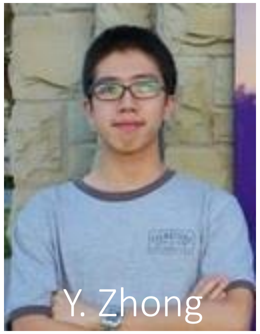
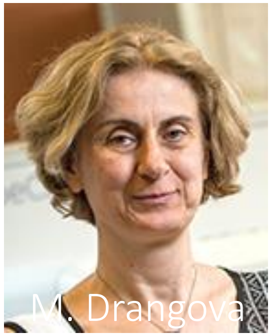
# *Curvature*, *Divergence*, and *Confluence*
## for unsupervised reconstruction of directed vessel trees

Yuri Boykov

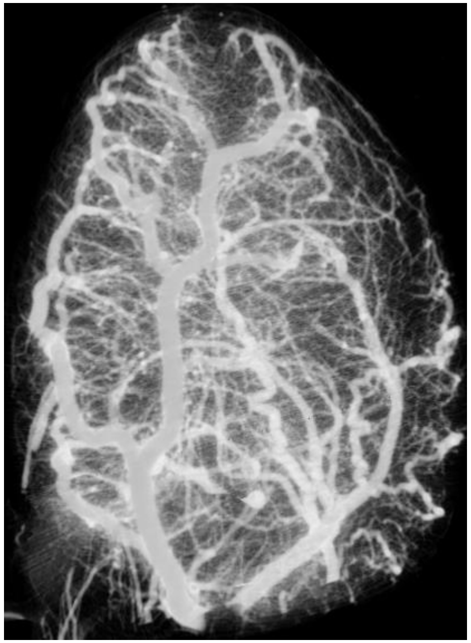
joint work with



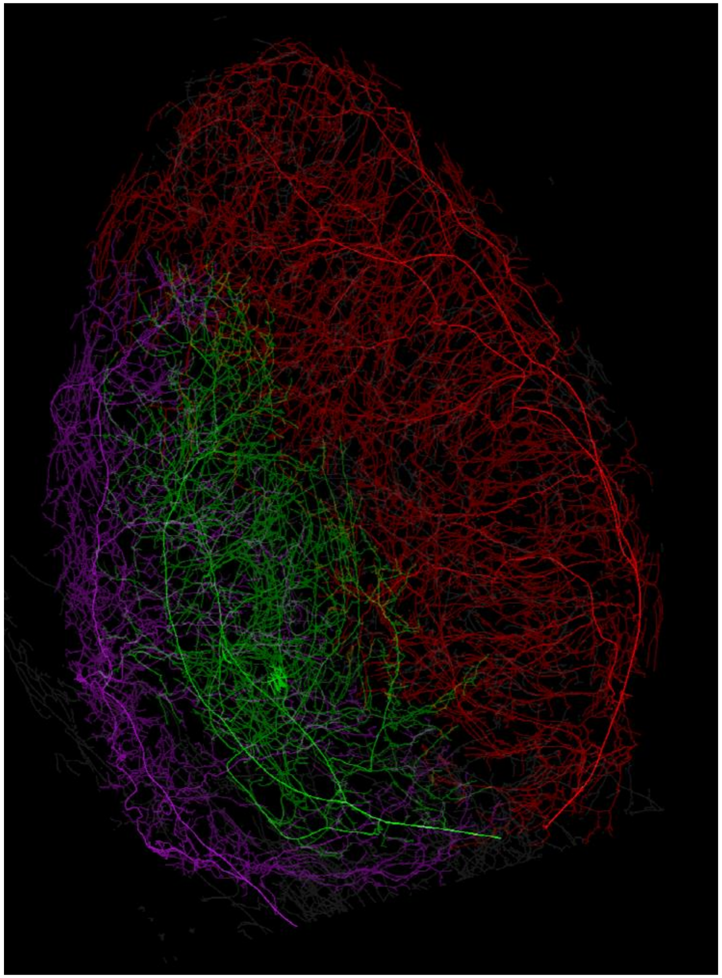


# Reconstructing vessel trees (outline)

- Challenges, basic techniques,
- Curvature - ICCV 2015
- Divergence - CVPR 2019
- Confluence - CVPR 2021



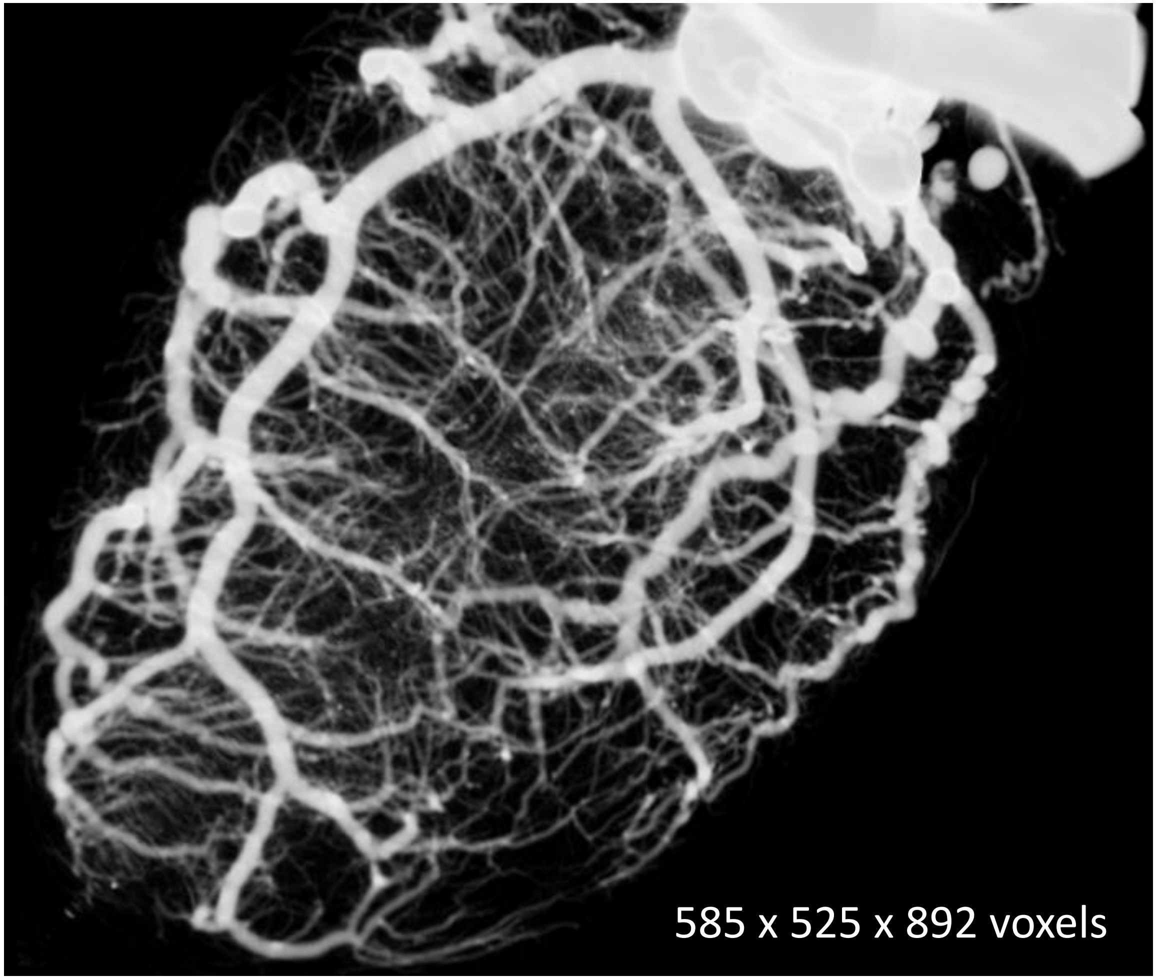
micro-CT vessel volume
mouse heart

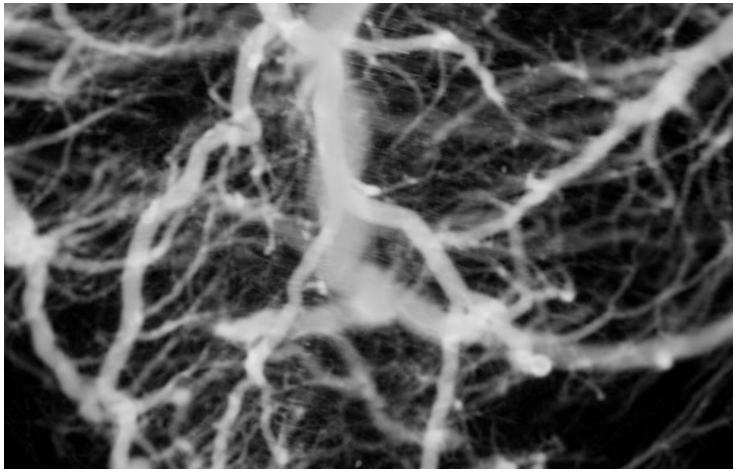


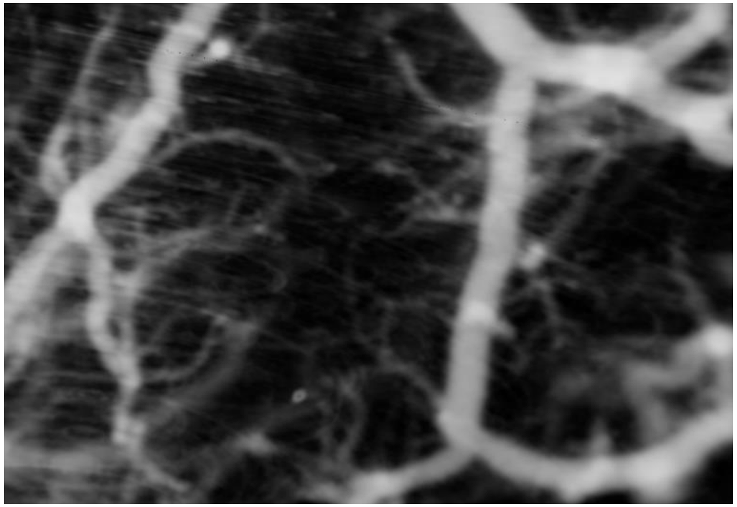
reconstructed tree structure

# Biomedical motivation

High resolution 3D imaging: **micro-CT**





**vascular data from Robarts Research, M. Drangova**

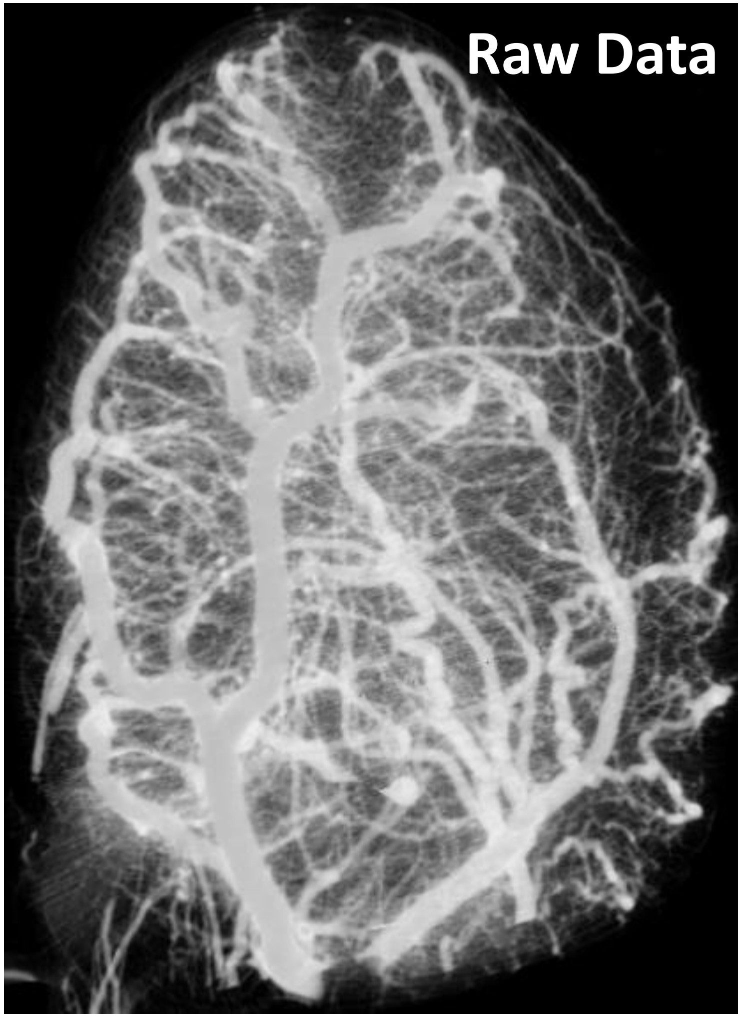
**most of the vessels are <u>thinner than voxel size</u>**

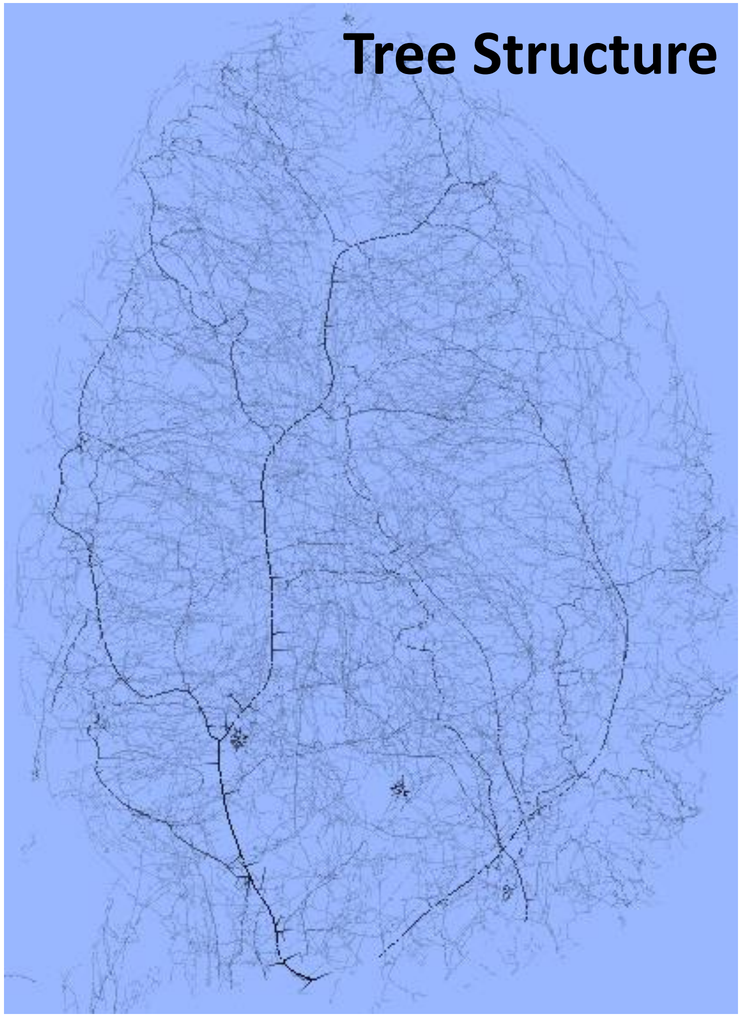
# Biomedical motivation



Goal: **vascular tree structure**

(bifurcation points, angles, connectivity,…)

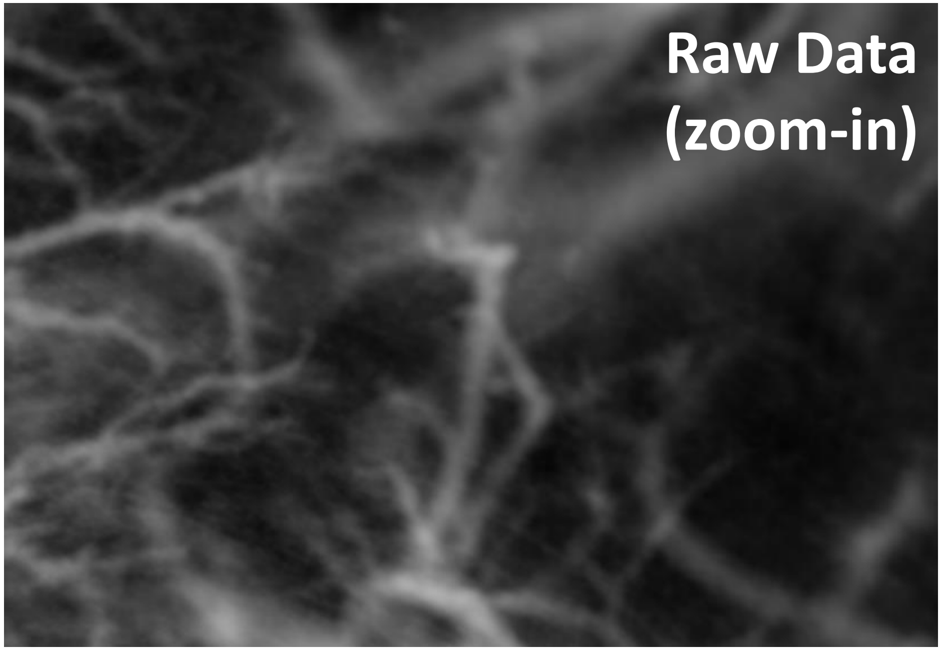


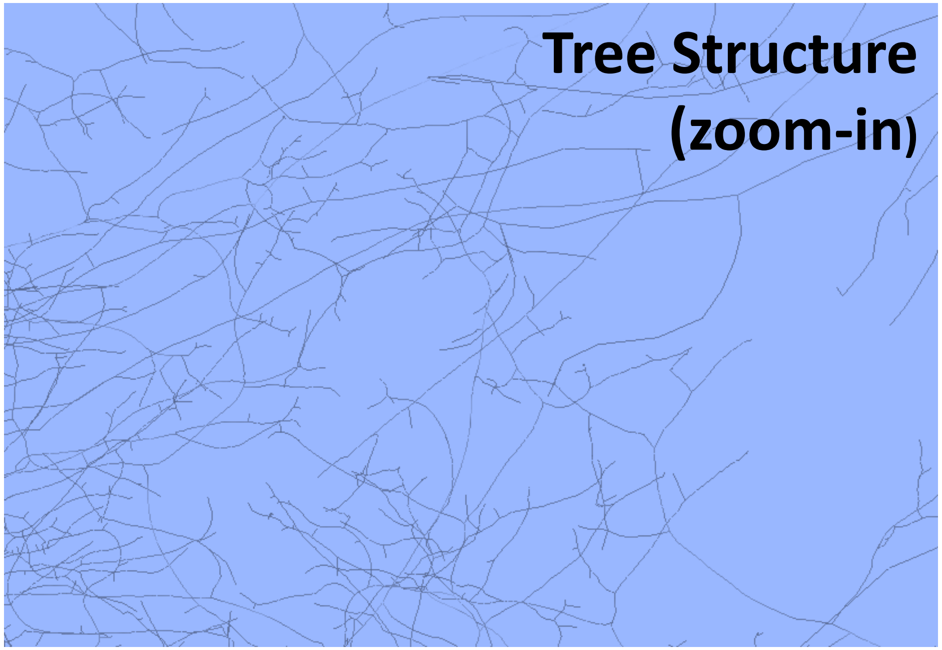



# Biomedical motivation

Goal: **vascular tree structure**

(bifurcation points, angles, connectivity,…)

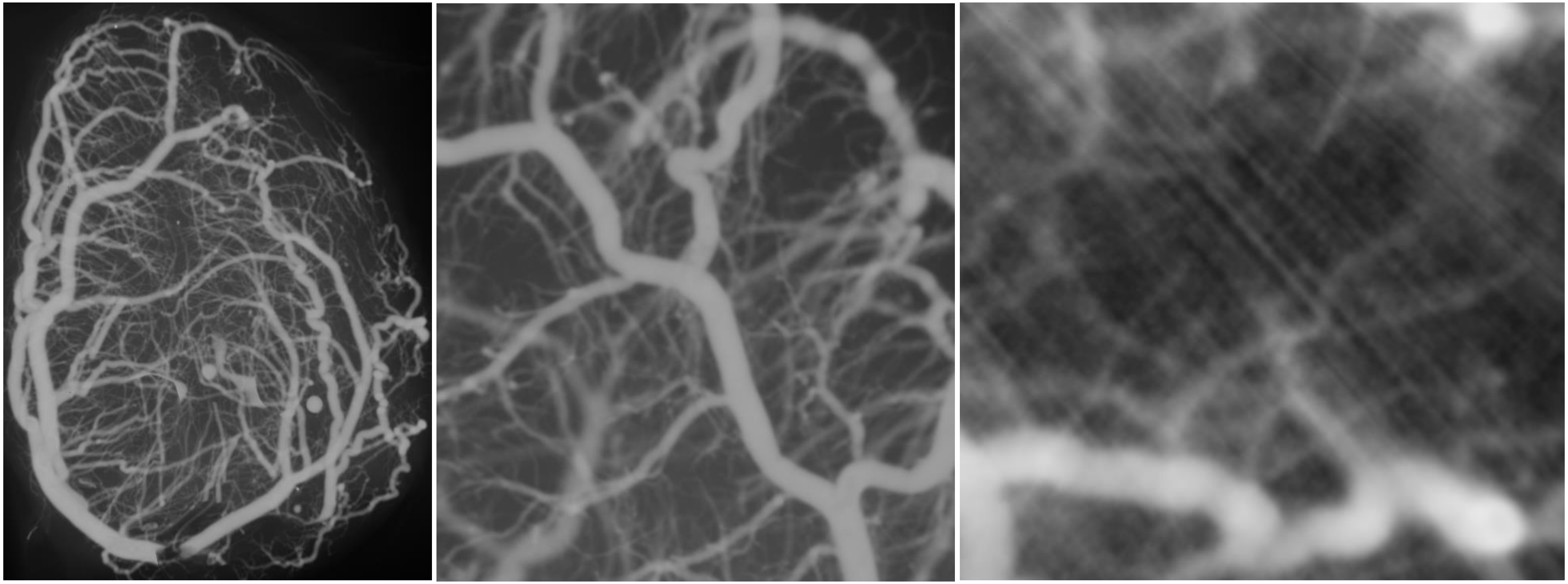



resolving near-capillary vessels

# Main Challenges:

**Noise**
**Ring artefacts**
**Loss of signal at thin vessels** (due to *partial voluming*)



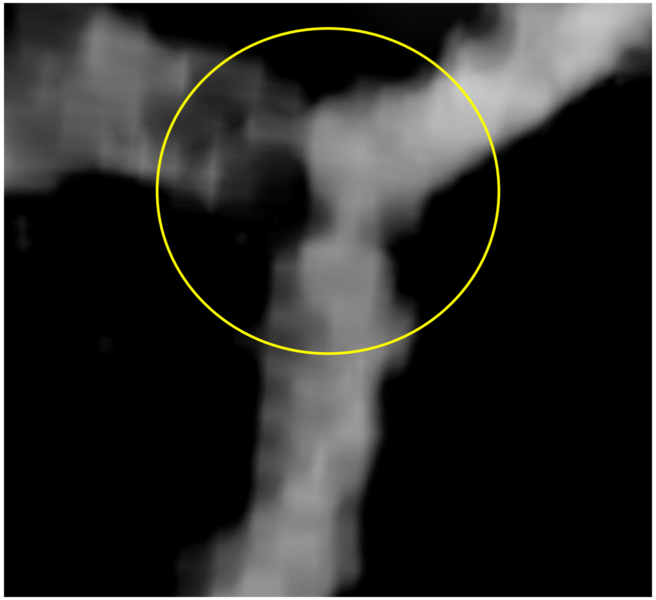
# Main Challenges:

Noise

Ring artefacts

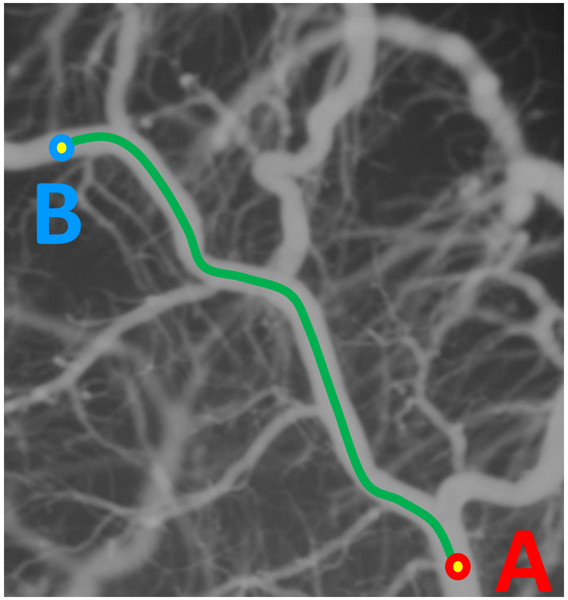
Loss of signal at thin vessels (due to *partial voluming*)

**Loss of signal at bifurcations** (due to *Frangi filtering*, more later)



# Main Challenges:

Noise
Ring artefacts
Loss of signal at thin vessels (due to *partial voluming*)
Loss of signal at bifurcations (due to *Frangi filtering*, more later)
**No user assistance** (except for one branch or very small trees)

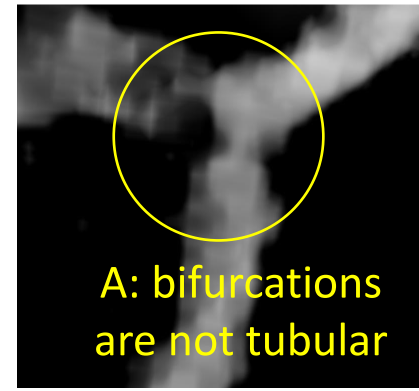


# Basic techniques

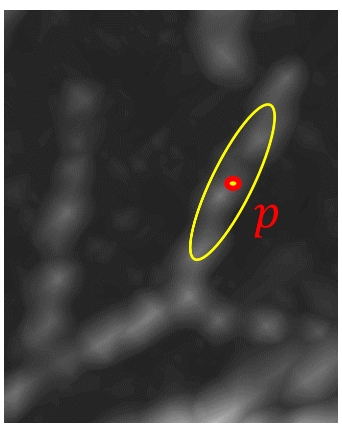
**Preprocessing** (data cleaning):

- ring artefact filtering
- tubular structure filtering [**Frangi** et al, 1998]

**Why?**

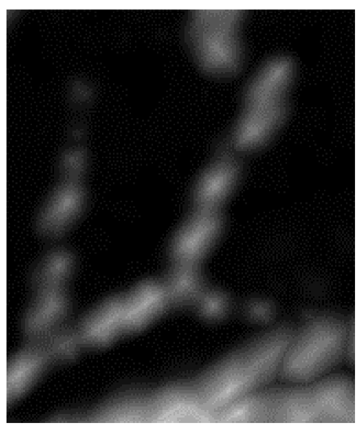



analyze
***Hessian***
of intensities
around $p$

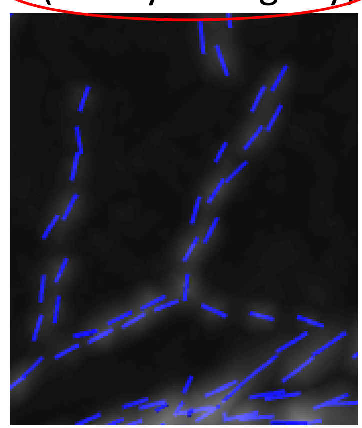



**raw data**

**vesselness measure**



**direction**
(binary ambiguity)

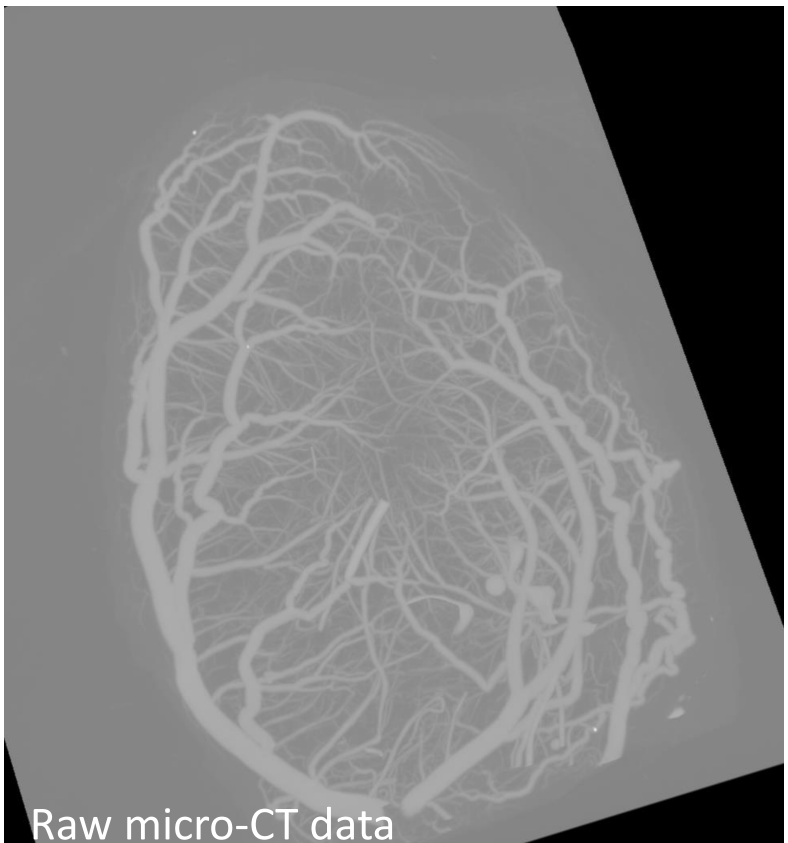


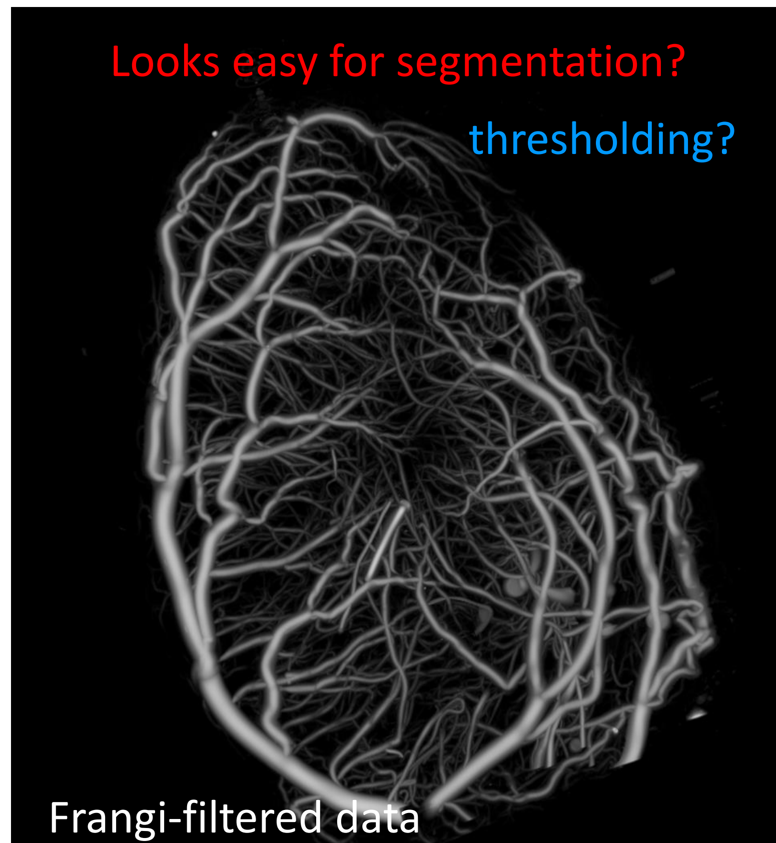
**diameter**

# Basic techniques

**Preprocessing** (data cleaning):

- ring artefact filtering
- tubular structure filtering [**Frangi** et al, 1998]



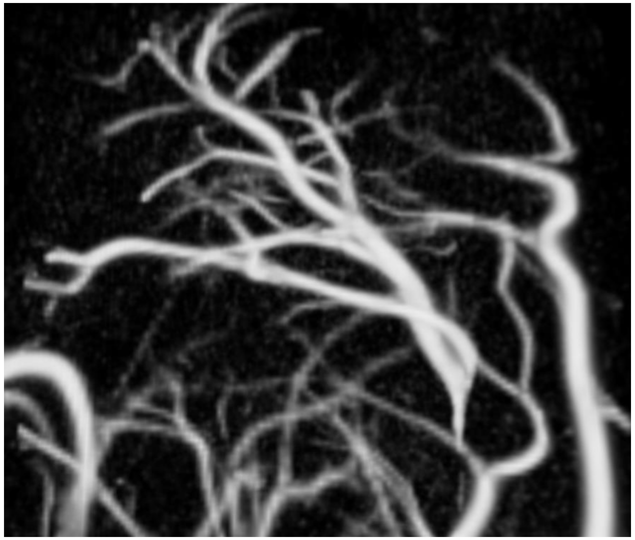
# Basic techniques

**Preprocessing** (data cleaning):

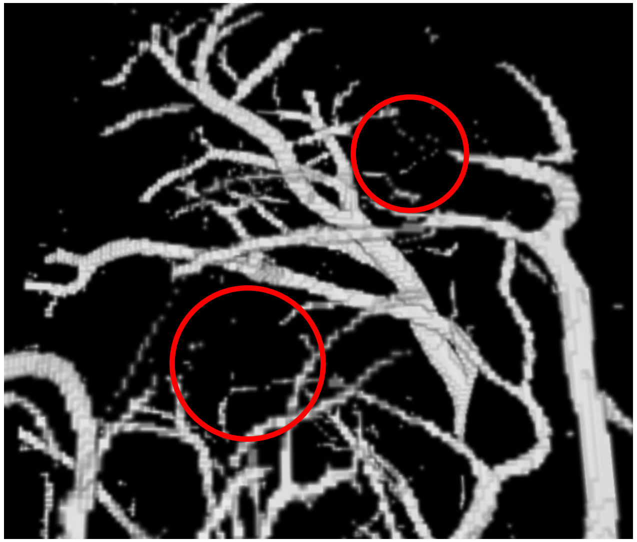
- ring artefact filtering
- tubular structure filtering [**Frangi** et al, 1998]

**Vessel segmentation**:

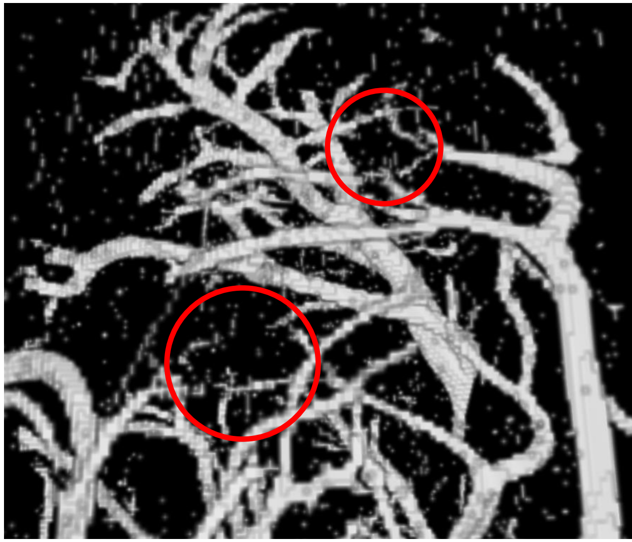
- **thresholding**



Frangi-filtered data (zoom-in)



higher threshold
loses thin vessels



lower threshold
keeps noise

# Basic techniques

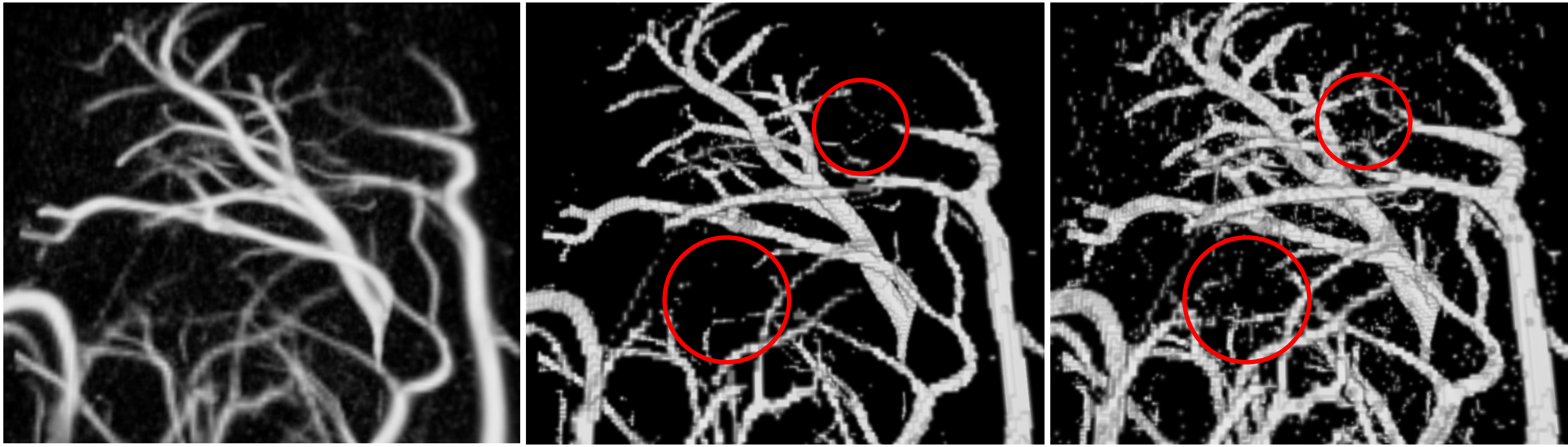
**Preprocessing** (data cleaning):

- ring artefact filtering
- tubular structure filtering [**Frangi** et al, 1998]

**Vessel segmentation**:

- ~~thresholding~~

vessel **continuation** problem
**regularization**



Frangi-filtered data (zoom-in) | higher threshold loses thin vessels | lower threshold keeps noise

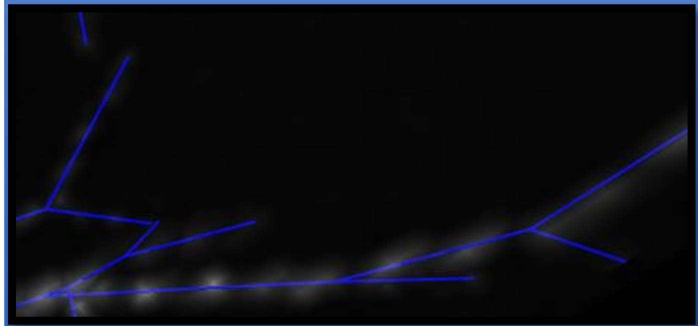
# Regularizing what?

(vessel representation?)

centerline regularization

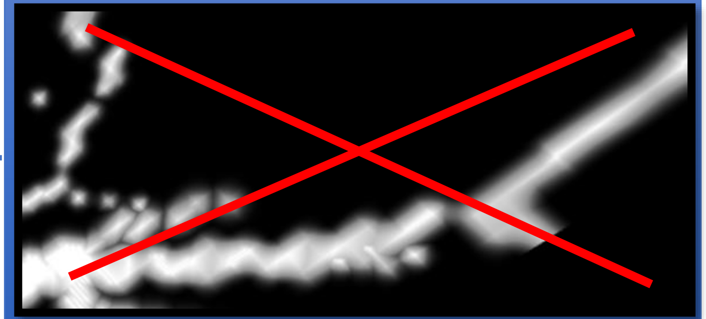
surface regularization

## centerline detection



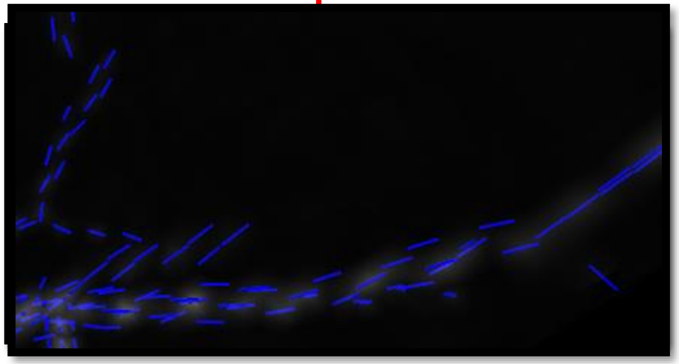
Skeletonization
[Sylvain Bouix *et.al.* 2005]

## volumetric segmentation



**1D curvature regularization**

[D. Marin *et.al.* *ICCV* 2015]
[E. Chesakov 2015]



**Frangi output**

Area (first-order regularization)

[Caselles, Kimmel, Sapiro, 1995]

[Boykov Kolmogorov, 2003]

Mean curvature

[J.Yi *et.al.* 2003]

[P. Strandmark *et.al.* 2011]

[T. Schoenemann *et.al.* 2012]

[C. Nieuwenhuis *et.al.* 2014]

Gaussain or min curvature

[?]

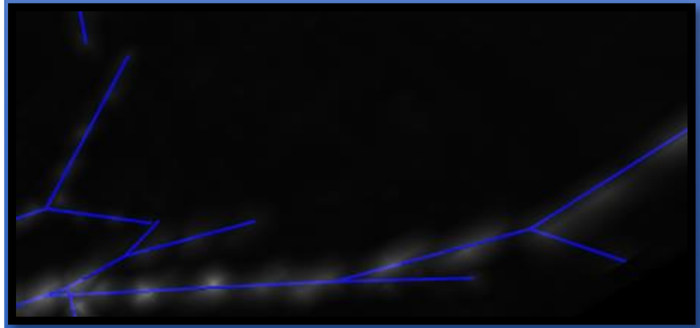
# Regularizing what?

(vessel representation?)

centerline regularization

surface regularization

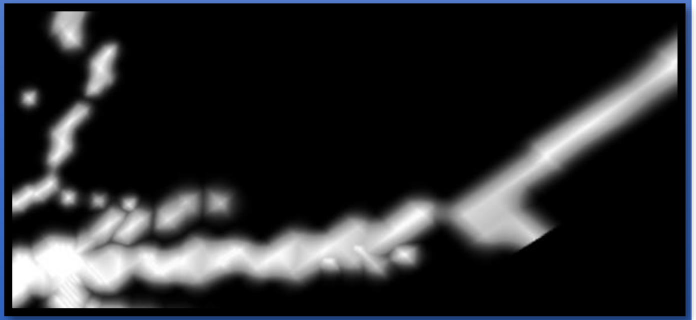
## centerline detection



MAT
[Kaleem Siddiqi *et.al.* 2008]

Skeletonization
[Sylvain Bouix *et.al.* 2005]

## volumetric segmentation



# Outline

A

**Preprocessing**:

- ring artefact filtering
- tubular structure filtering [Frangi et al, 1998]

B

**Smooth "centerline" estimation**

- denoising
- local connectivity
- flow direction

$\Rightarrow$ connectivity graph estimation, a.k.a.

**directed Tubular graph**

C

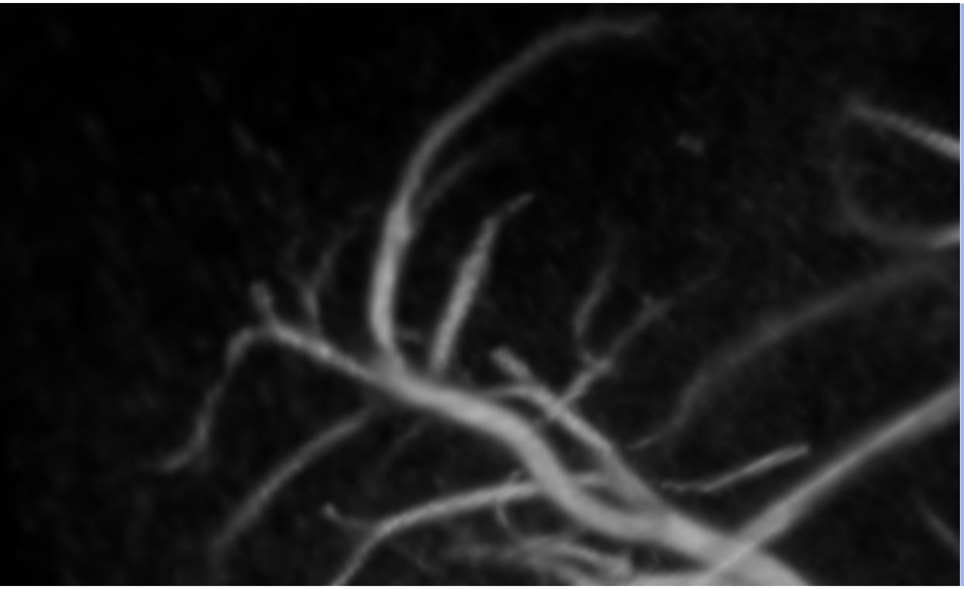
**Tree topology estimation** (over local connectivity graph):

- ~~shortest path (assumes user-specified end points)~~
- variants of *minimum spanning tree* (MST)
- ***minimum arborescence*** (directed tree)

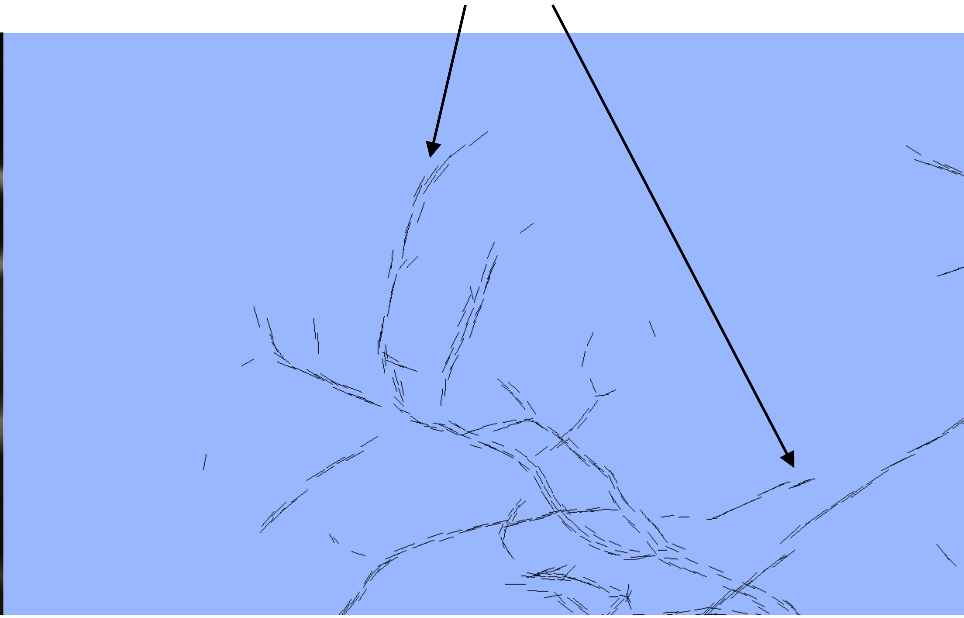
# Motivating example:

noisy **vessel tangents** observations

local vessel orientations from Frangi



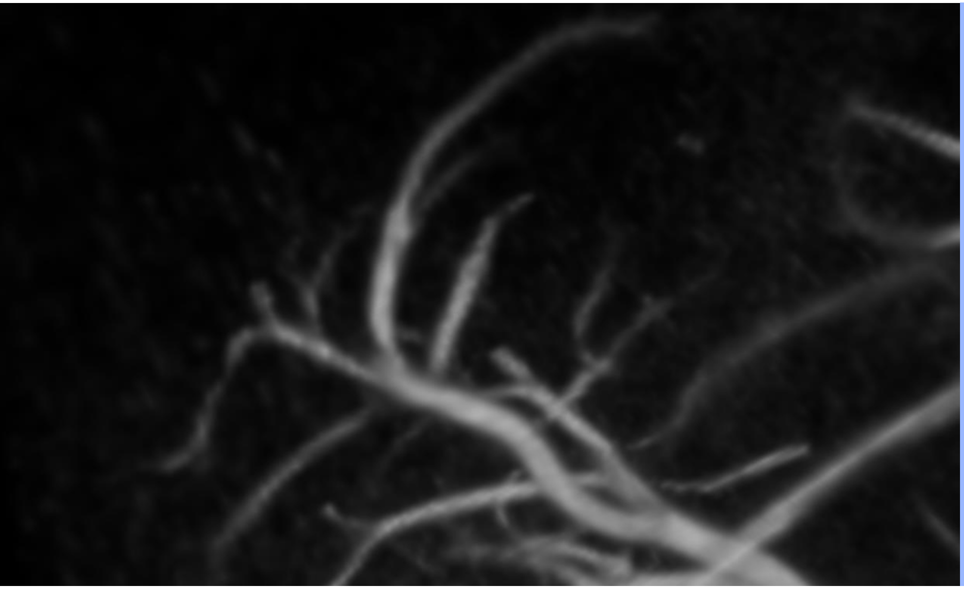
vesselness measure (Frangi)



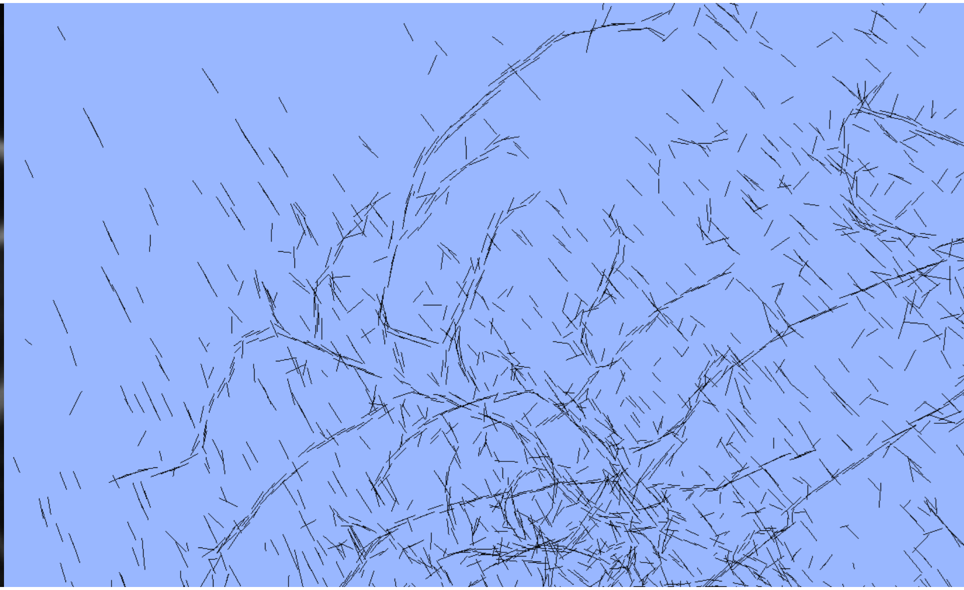
(high threshold)

# Motivating example:

clutter, outliers



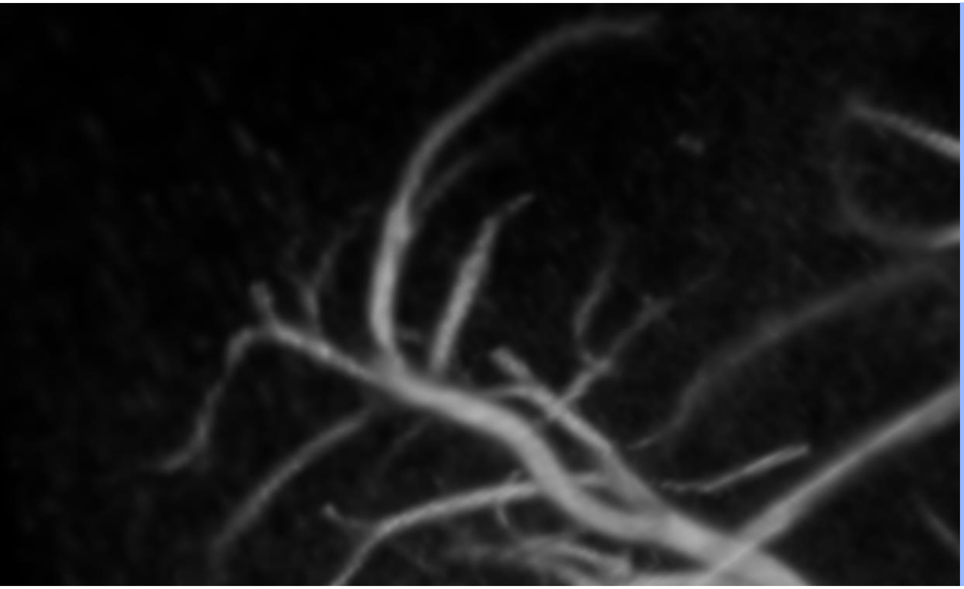
vesselness measure (Frangi)



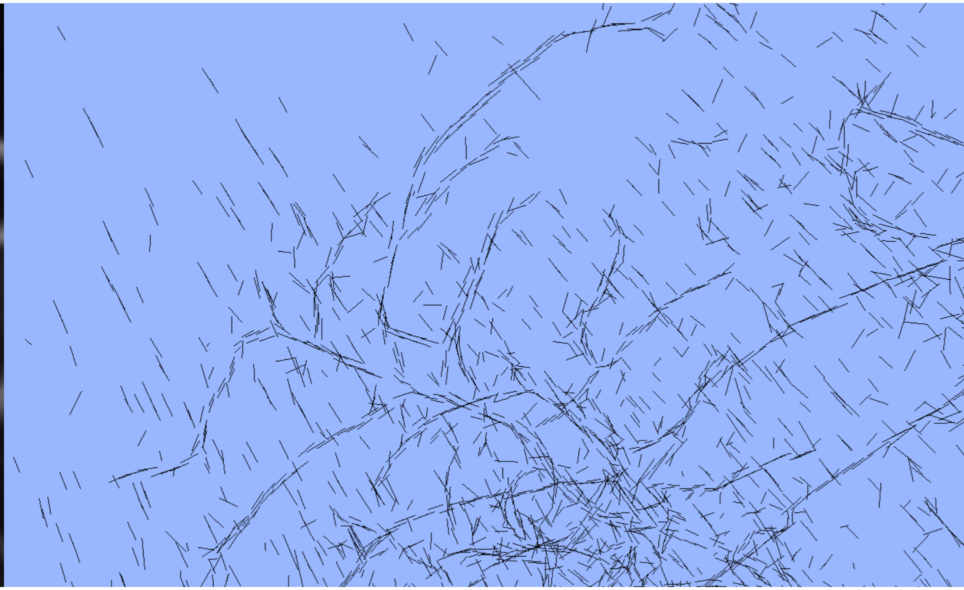
(low threshold)

tree growth from seeds (local heuristics) a la “Canny edges”
[Aylward et al. 2002]

**Our approach**: regularize local tangents via curvature



vesselness measure (Frangi)



(low threshold)

# Our prior work:

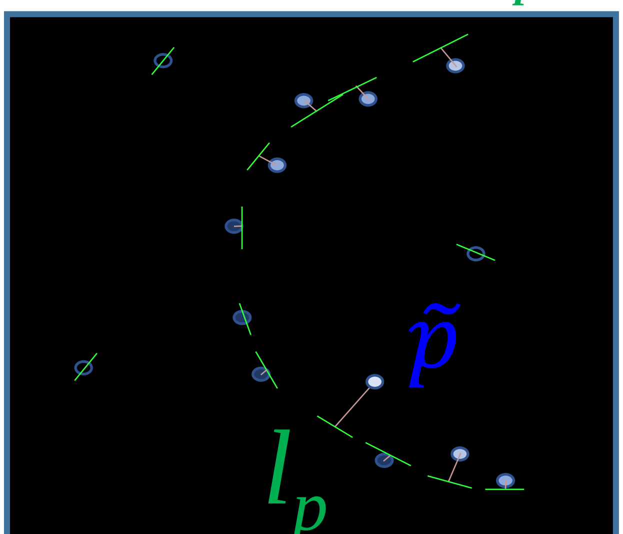
## implicit curve/surface fitting [Olsson et al CVPR 2012,13]

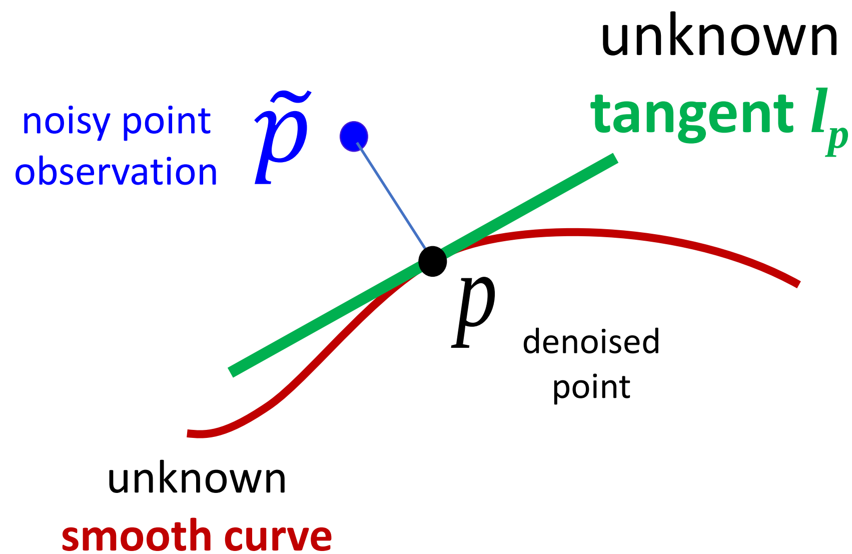
$$E(L) = \sum_p \frac{1}{\sigma_p^2} \left\| l_p - \tilde{p} \right\|^2 + \lambda \sum_{p,q \in N} \kappa^2(l_p, l_q)$$

fitting errors

regularization
(smooth tangents)

estimate
local **tangents** $l_p$



# Our prior work:

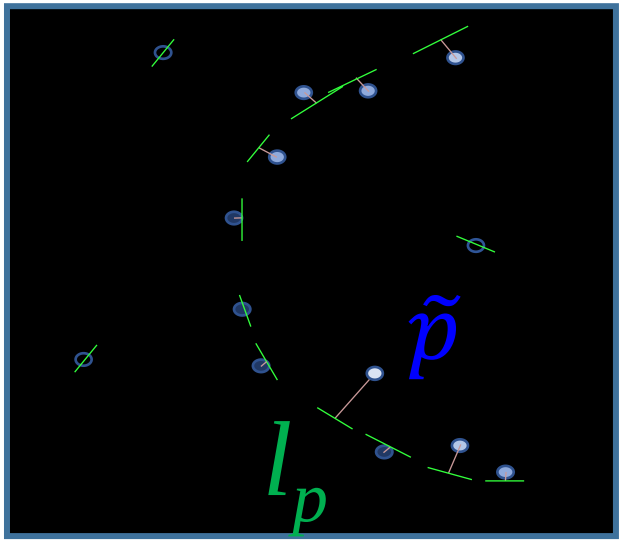
## implicit curve/surface fitting [Olsson et al CVPR 2012,13]

$$E(L) = \sum_p \frac{1}{\sigma_p^2} \left\| l_p - \tilde{p} \right\|^2 + \lambda \sum_{p,q \in N} \kappa^2(l_p, l_q)$$

fitting errors

regularization
(smooth tangents)

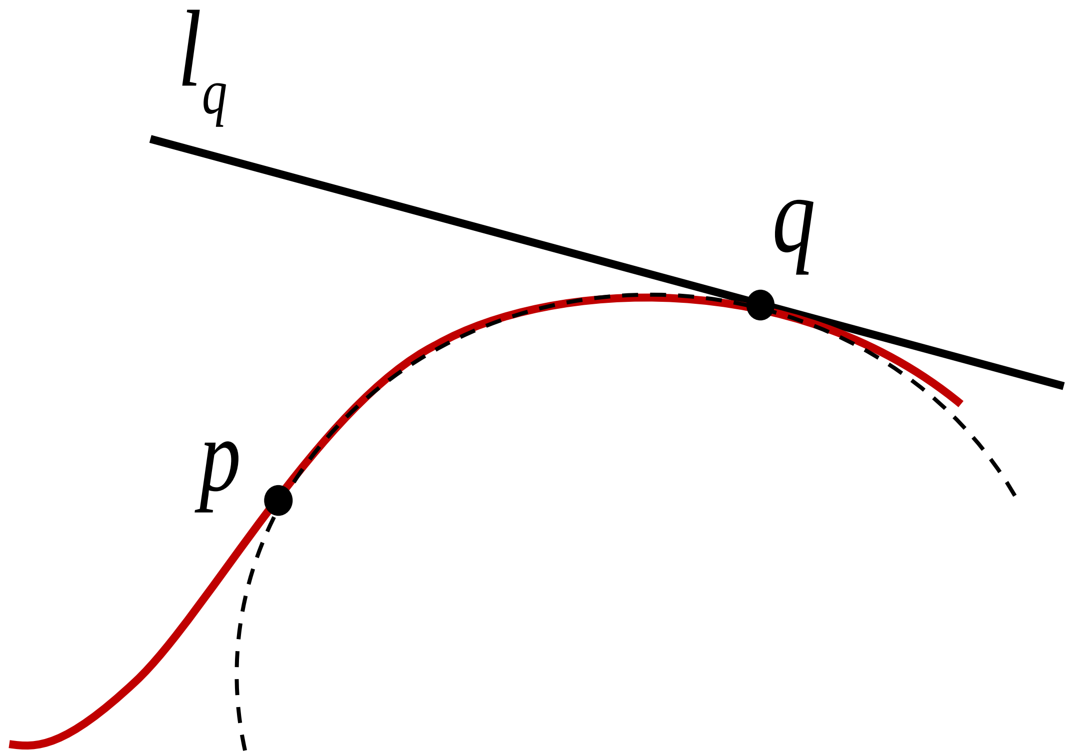
estimate
local **tangents** $l_p$




**curvature** of implicit curve
between two points
can be **estimated from tangents**
(under mild assumptions)

# Our prior work:

## curvature estimation [Olsson et al CVPR 2012,13]

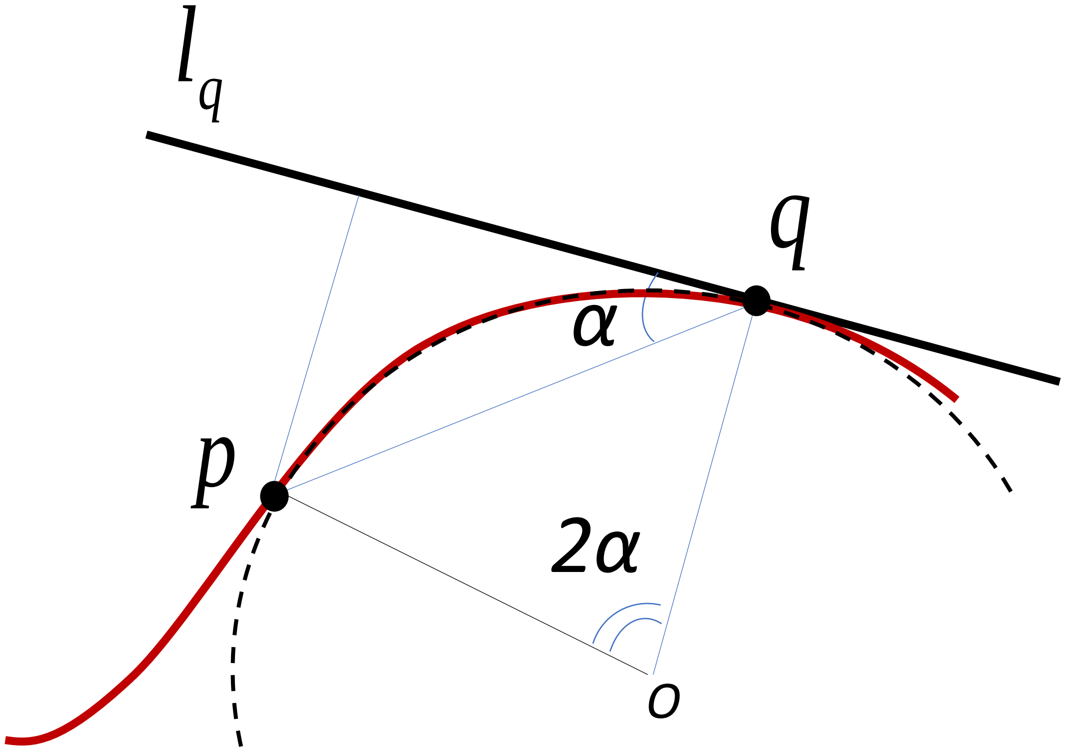



One tangent and a point are enough to estimate curvature
(assuming curve has constant curvature in between)

# Our prior work:

## curvature estimation [Olsson et al CVPR 2012,13]

$$\alpha \approx \frac{\|p - l_q\|}{\|p - q\|}$$

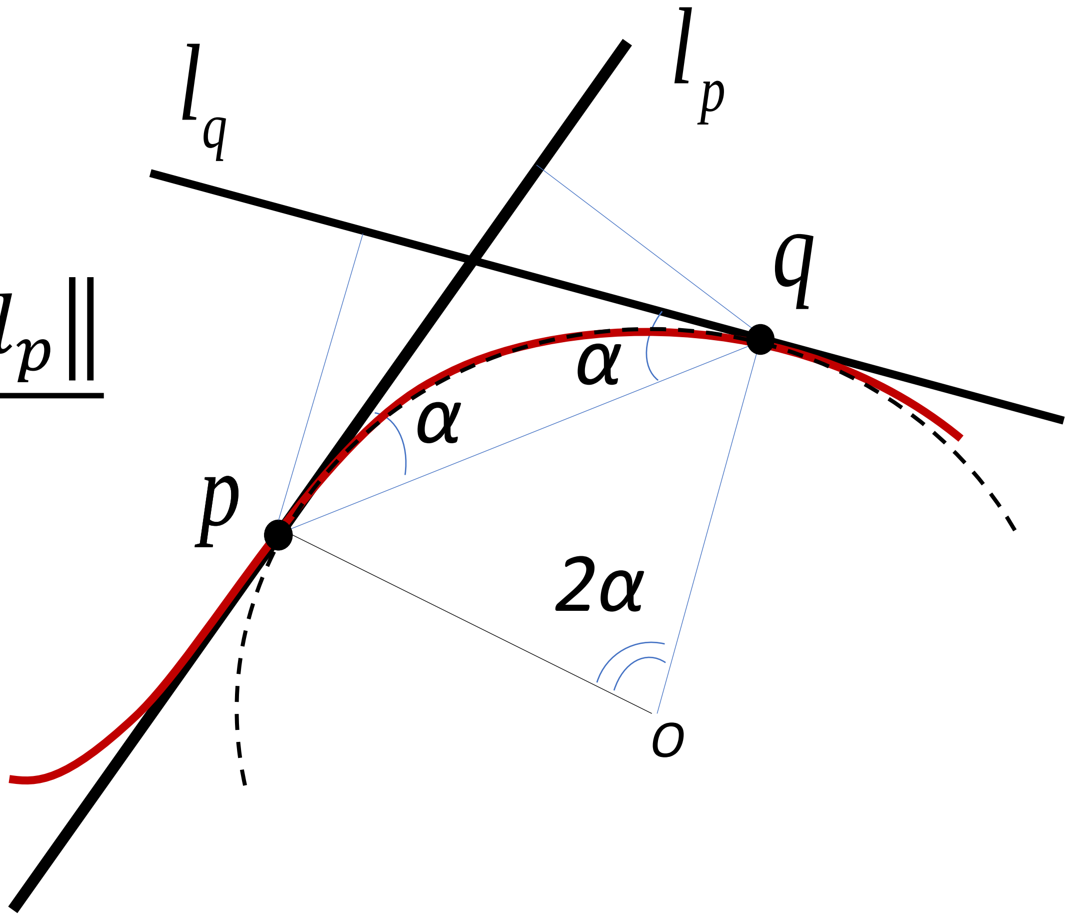



One tangent and a point are enough to estimate curvature
(assuming curve has constant curvature in-between)

# Our prior work:

## curvature estimation [Olsson et al CVPR 2012,13]

$$\alpha \approx \frac{\|p - l_q\| + \|q - l_p\|}{2\|p - q\|}$$




symmetric version using two tangents

# Our prior work:

[Olsson et al CVPR 2012,13]

**absolute curvature approximation:**

$$\int_p^q |\kappa| \cdot ds \approx 2\alpha \approx \frac{\|q - l_p\| + \|p - l_q\|}{\|p - q\|} \equiv \kappa(l_p, l_q)$$

**squared curvature approximation:**

$$\int_p^q |\kappa|^2 \cdot ds \approx \frac{\|q - l_p\|^2 + \|p - l_q\|^2}{\|p - q\|^3} \equiv \kappa^2(l_p, l_q)$$

$$E(L) = \sum_p \frac{1}{\sigma_p^2} \|l_p - \tilde{p}\|^2 + \lambda \sum_{p,q \in N} \kappa^2(l_p, l_q)$$

fitting errors

regularization

# Curvature regularized centerline fitting

Example of curvature regularization for centerline (tangent) fitting

# HPC: GPU accelerated optimization



Inexact Levenberg-Marquardt [Wright and Holt 1985]

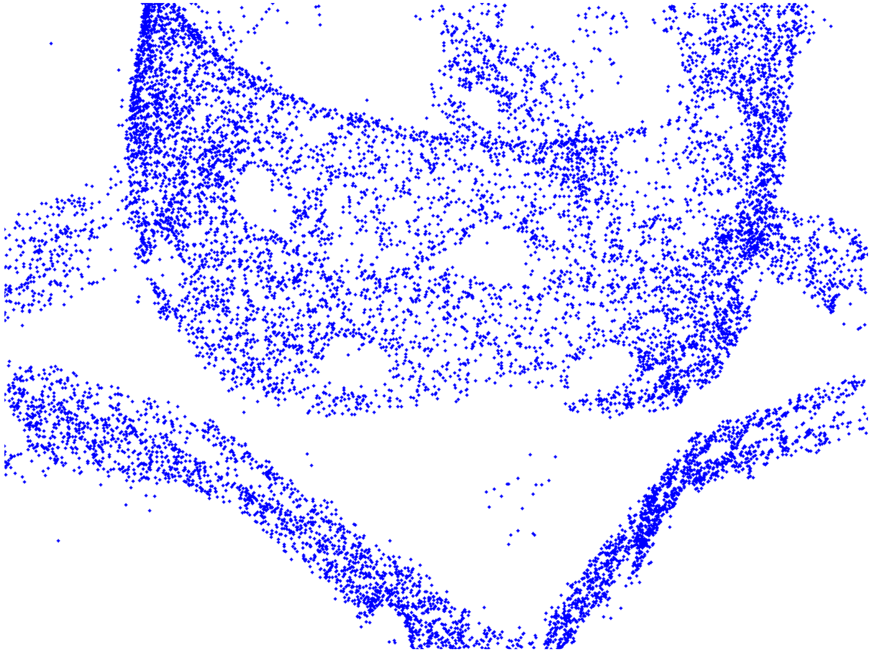
- Designed for solving **sparse** non-linear large least squares problem
- **Requires** efficient sparse matrix algebra implementation
- **Requires** the Jacobian computed at each iteration
- Automatic differentiation

[Chesakov, 2015]

# Prior work:

## used in stereo and N-view reconstruction [Olsson et al 2012, 13]

multiple images of object
(different view points)



noisy 3D points

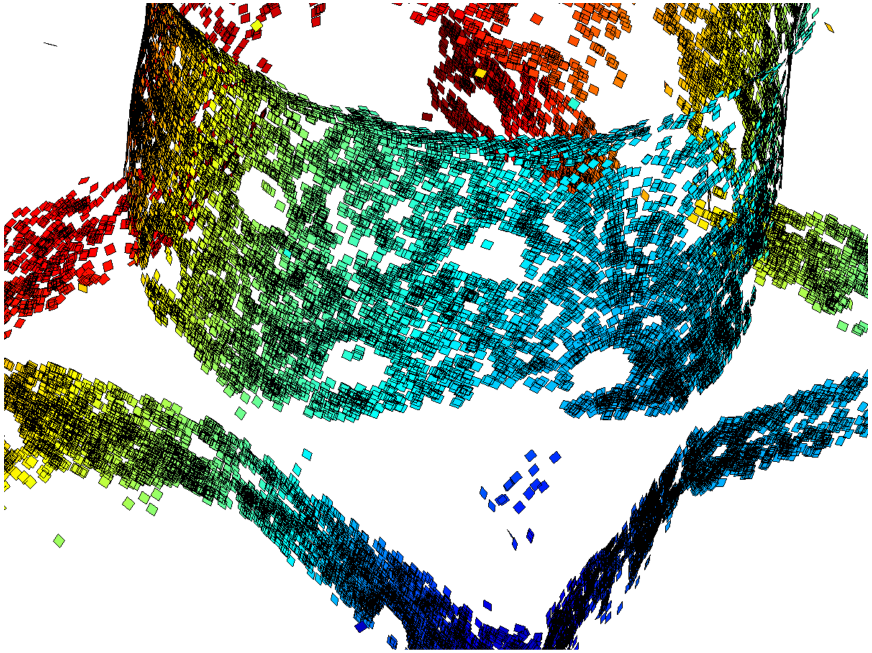
Should fit a smooth surface

# Prior work:

UNIVERSITY OF WATERLOO

## used in stereo and N-view reconstruction [Olsson et al 2012, 13]

multiple images of object
(different view points)



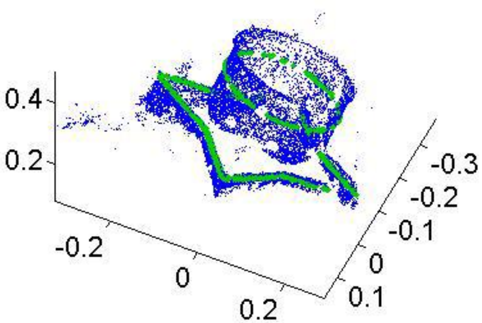
smoothly fit local tangents
(color = orientation)

- **tangent planes** instead of **tangent lines**

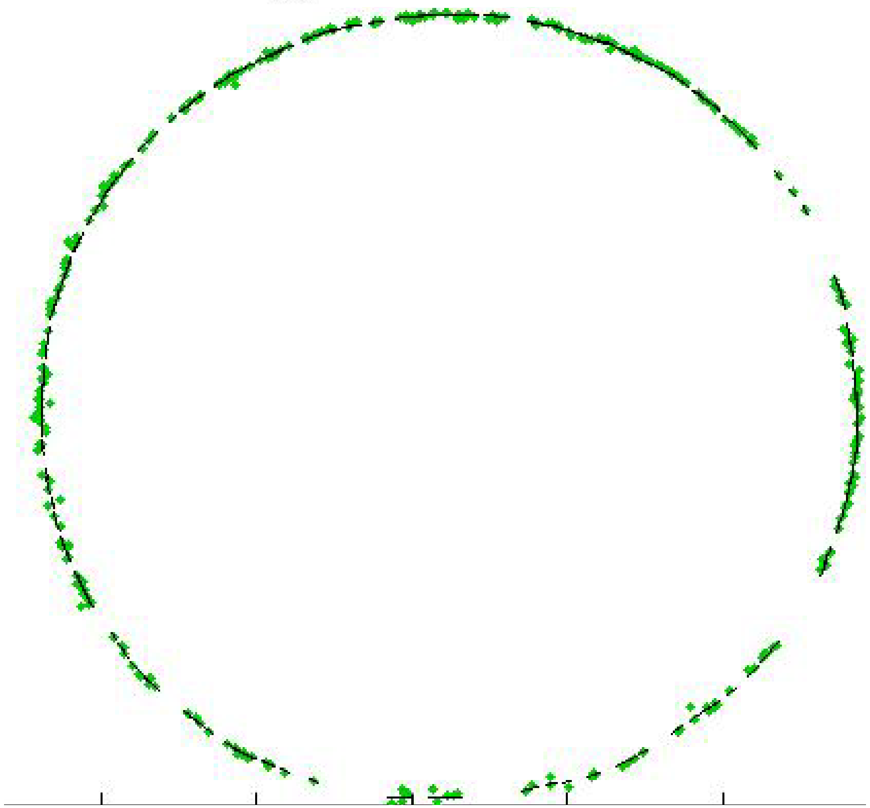
$-\sum_{p,q \in N} \kappa^2(l_p, l_q)$ approximates **mean curvature of surface** in 3D instead of basic curvature of 1D curve (in 2D or 3D)

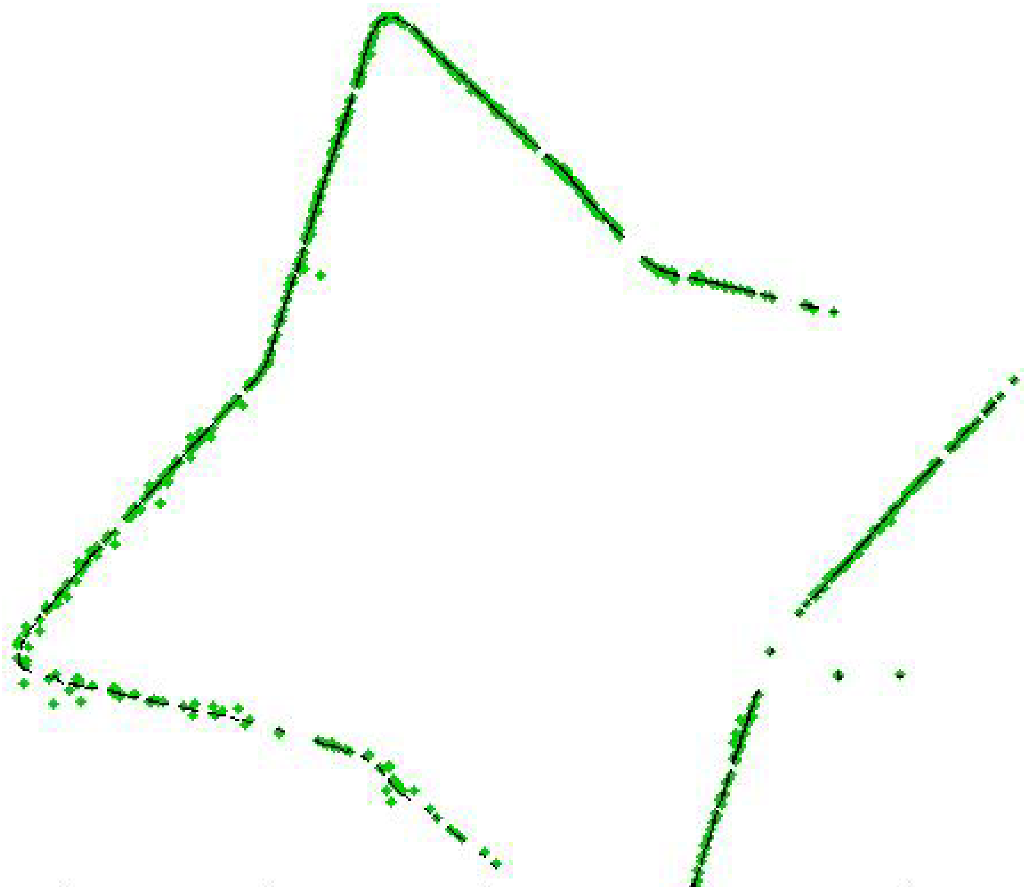
# Prior work:

## used in stereo and N-view reconstruction [Olsson et al 2012, 13]




NOTE:
top of Middlebury stereo (2017)
is based on this curvature model





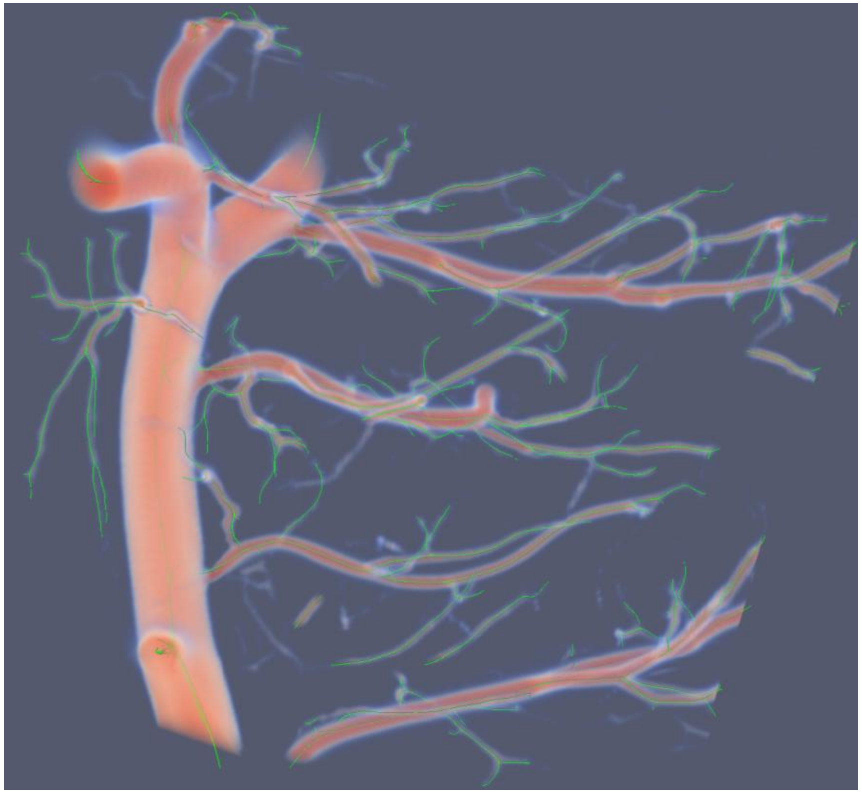
# Joint fitting and detection [Marin et al ICCV 2015]

unary potentials

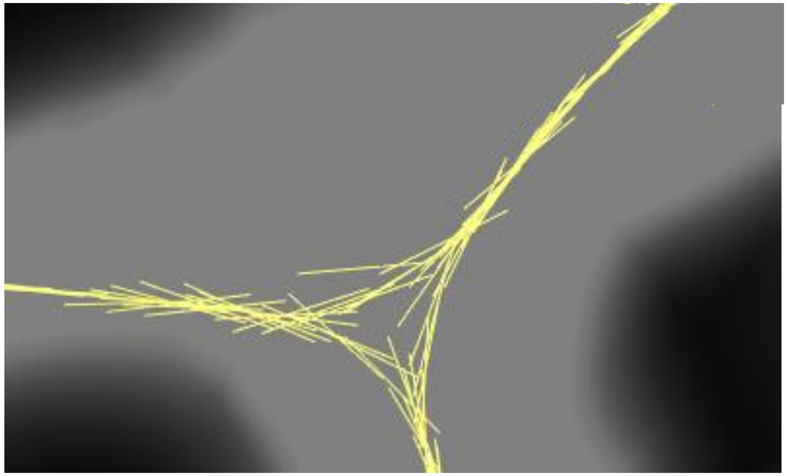
$$E(L,X) = \sum_{(i,j)\in N} \kappa^2(l_i,l_j)x_i x_j + \sum_i \frac{1}{\sigma^2}\|l_i - \widetilde{p}_i\|^2 + \sum_i \lambda_i x_i$$

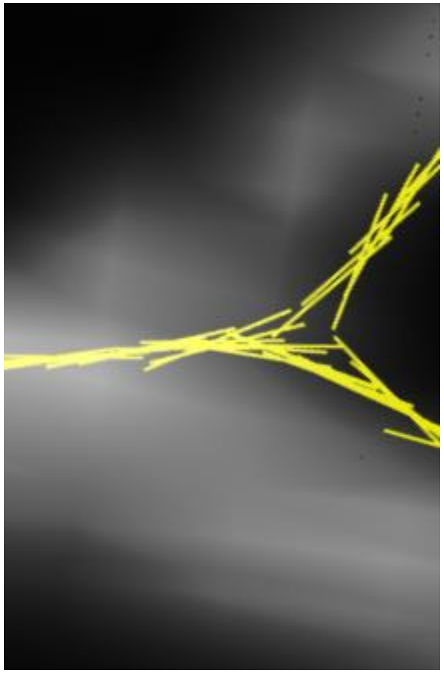
$X_i$ = 1 or 0
(vessel or not)

mean-field approximation
gives probabilities in [0,1] for X



## issues: artifacts at bifurcations





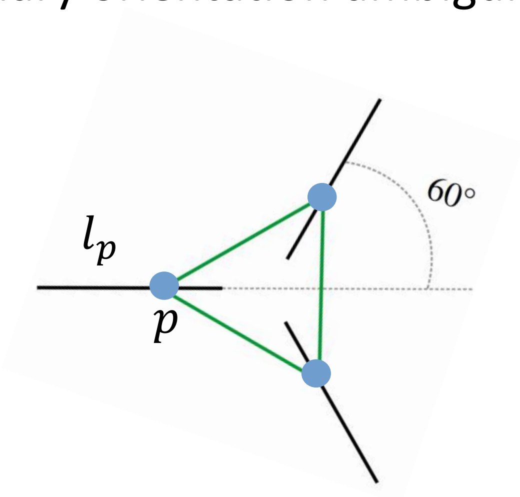
**intuition**: no flow orientation

# towards **directed** **Tubular graphs**…

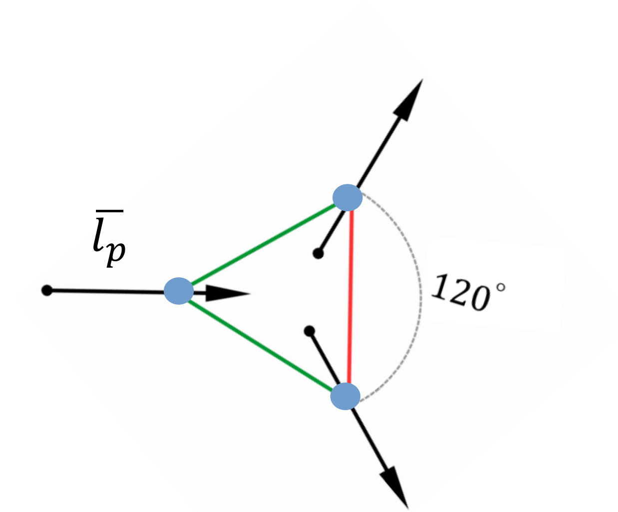
# Artifacts at bifurcation?



**unoriented** tangents
(binary orientation ambiguity)



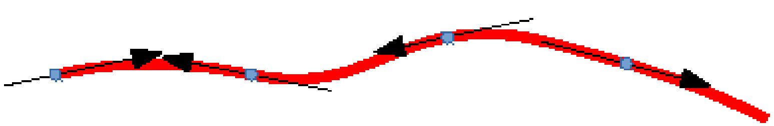

**oriented** tangents



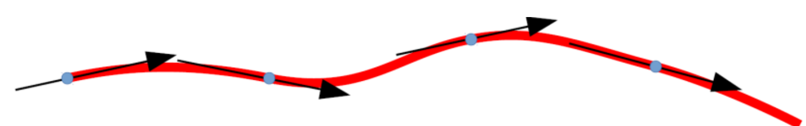

$$\bar{l}_p = x_p \cdot l_p$$

$$x_p \in \{-1, 1\}$$

**robust**
**oriented curvature**

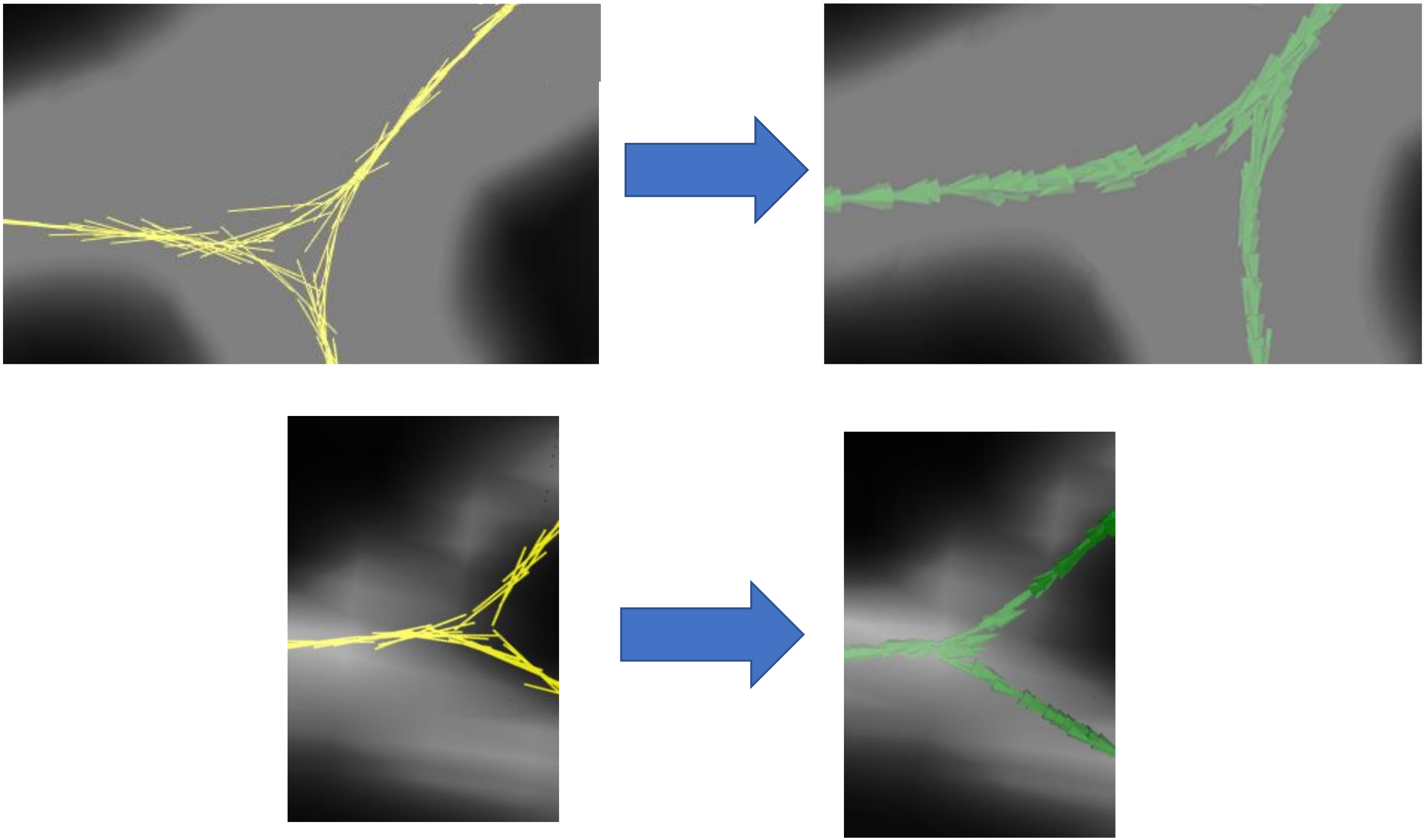


low energy



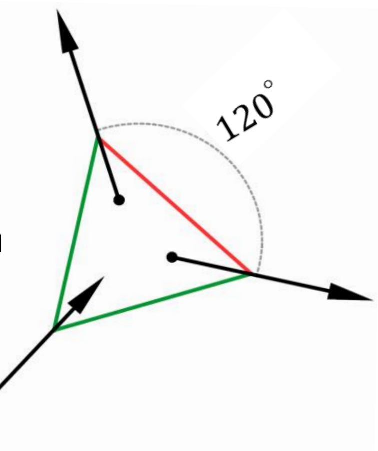
low energy
when consistent directions

# orientated curvature breaks "loops"

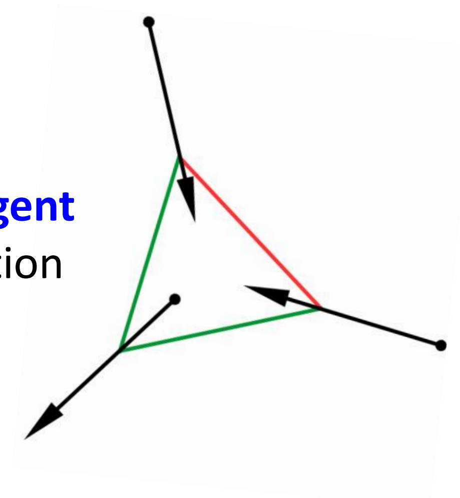


# However…

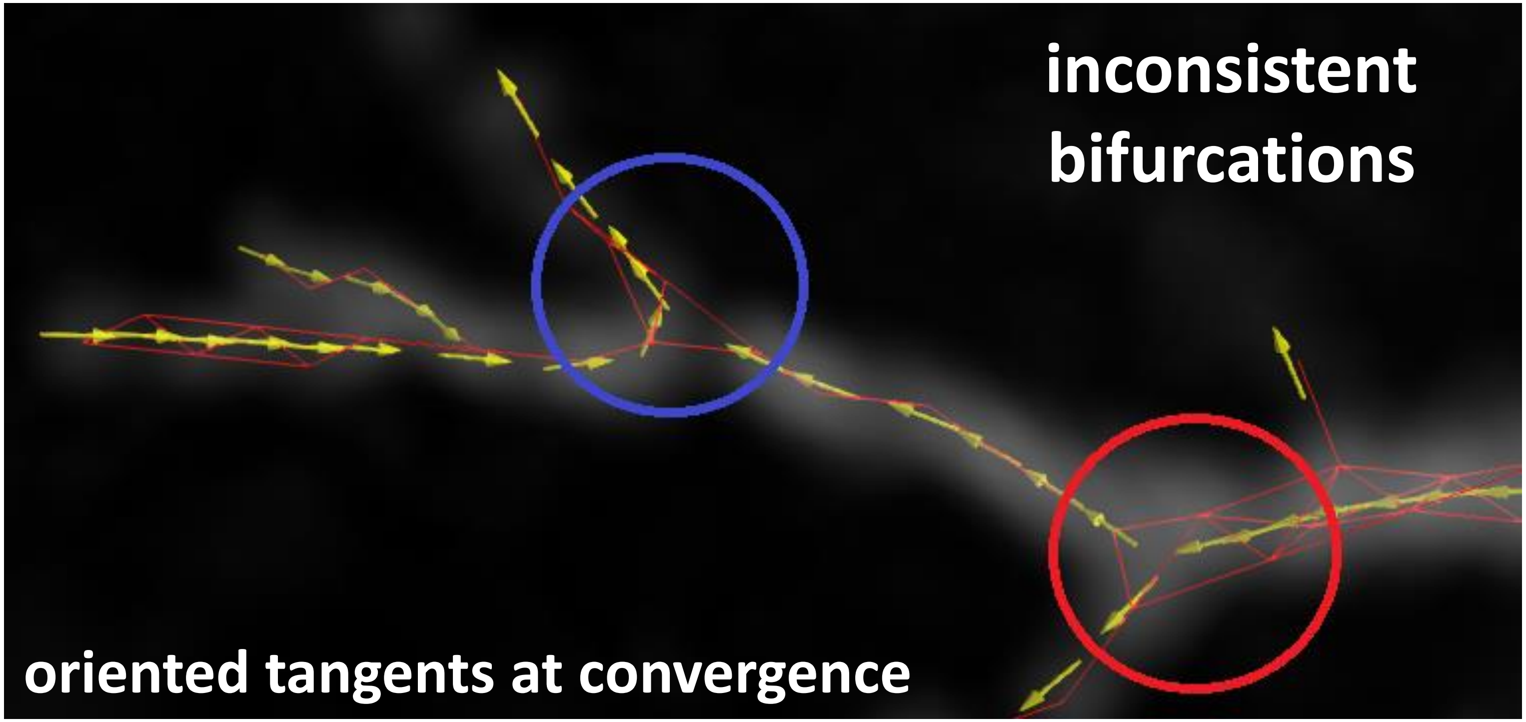
**divergent** bifurcation

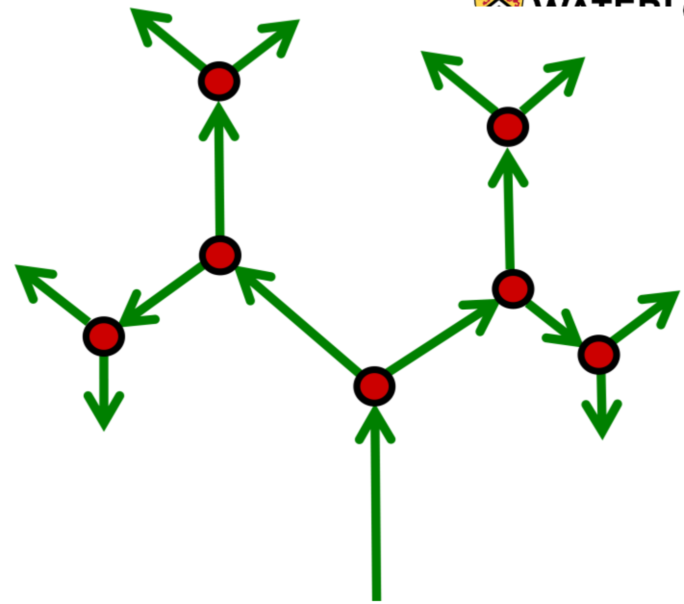



**convergent** bifurcation

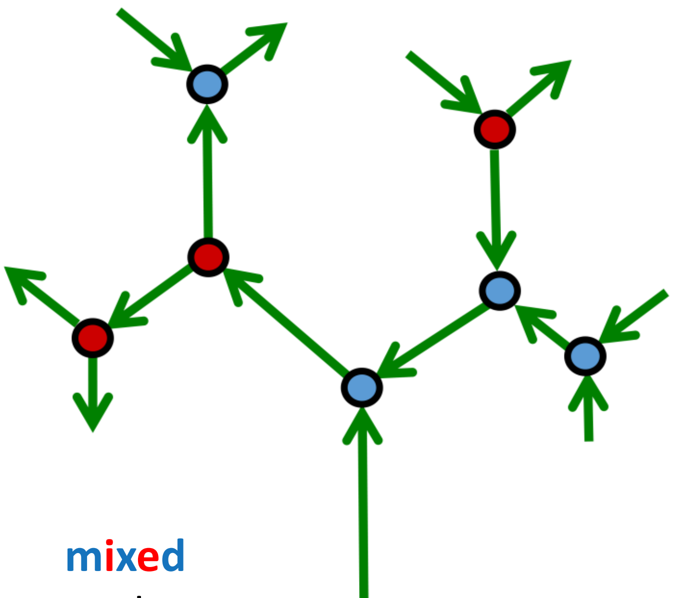


**two equally good solutions!**

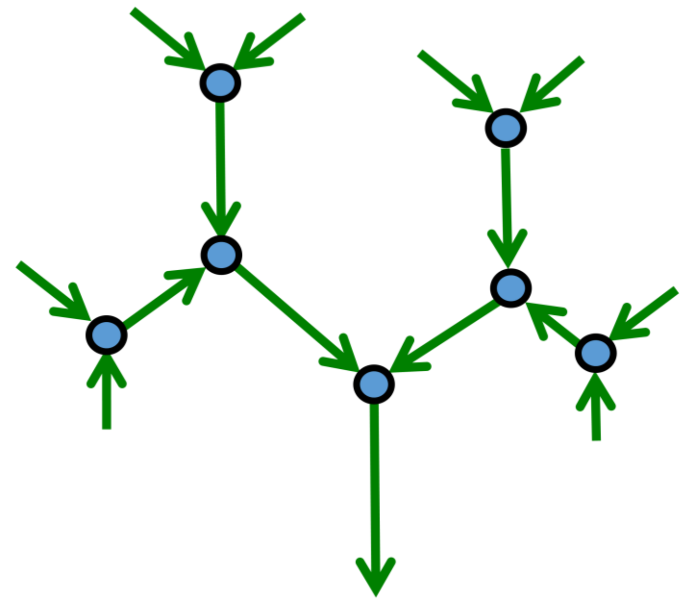

inconsistent bifurcations
oriented tangents at convergence



**divergent**
vessel tree
(AORTA)



**mixed**
vessel tree

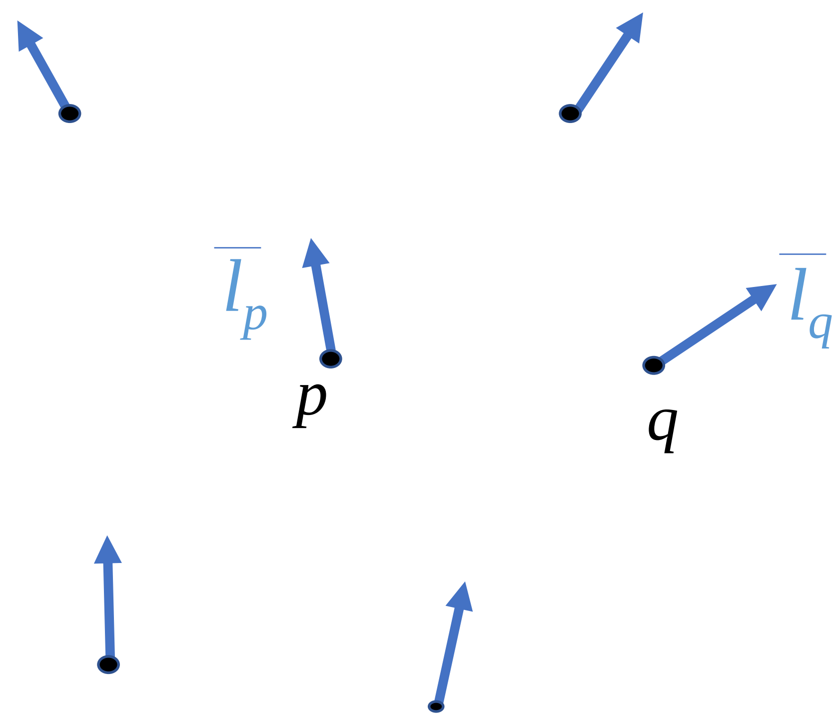


**convergent**
vessel tree
(VEIN)

# Divergence prior

enforcing **consistent flow pattern… divergent** (or convergent)

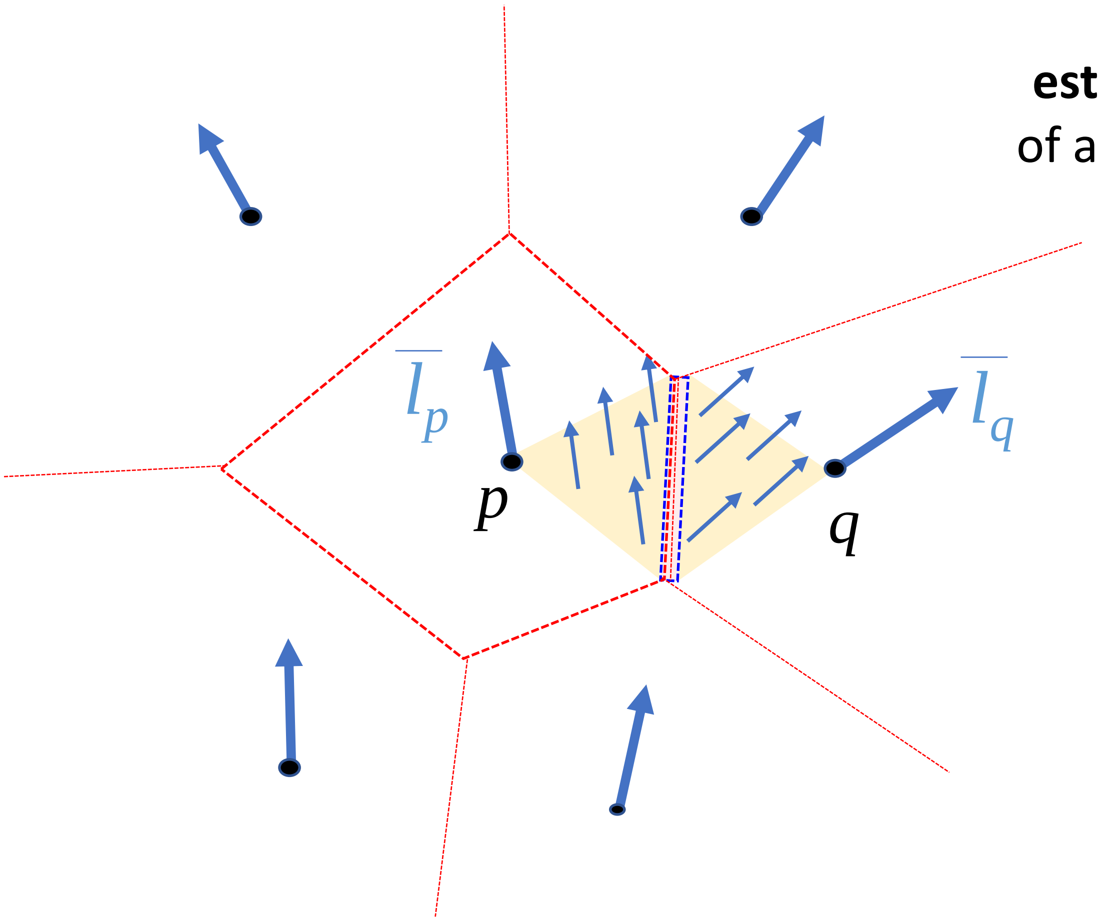
First: how to **estimate divergence** of a given vector field?



# Divergence prior

enforcing **consistent flow pattern…** **divergent** (or convergent)

First: how to **estimate divergence** of a given vector field?

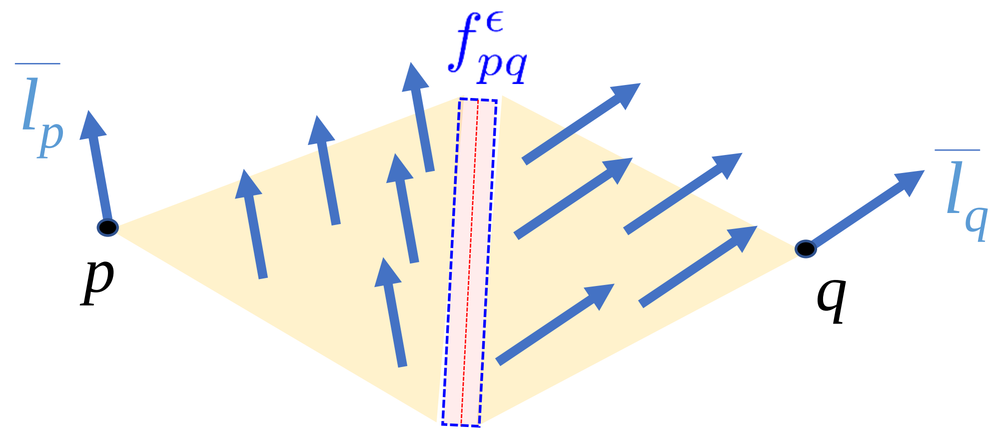



assume constant vector field inside each Voronoi cell

# Divergence prior

enforcing **consistent flow pattern… divergent** (or convergent)

First: how to **estimate divergence** of a given vector field?

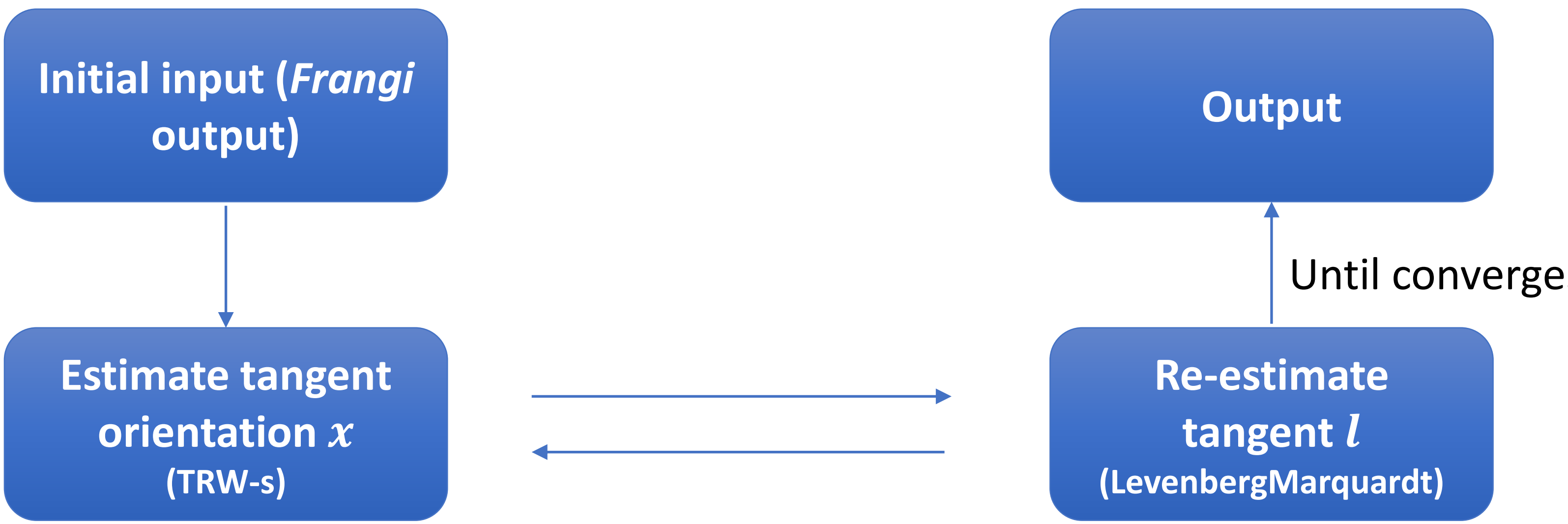



$$\nabla \bar{l}_{pq} = \int_{f^{\epsilon}_{pq}} \langle \bar{l}, n_s \rangle \, ds = \frac{\langle \bar{l}_q, pq \rangle - \langle \bar{l}_p, pq \rangle}{|pq|} \cdot |f_{pq}| + o(\epsilon)$$

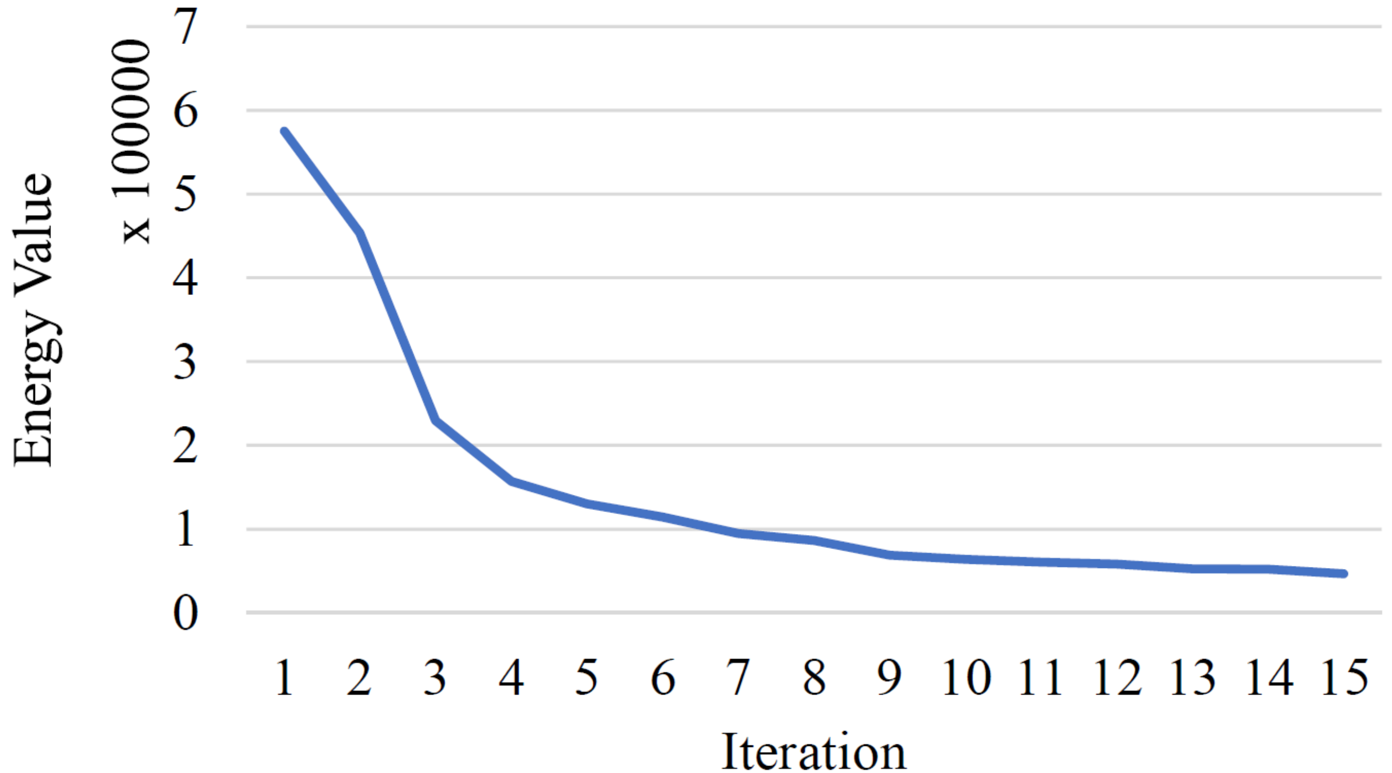
Divergence = Flux

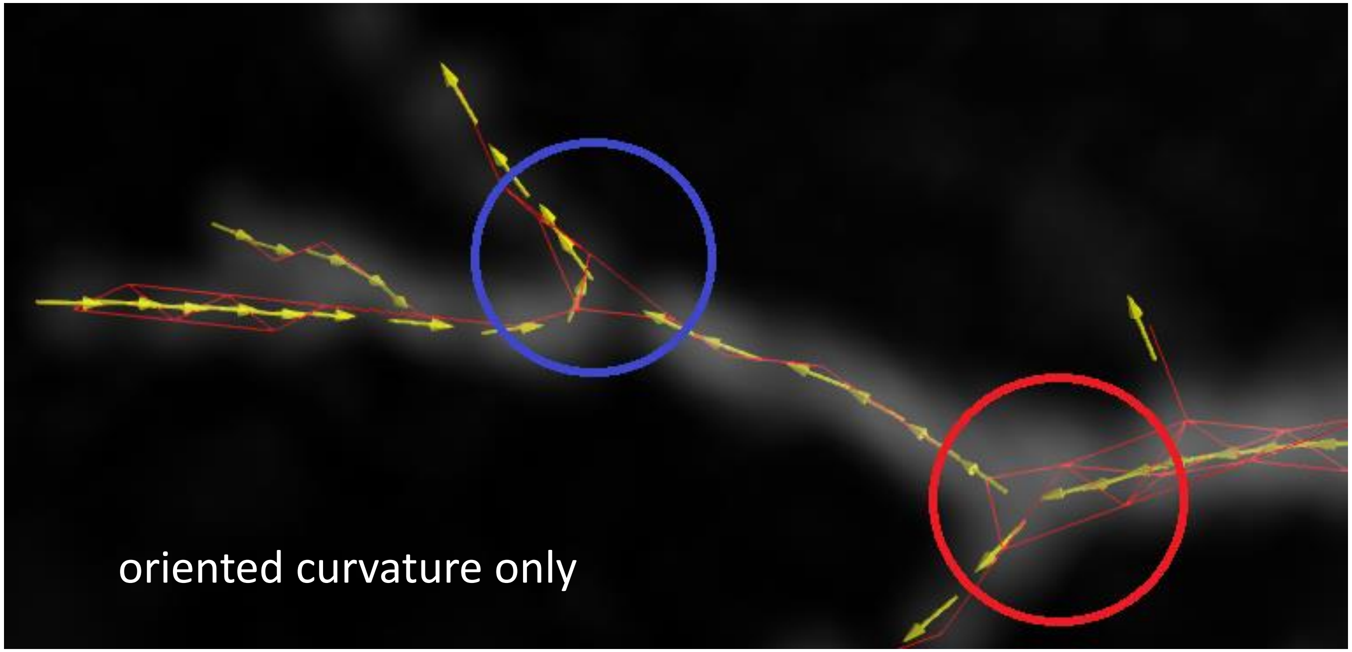
# **Joint energy** (curvature + divergence):

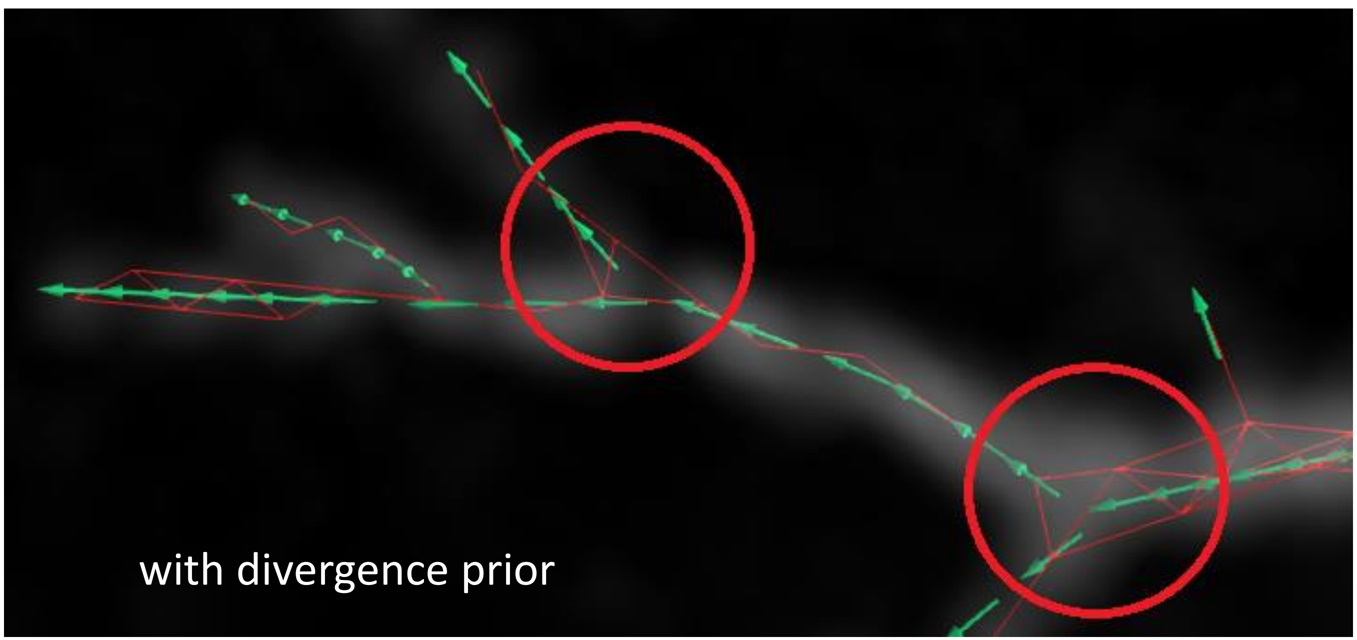
$$E(\bar{l}) = \sum_{p} \|\tilde{p} - \bar{l}_p\|^2 + \gamma \sum_{(p,q)\in \mathcal{N}} \bar{\kappa}_{pq}(\bar{l}_p, \bar{l}_q) + \lambda \sum_{(p,q)\in \mathcal{D}} (\nabla \bar{l}_{pq})^-$$




$$\bar{l}_p = x_p \cdot l_p$$

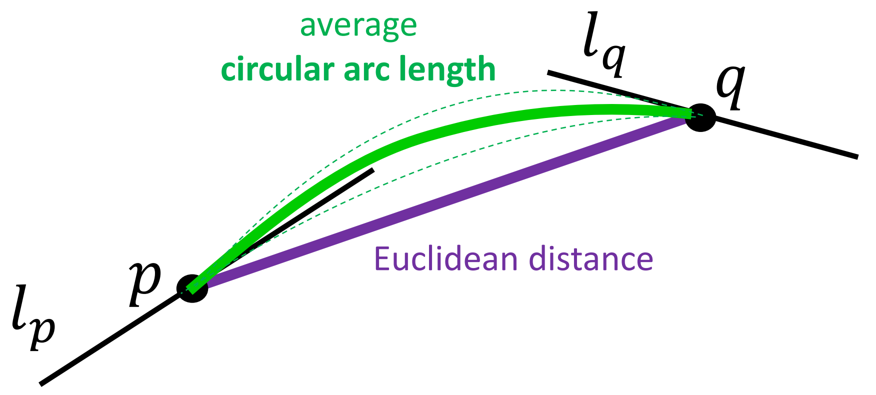

Energy Value x 1000000
7
6
5
4
3
2
1
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Iteration

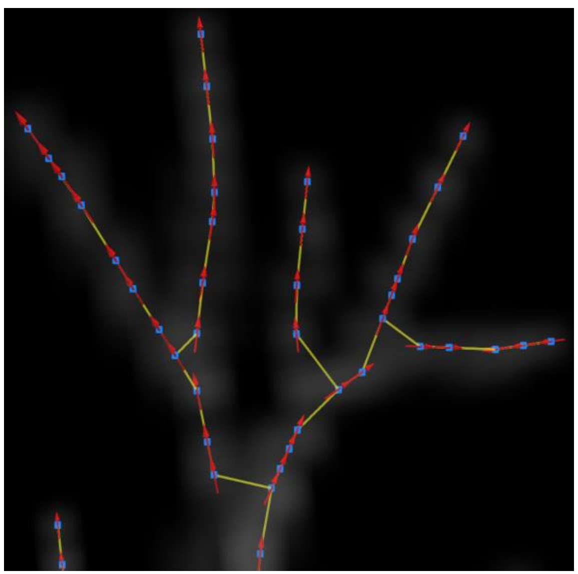

oriented curvature only


with divergence prior

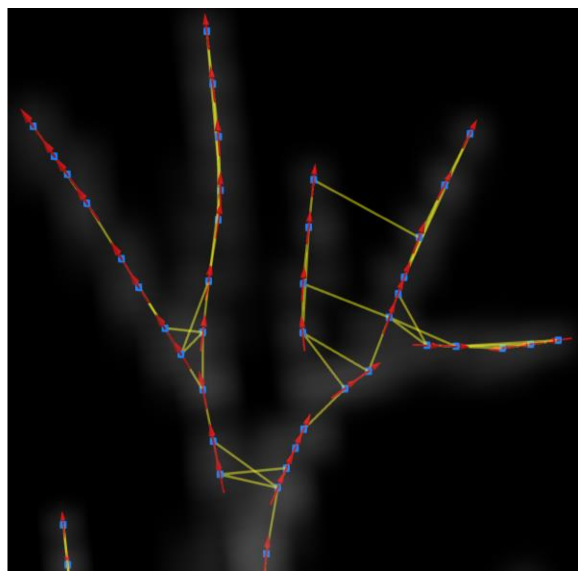
# Finally, constructing (standard) **undirected Tubular graph**

e.g. **KNN graph** connecting denoised points $\{p\}$ with **edge weights** based on…

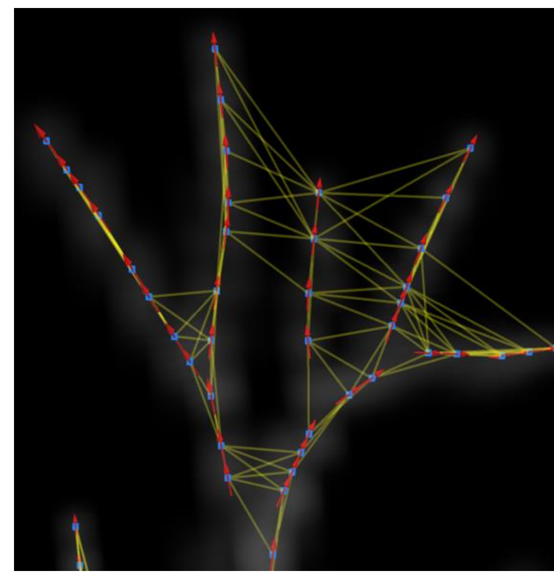


K=2



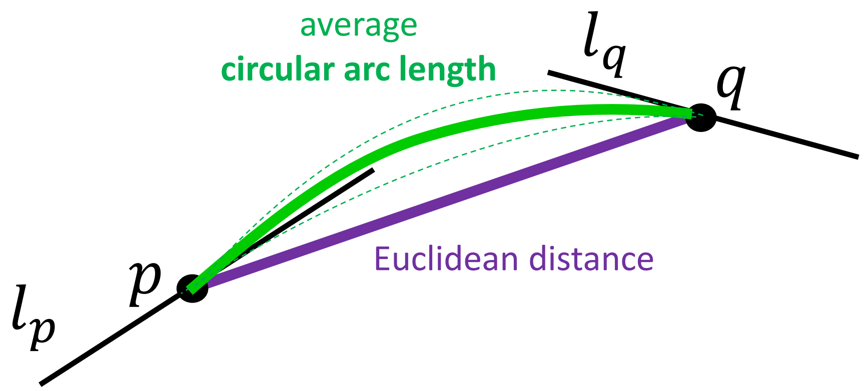
K=3

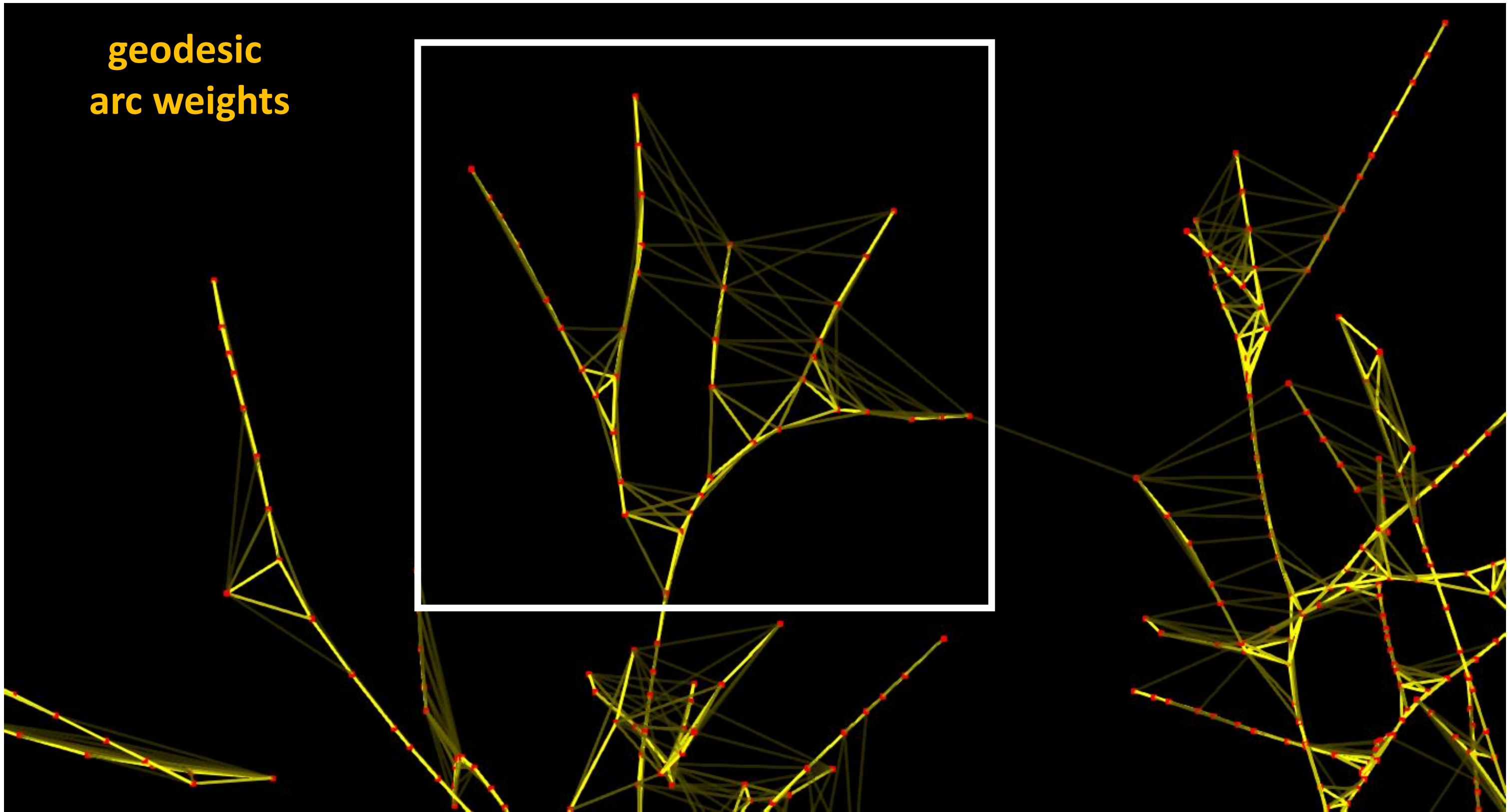


K=6

# Finally, constructing (standard) **undirected** **Tubular graph**

e.g. **KNN graph** connecting denoised points $\{p\}$ with **edge weights** based on…

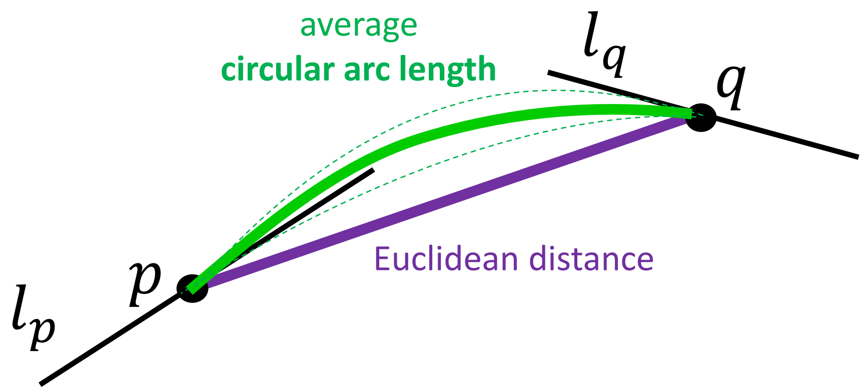


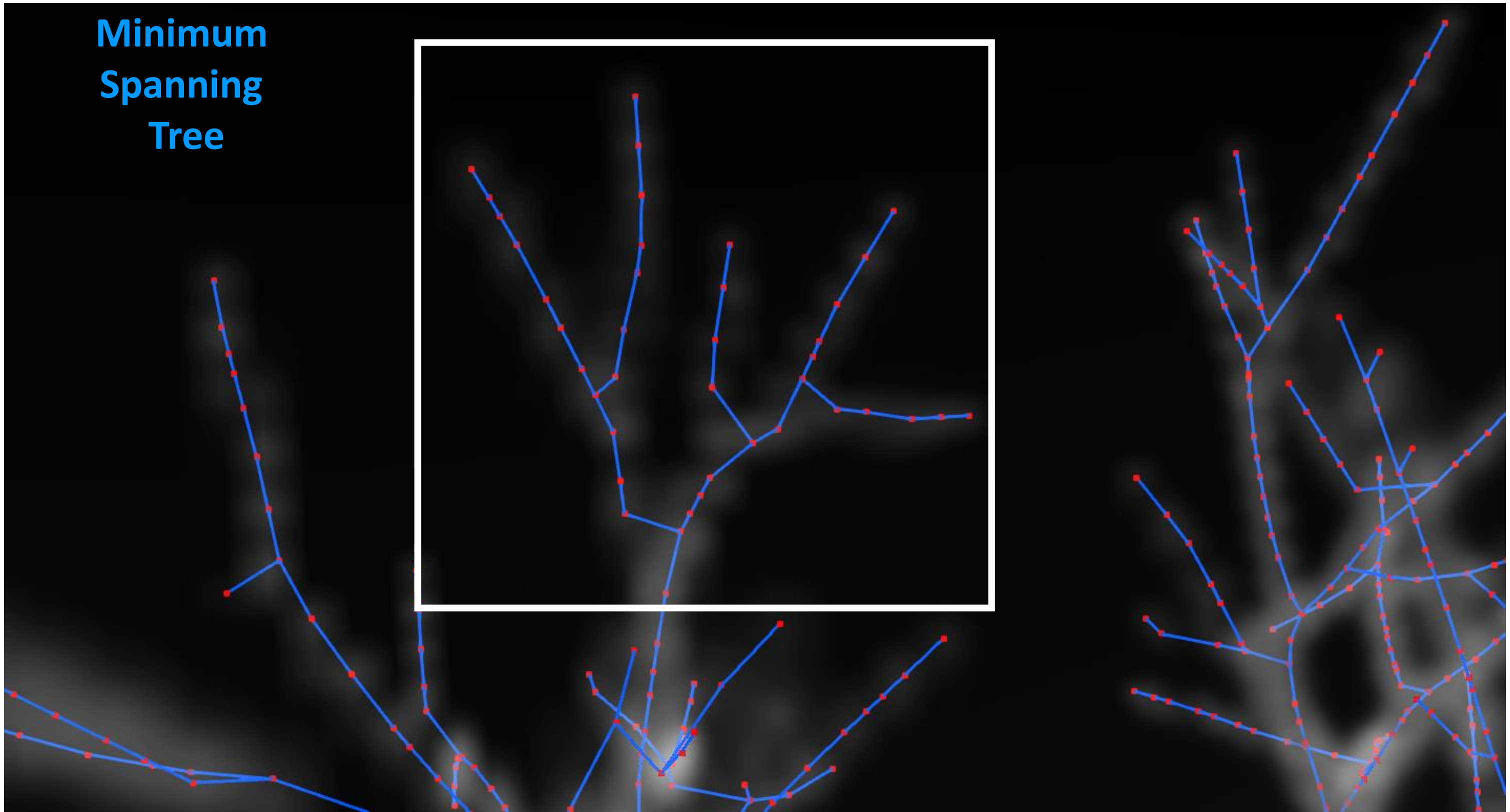




K=6

# Finally, constructing (standard) undirected Tubular graph

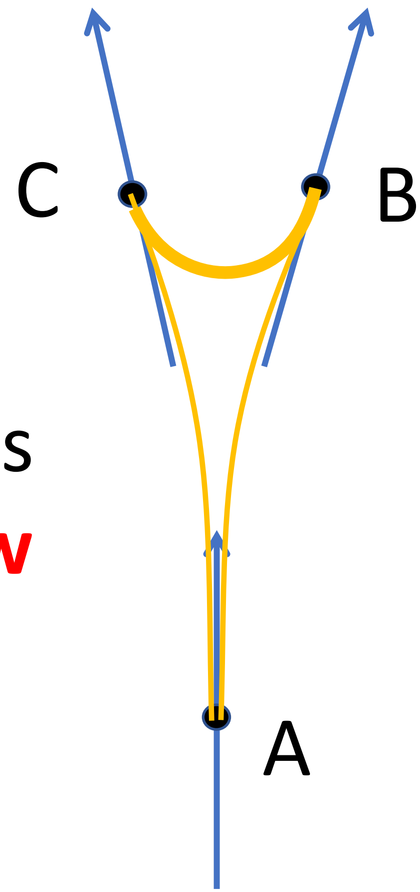
e.g. **KNN graph** connecting denoised points $\{p\}$ with **edge weights** based on…



# (again) **a problem at bifurcations**

$$\|AB\| + \|AC\| \;\; >> \;\; \|AB\| + \|BC\|$$



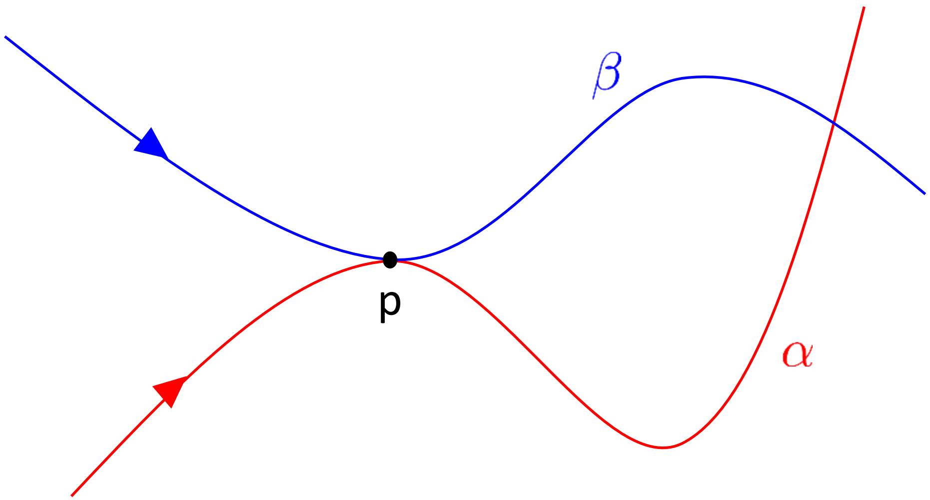

**shorter** arc BC gives **"non-smooth" flow**

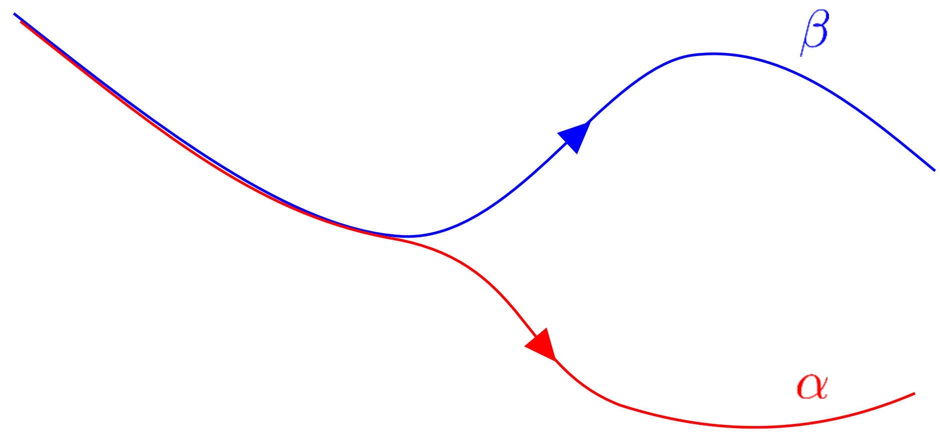
$\Rightarrow$ need constraint on **flow smoothness**

undirected arcs can't represent **flow orientation**

$\Rightarrow$ need **directed graph arcs**

# **confluence** of (oriented) continuous curves




**confluence of curves**
**at** (common) **point p**

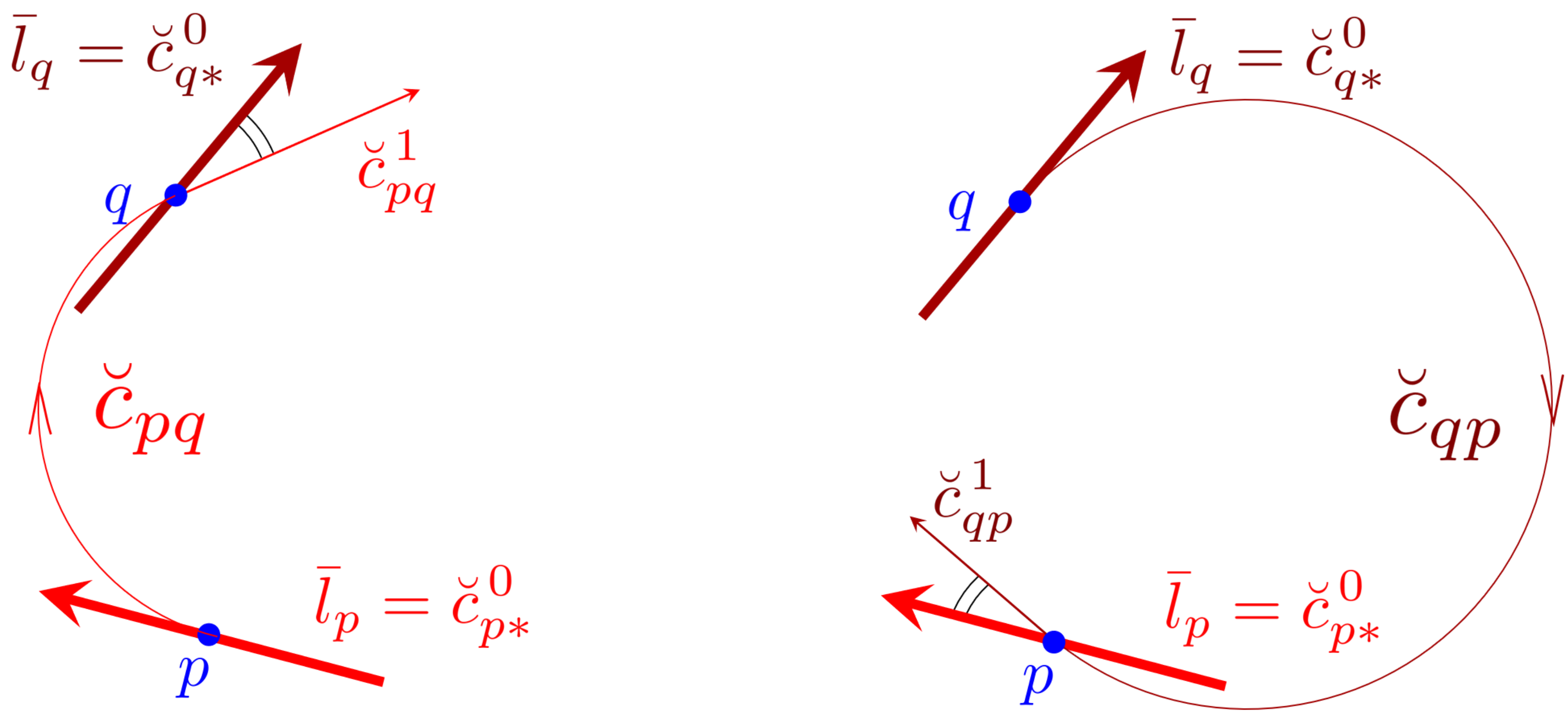
$$\alpha'(p) = \lambda\beta'(p)$$

$$\lambda > 0$$

**confluence of curves**
(at all common points)

# confluence of (directed) graph arcs

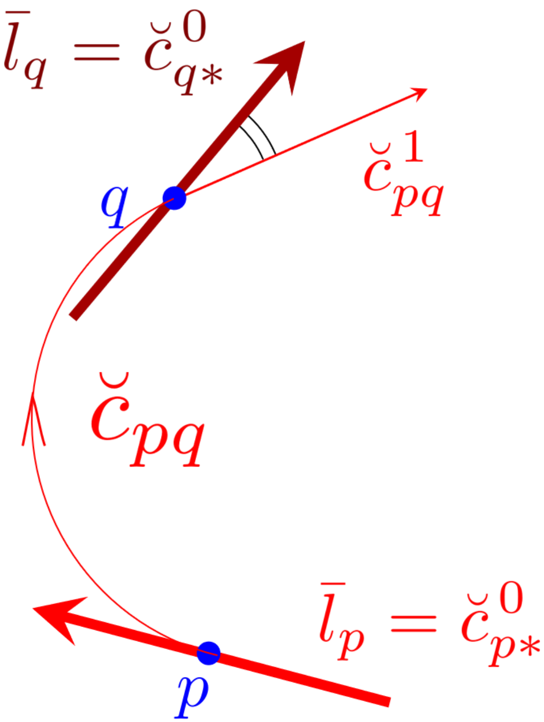
$$\angle(c^0_{q*}, c^1_{pq}) < \epsilon$$

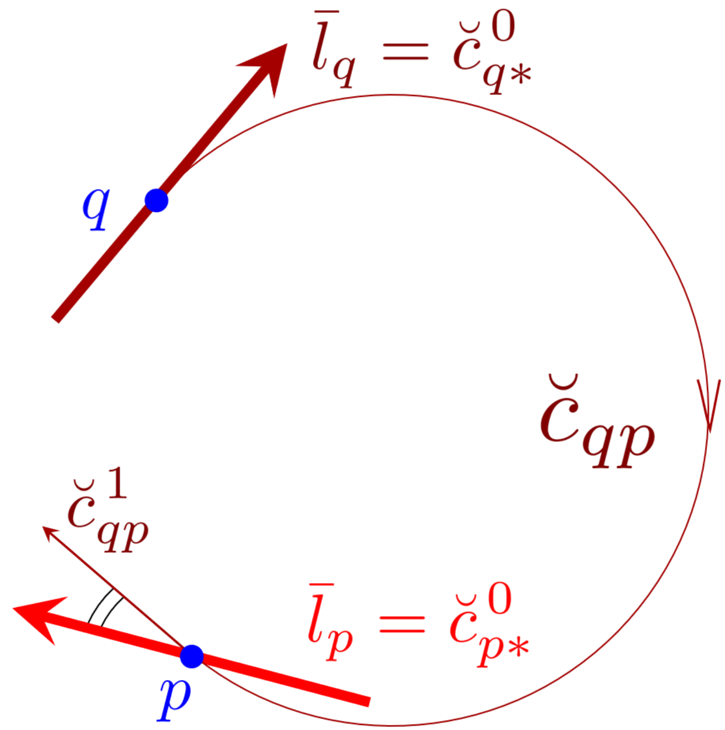



**Theorem:** $\angle(c^0_{q*}, c^1_{pq}) = \angle(c^0_{p*}, c^1_{qp})$ (valid in 3D)

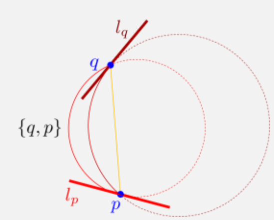
# **confluence** of (directed) graph arcs

$$\angle(c^0_{q*}, c^1_{pq}) < \epsilon$$

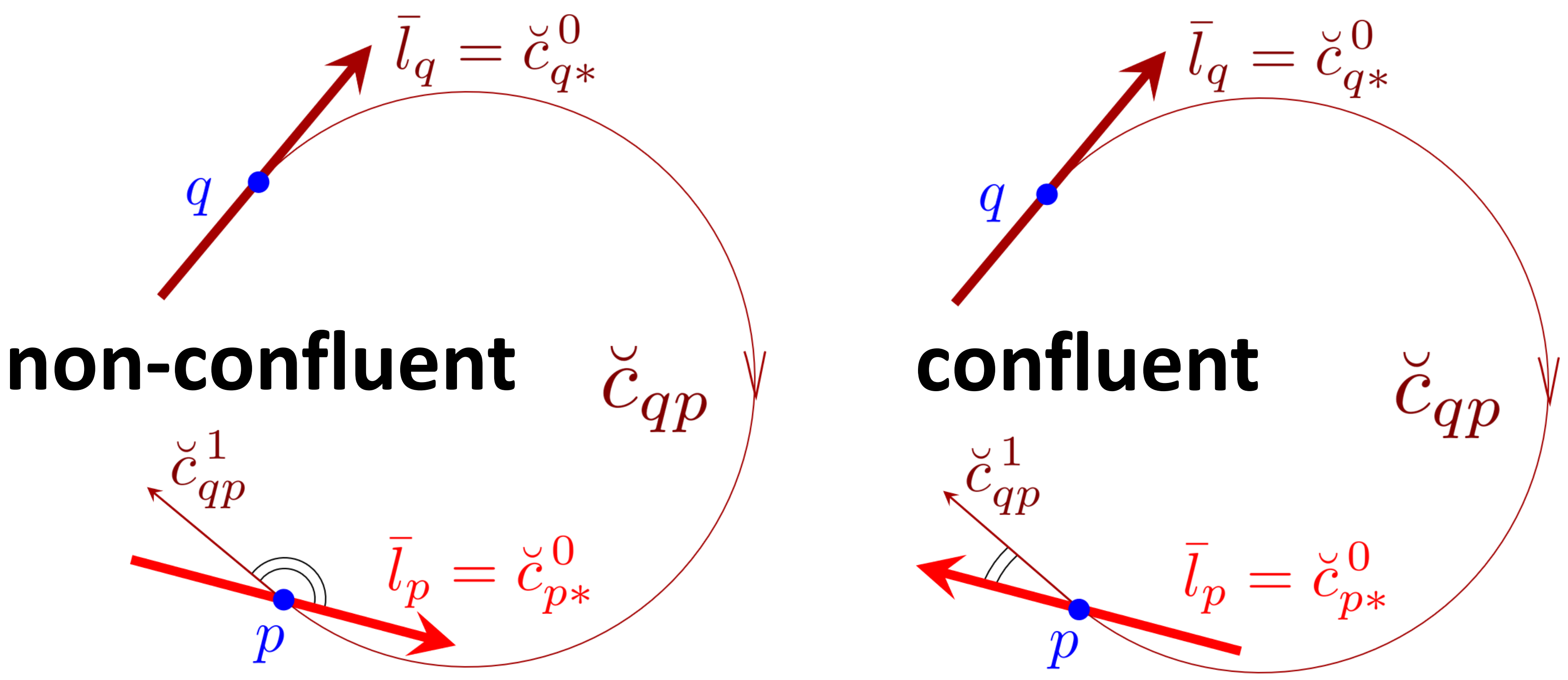


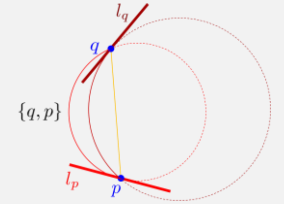

related to **co-circularity** (2D)
[Pierre Parent, Steven Zucker - TPAMI 1989]



# **confluence** depends on directions

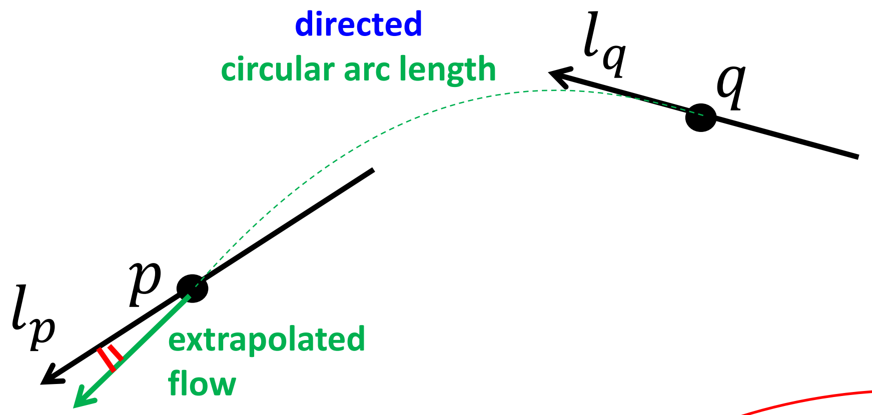


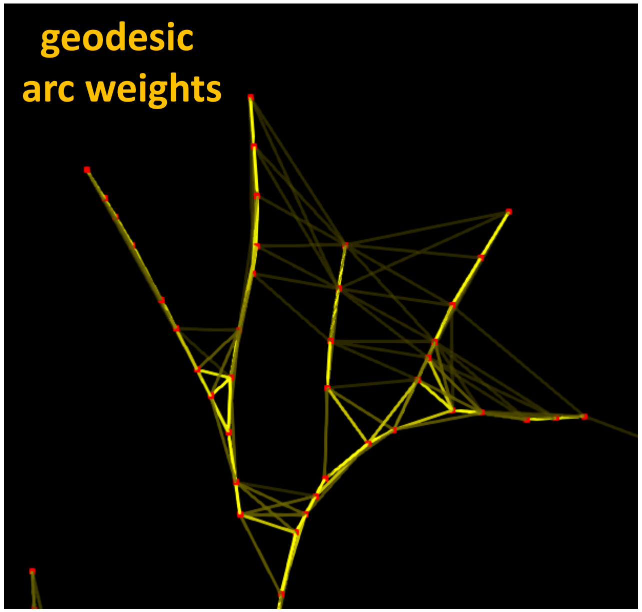

**(directed) generalization of co-circularity**
[Pierre Parent, Steven Zucker - TPAMI 1989]



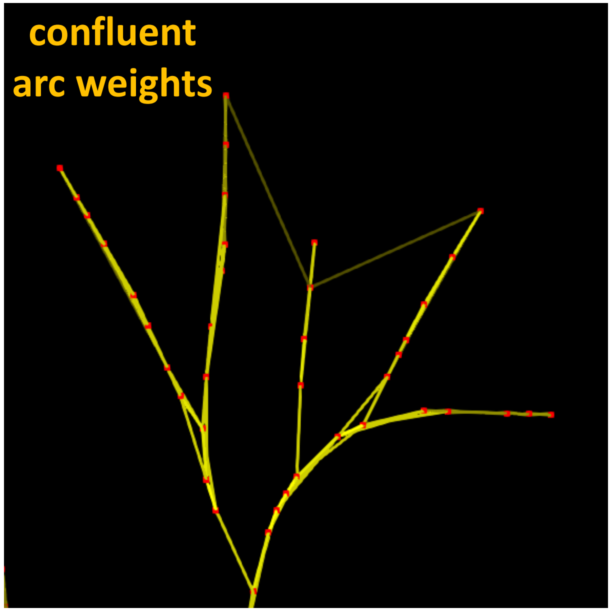
# Constructing confluent directed Tubular graph

e.g. **KNN graph** connecting denoised points $\{p\}$ with **edge weights** based on…



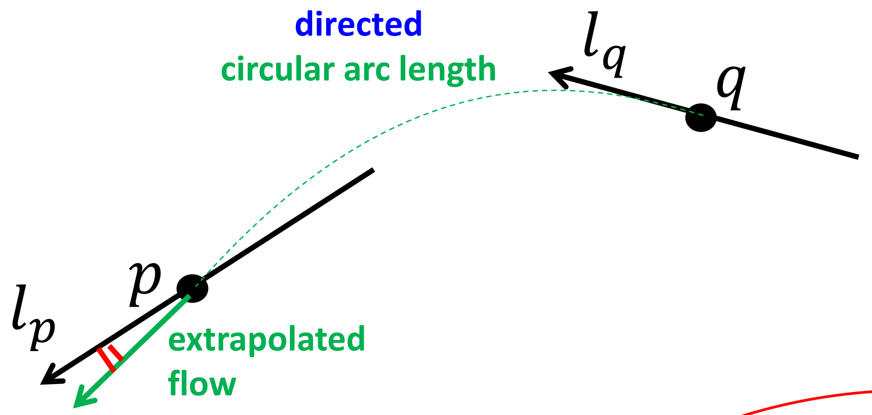

K=6

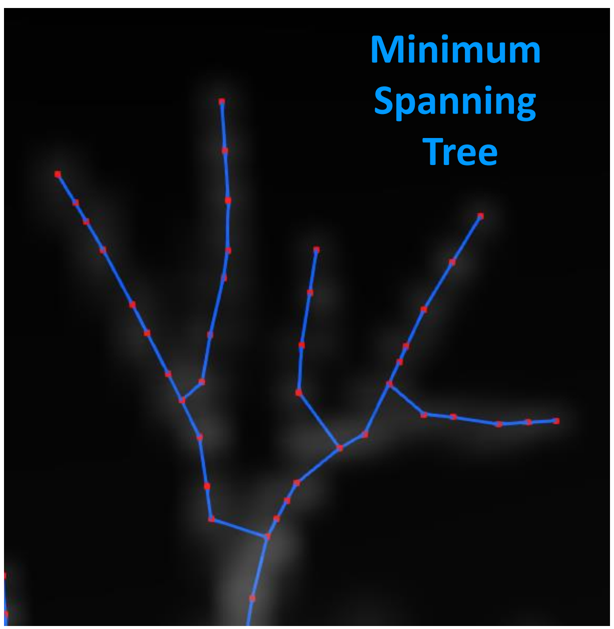


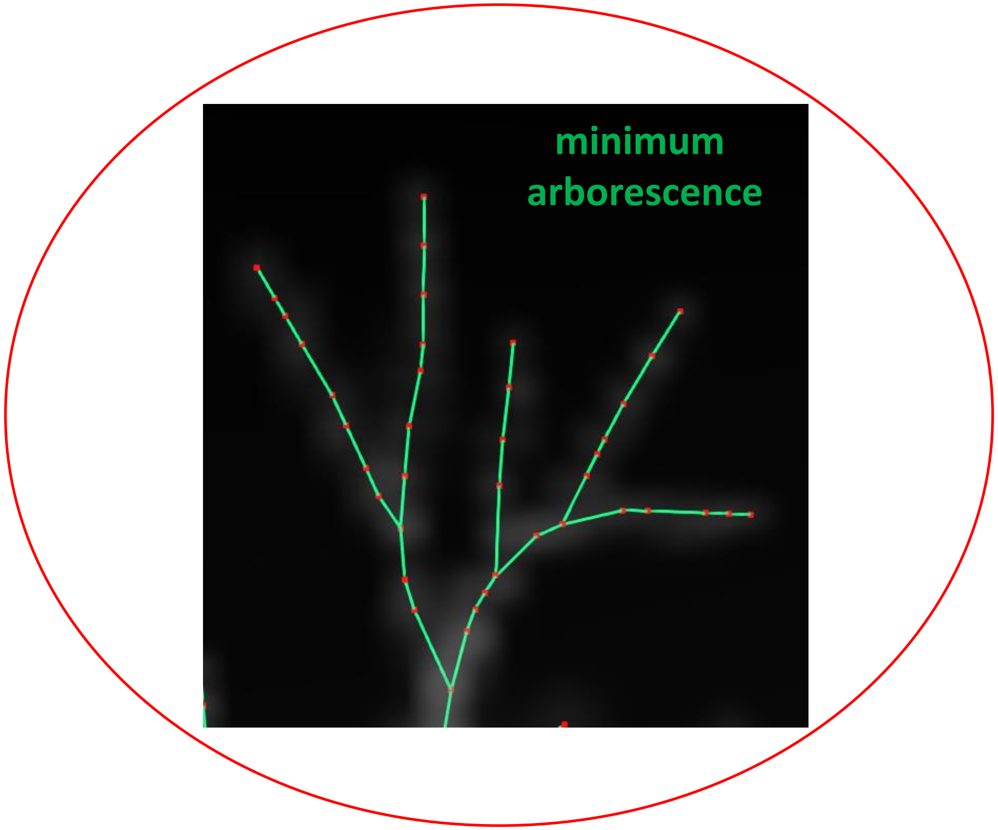

K=6

# Constructing confluent directed Tubular graph

e.g. **KNN graph** connecting denoised points $\{p\}$ with **edge weights** based on…



# Practicalities:

**For efficiency on large 3D data,**

- Frangi output (dominant eigen vector) **initializes** tangents $l_p$

- initial binary orientation variables $x_p$ are random

- for now, drop detection variables (too slow)

- loose thresholding, non-maxima suppression reduces the number of data points

# Our solution: curvature regularized centerline fitting

Raw data

↓

Tubular measure filtering
[Frangi, et al.]

↓

Non maxima suppression

↓

Flow estimation
(**curvature** + **divergence**)

↓

Minimum Arborescence on
(**confluent**) directed Tubular Graph

# Our solution: curvature regularized centerline fitting

Raw data

↓

**Tubular measure filtering**
[Frangi, et al.]

↓

Non maxima suppression

↓

Flow estimation
(**curvature** + **divergence**)

↓

Minimum Arborescence on
(**confluent**) directed Tubular Graph

# Our solution: curvature regularized centerline fitting

Raw data

↓

Tubular measure filtering
[Frangi, et al.]

↓

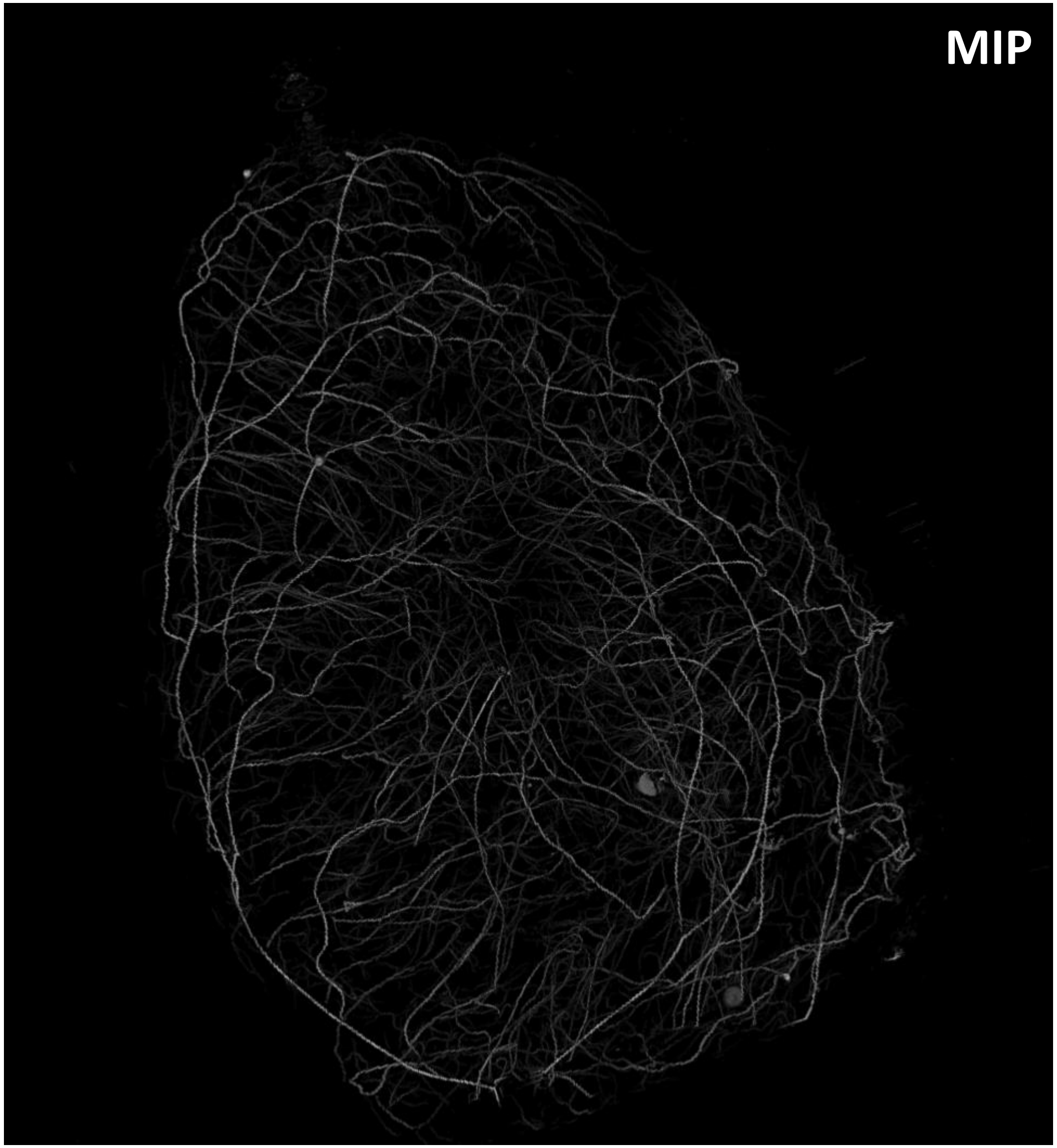
**Non maxima suppression**

↓

Flow estimation
(**curvature** + **divergence**)

↓

Minimum Arborescence on
(**confluent**) directed Tubular Graph



# Our solution: curvature regularized centerline fitting

Raw data

↓

Tubular measure filtering [Frangi, et al.]

↓

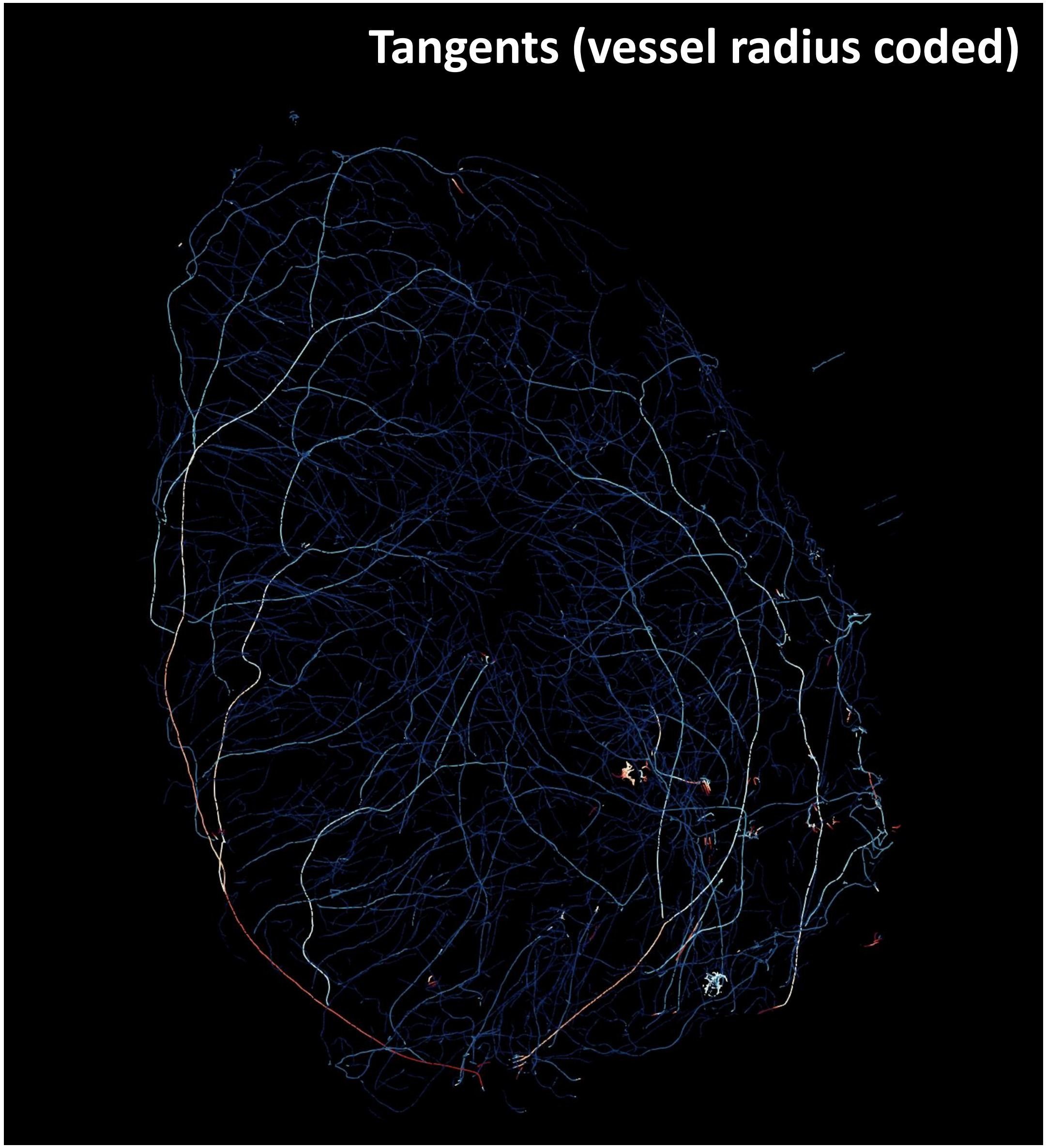
Non maxima suppression

↓

Flow estimation (**curvature** + **divergence**)

↓

Minimum Arborescence on (**confluent**) directed Tubular Graph



# Our solution: curvature regularized centerline fitting

Raw data

Tubular measure filtering [Frangi, et al.]

Non maxima suppression

Flow estimation (**curvature** + **divergence**)

Minimum Arborescence on (**confluent**) directed Tubular Graph