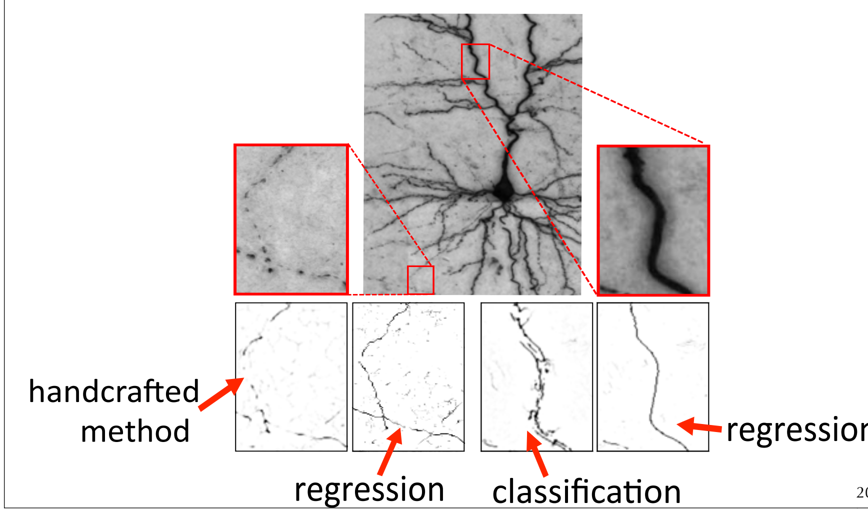
$$\varphi(f(\mathbf{x})) = \sum_{k=1}^{K} \alpha_k h_k(f(\mathbf{x}))$$

  - h_k Regression Trees
  - Minimize Squared Loss

$$\mathcal{L} = \sum_i \|y_i - \varphi(f_i))\|_2^2$$

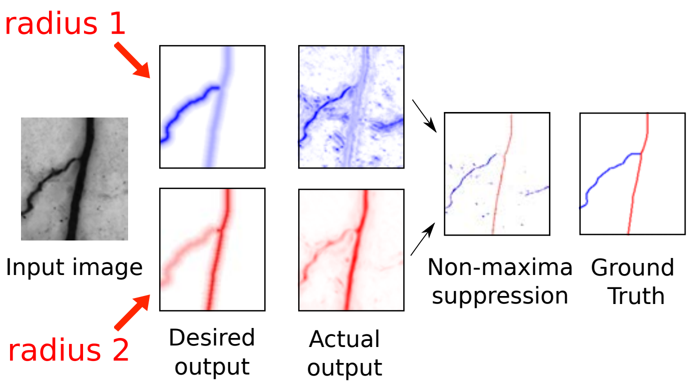
Single Scale Results

- More accurate than classification methods

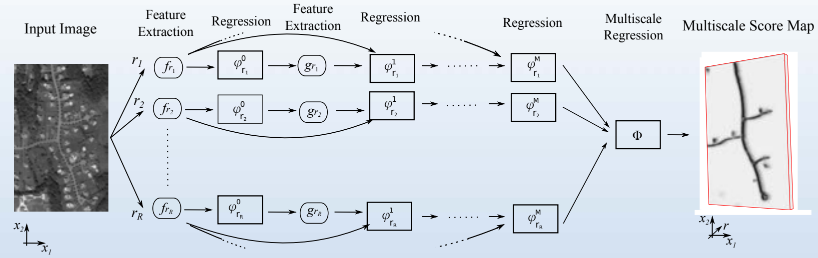


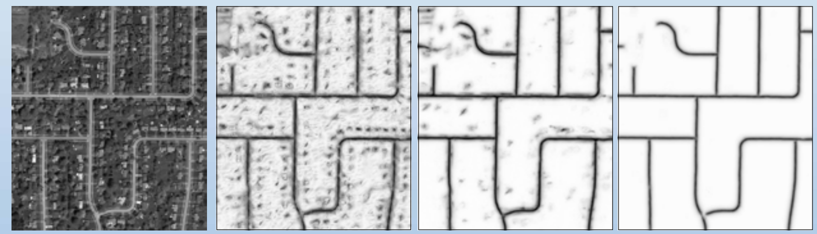
Multi Scale Results

- Multiscale detection
  ⇒ We also estimate tubular structures radius



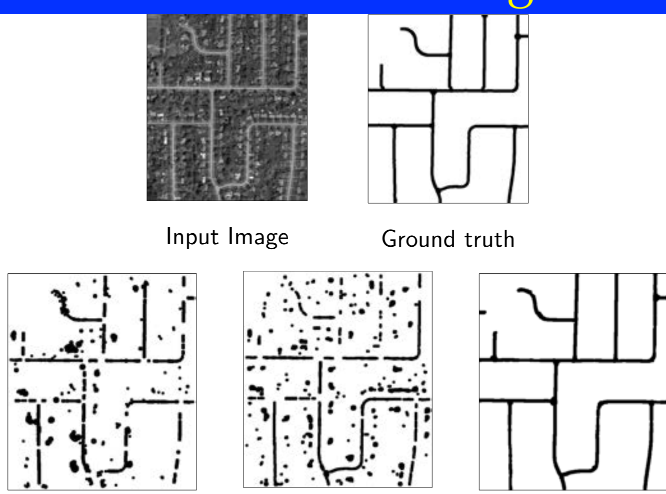
Recursive Approach





(a) Input Image I (b) Score map $\varphi^{(0)}$ (c) Score map $\varphi^{(M)}$ (d) Final approximation Φ

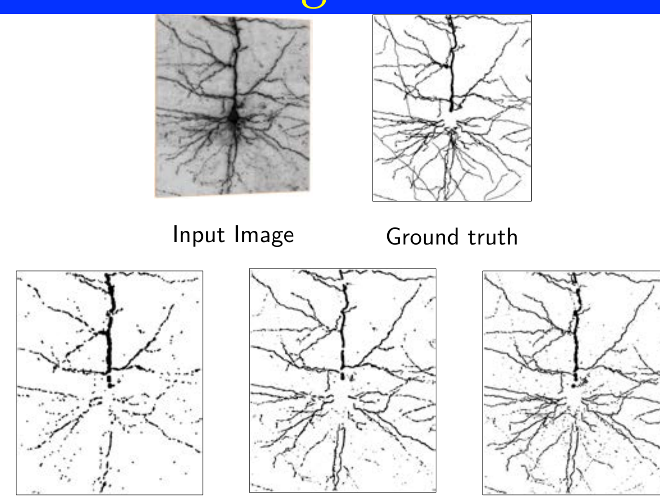
Comparing Performance on an Aerial Image



Input Image Ground truth

Classification MDOF [Turetken '13] Regression

Comparing Performance on an Brighfield Stack



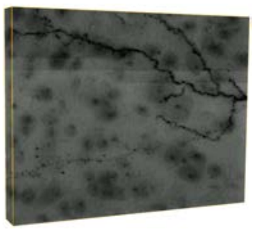
Input Image Ground truth

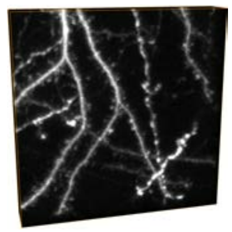
Classification MDOF [Turetken '13] Our method

Test Datasets


Aerial


Brightfield

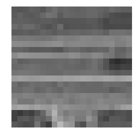

VC6

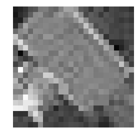

Vivo2P

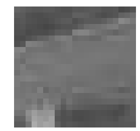
Aerial Images

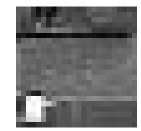
- Sample Aerial images:

- Sample patches:


Road


Not a road

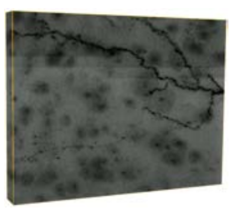

Road

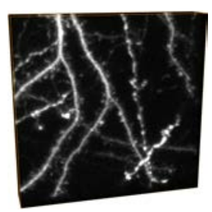

Not a road

Microscopy Images

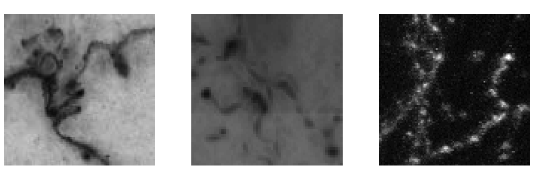
- Sample image stacks:


Brightfield

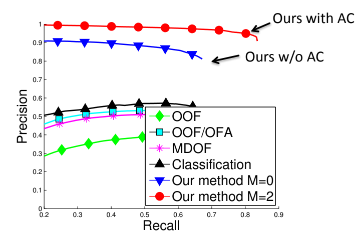

VC6


Vivo2P

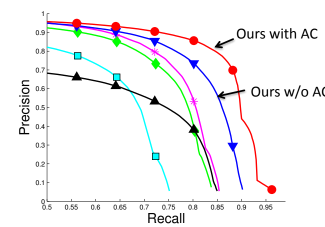
- Sample patches:



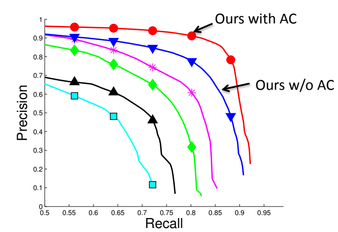
Precision Recall Curves



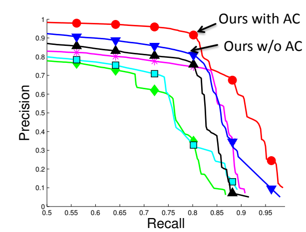
Aerial



Brightfield

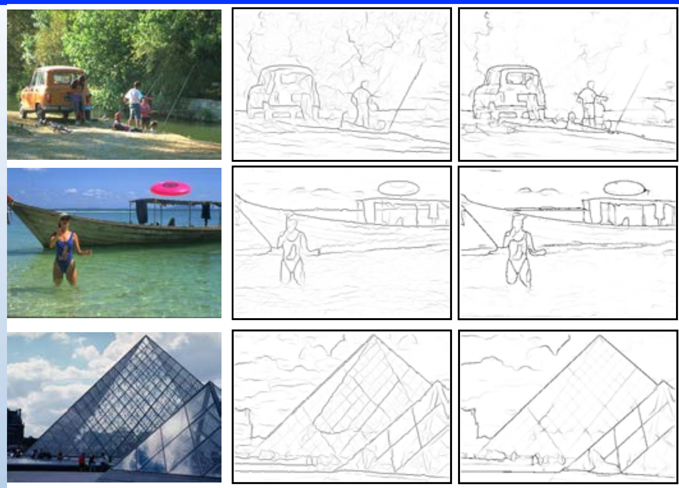


VC6



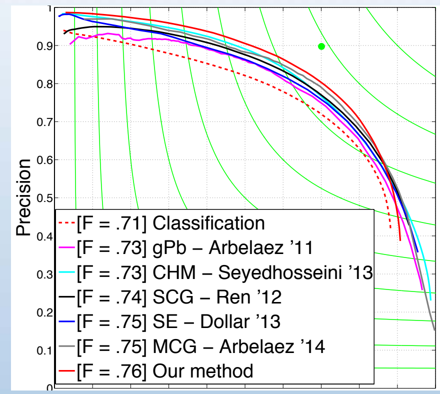
Vivo2P

Regression for Contour Detection



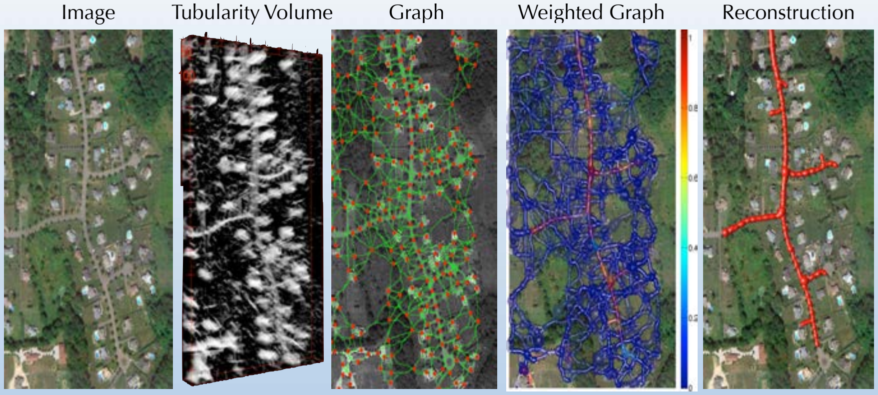
BSDS500 Dataset | Ren, NIPS'12 Classification | Our approach Regression

Regression for Contour Detection



Back to Delineation

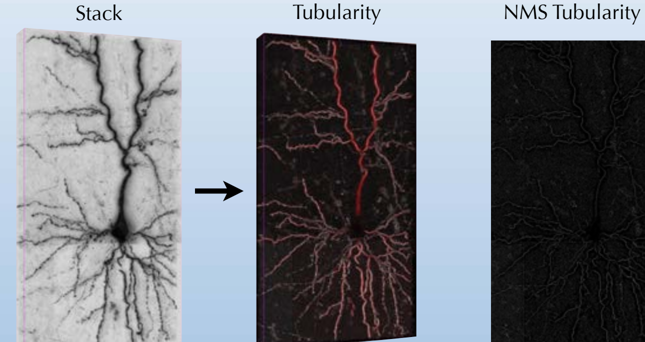
Image | Tubularity Volume | Graph | Weighted Graph | Reconstruction



Building the Graph

- Compute tubularity.
- Canny like Non maximum suppression.

Stack | Tubularity | NMS Tubularity

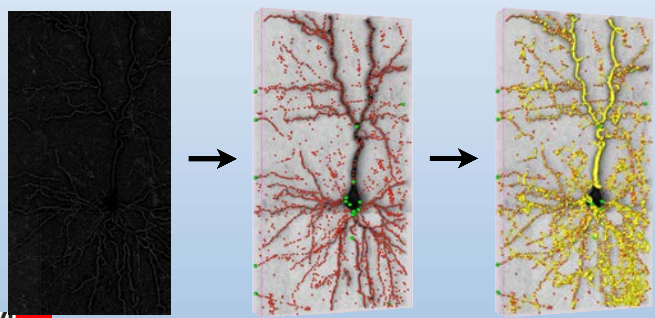


Constructing the Tubular Graph

- Vertices: Local-maxima sampled iteratively at regular intervals.
- Edges: Tubular paths linking pairs of samples through high-tubularity voxels.

$$\gamma^*(v_i, v_j) = \underset{\gamma}{\operatorname{argmin}} \int_0^1 \mathcal{P}(\gamma(s))\, d(s)$$

NMS Tubularity → Samples → Graph



Giving Weights to the Edges

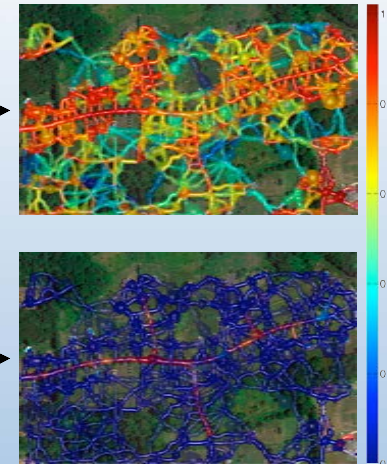
- Standard Approach: Integrate tubularity values along paths.

$$\mathcal{E}(\gamma) = \int_\gamma \mathcal{P}(\gamma)\mathrm{d}\gamma$$

Uninformative for paths that partially right and partially wrong.

- Better Approach: Learn a quality function based on global appearance and geometry.

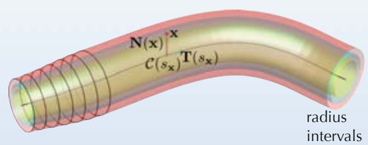
Provides much more informative scores!



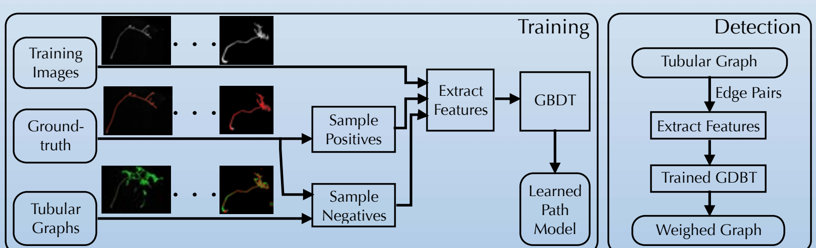
Path Classification

- Features:
  - Appearance: Histogram of gradient deviations (HGD) from path's principle axis:

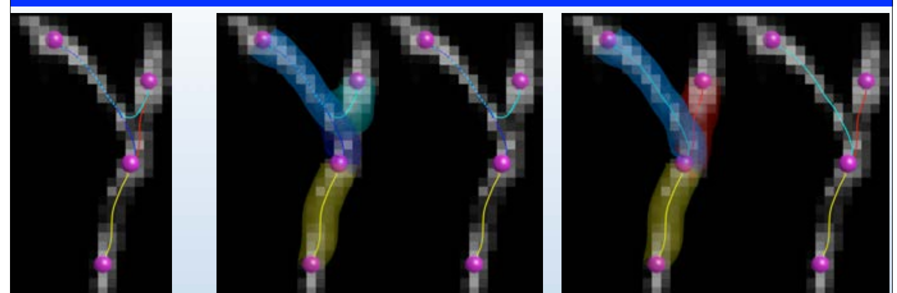
    $\Psi(\mathbf{x}) = \operatorname{angle}(\nabla I(\mathbf{x}), \mathbf{N}(\mathbf{x})), \quad \text{if } \|\mathbf{x} - \mathcal{C}(s_\mathbf{x})\| > \varepsilon$
  - Geometry: Max centerline curvature, tortuosity along xy and radius dimensions.



- Pipeline:



Finding the Best Tree



Tubularity graph | Without edge pair term | With edge pair term

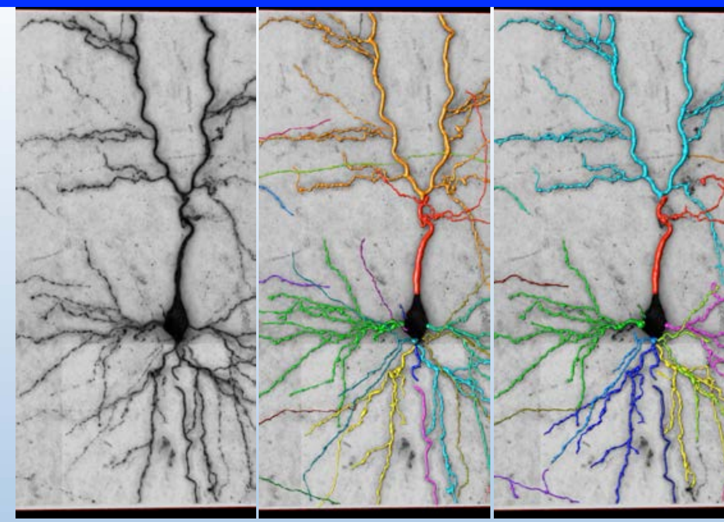
$$
\begin{aligned}
\mathbf{t}^* &= \underset{\mathbf{t}\in\mathcal{T}(G)}{\operatorname{argmax}} P(\mathbf{T}=\mathbf{t}|I)\ , \\
&= \underset{\mathbf{t}\in\mathcal{T}(G)}{\operatorname{argmax}} P(I|\mathbf{T}=\mathbf{t})P(\mathbf{T}=\mathbf{t})\ , \\
&= \underset{\mathbf{t}\in\mathcal{T}(G)}{\operatorname{argmin}} \sum_{e_{ij}\in G} c^d_{ij} t_{ij}
\end{aligned}
$$

QMIP Formulation

$$
\begin{aligned}
\min \quad & \sum_{e_{ij}\in E, e_{jk}\in E} c_{ijk}\, t_{ij}\, t_{jk} \\
\text{s.t.} \quad & \sum_{v_j\in V\setminus\{v_r\}} y^l_{rj} \le 1, \qquad \forall v_l \in V\setminus\{v_r\}, \\
& \sum_{v_j\in V\setminus\{v_k\}} y^l_{jk} \le 1, \qquad \forall v_l \in V\setminus\{v_r\}, \\
& \sum_{v_j\in V\setminus\{v_i,v_r\}} y^l_{ij} - \sum_{v_j\in V\setminus\{v_i,v_l\}} y^l_{ji} = 1, \quad \begin{array}{l}\forall v_k \in V\setminus\{v_r\},\\ \forall v_i \in V\setminus\{v_r, v_k\},\end{array} \\
& y^l_{ij} \le t_{ij}, \qquad \forall e_{ij}\in E,\ v_l \in V\setminus\{v_r, v_i, v_j\}, \\
& y^l_{il} = t_{il}, \qquad \forall e_{il}\in E, \\
& y^l_{ij} \ge 0, \qquad \forall e_{ij}\in E,\ v_l \in V\setminus\{v_r, v_i\}, \\
& t_{ij} \in \{0,1\}, \quad \forall e_{ij}\in E.
\end{aligned}
$$

given the root note v_r.

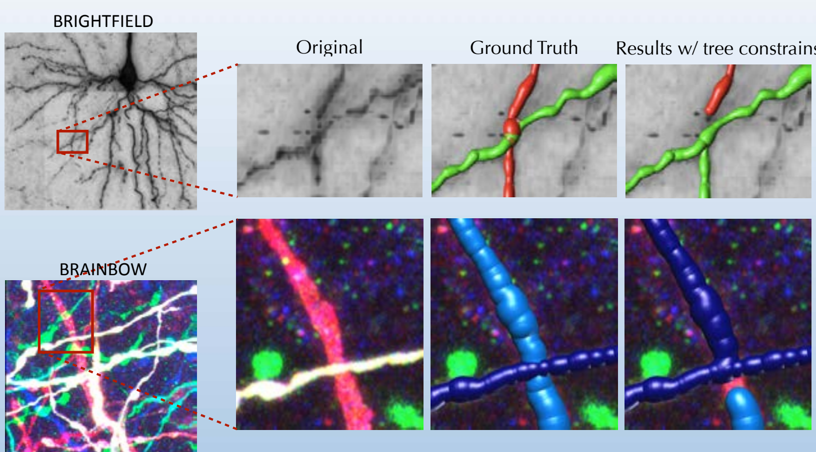
Brightfield



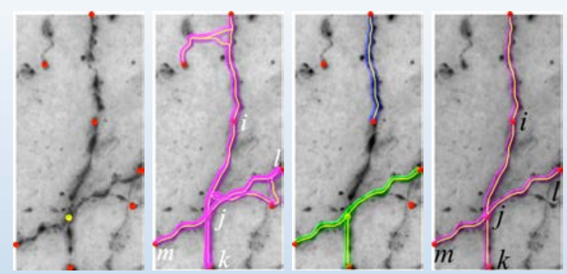
Brightfield Stack | Ground Truth | QMIP reconstruction

Turetken, CVPR'12

Problems with Tree Constraints



From Trees to Subgraphs

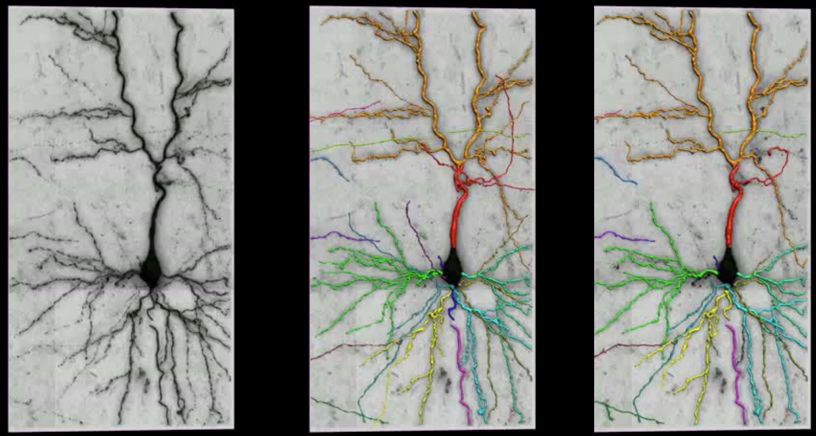


Even in true tree structures, some vertices might have to be shared to avoid topological errors:

- Allow loops
- Penalize hanging branches.

➡Still a QMIP problem.

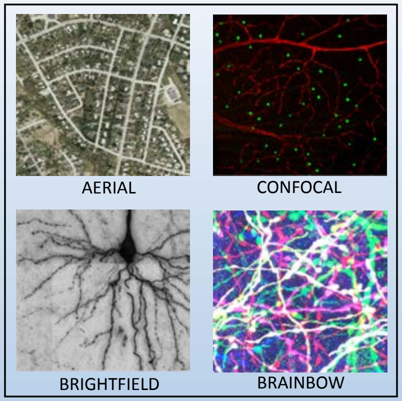
Loopy Brightfield



Brightfield Stack | Ground Truth | QMIP reconstruction

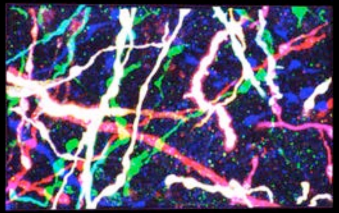
Test Datasets

- **Aerial** Roads: 21 training and 8 test images.
- **Confocal** Blood Vessels: 1 partially traced image stack for training and 2 for testing. (Maximilian Joesch, Markus Meister - Harvard)
- **Brightfield** Neurons: 1 training and 3 test image stacks. (Felix Schürmann, Henry Markram - EPFL)
- **Brainbow** Neurites: 3 training and 3 test image stacks. (Luke Bogart, Takao Hensch, Jeff Lichtman - Harvard)
- 80000 positive and negative samples for each dataset.
- Same path classifier parameters for all four datasets.



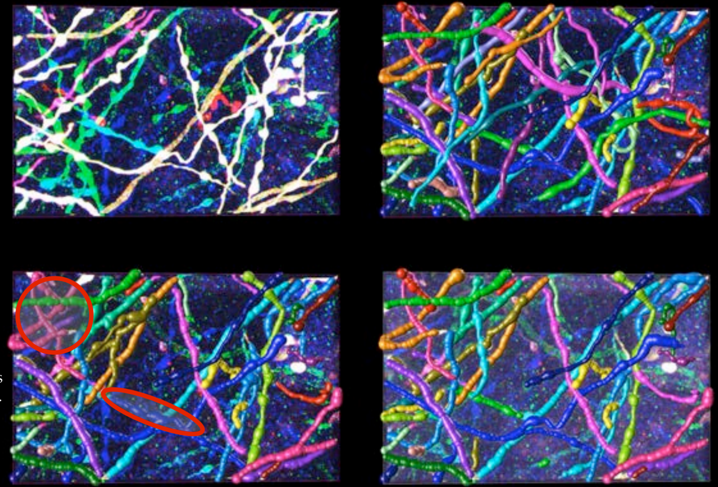
Results: Brainbow

Stack



Results: Brainbow

Stack



Ground Truth

Loopless Reconst.

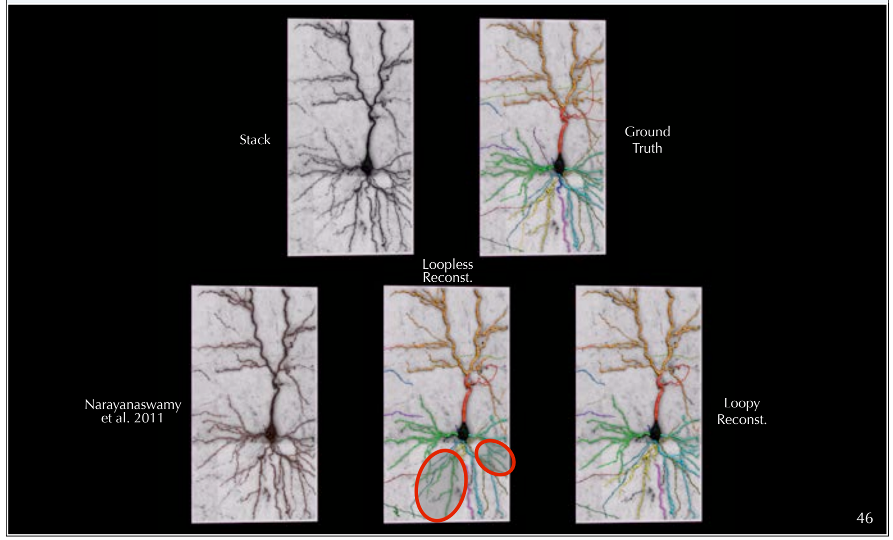
Loopy Reconst.

Quantitative Quality Measures For Trees

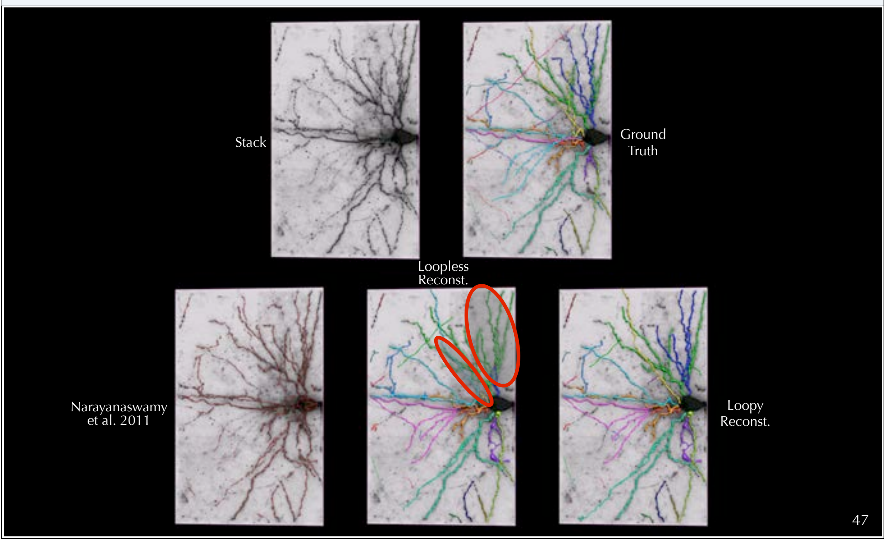
- DIADEM Measure (Topological Quality):

	BRBW1	BRBW2	BRBW3
Loopless Reconst.	0.3692	0.5118	0.4016
Loopy Reconst.	**0.8327**	**0.6897**	**0.7848**

Results: Brighfield



Results: Brighfield



Quantitative Quality Measures For Trees

- DIADEM Measure (Topological):

	BRF1	BRF2	BRF3
Loopless Reconst.	0.6114	0.4263	0.6551
Loopy Reconst.	**0.7282**	**0.5122**	**0.7391**

- NETMETS Measure (Geometric & Connectivity):

	BRF1				BRF2				BRF3			
Loopy-IP	**0.05**	**0.29**	**0.71**	**0.65**	**0.11**	**0.29**	**0.81**	**0.78**	**0.07**	**0.28**	0.77	**0.70**
k-MST	0.10	0.44	0.79	0.88	**0.11**	0.53	0.84	0.91	0.13	0.35	0.81	0.92
Focus	0.39	0.54	0.75	1.00	0.49	0.53	0.90	1.00	0.38	0.46	**0.74**	1.00
OSnake	0.66	0.63	0.98	0.99	0.66	0.59	0.99	1.00	0.69	0.38	0.95	0.99
APP2	0.68	0.64	1.00	1.00	0.63	0.54	1.00	1.00	0.65	0.49	1.00	1.00

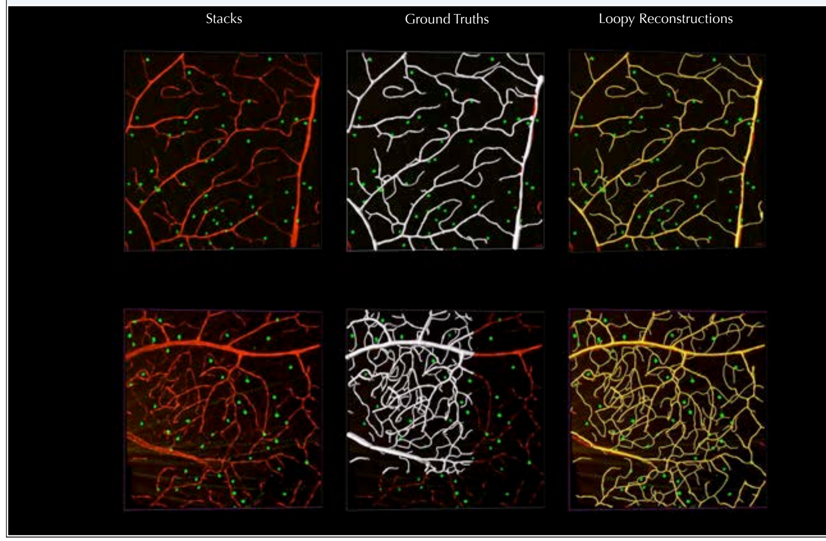
Geometric FPR **Geometric FNR** **Connectivity FPR** **Connectivity FNR**

Runtimes

	BRBW1	BRBW2	BRBW3	BRF1	BRF2	BRF3	CONV1	CONV2
Network Flow	166.7	132.9	291.1	379.3	17.2	**4.9**	8.4	106.8
Subset + Lazy	**32.9**	**4.9**	**19.5**	**36.6**	**10.9**	7.7	**0.6**	**4.2**
# Lazy Constr.	1008	892	1092	1556	1165	2259	972	5175

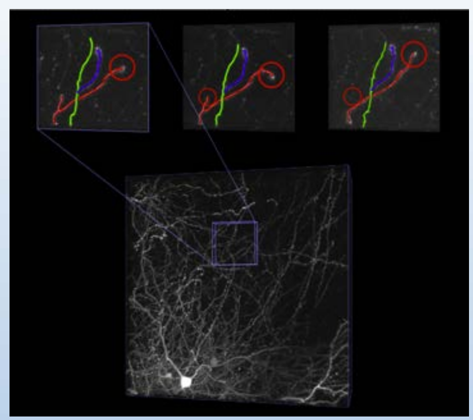
Times in minutes required to solve the QMIP, which is the most time-consuming part of the approach

Results: Confocal



Results: Roads

Evolving Structures



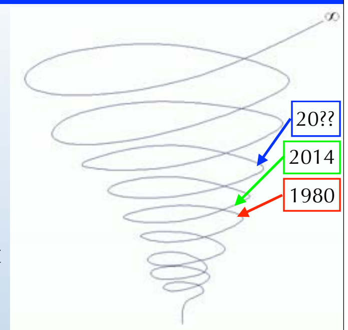
Courtesy of A. Holtmaat

- Three Two-photon image stacks taken a week apart in-vivo.
- Simultaneous reconstruction in all three stacks.

—> Automated change detection + More robust delineation.

Glowacki, CVPR'14

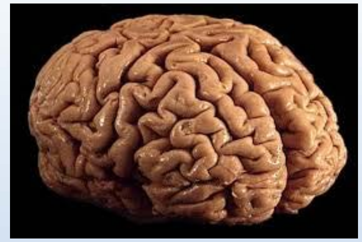
Conclusion



An automated reconstruction algorithm that

- reliably detects centerlines,
- relies on informative path classification scores,
- allows for loops and penalize early branch terminations,
- can be optimized exactly.

Future Work



A human brain contains approximately:

- 100 billion neurons.
- 100 trillion synapses.
- Some neurons extend over several cms.

- Develop optimisation algorithms that can handle these kinds of scales.
- Incorporate much more global connectivity constraints.
- Reduce the amount of training data required using Transfer Learning.

References

Centerline and Contour Detection.

- A. Sironi, V. Lepetit and P. Fua. Multiscale Centerline Detection by Learning a Scale-Space Distance Transform. Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, Ohio, USA, 2014.
- A. Sironi, E. Turetken, V. Lepetit and P. Fua. Multiscale Centerline Detection, submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014.

Delineating Curvilinear Structures

- E. Turetken, F. Benmansour and P. Fua, Reconstructing Loopy Curvilinear Structures Using Integer Programming, Conference on Computer Vision and Pattern Recognition, June 2013.
- E. Türetken, F. Benmansour, B. Andres, H. Pfister and P. Fua. Reconstructing Curvilinear Networks using Path Classifiers and Integer Programming, submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014.