Reconstruction from 3D Point Clouds

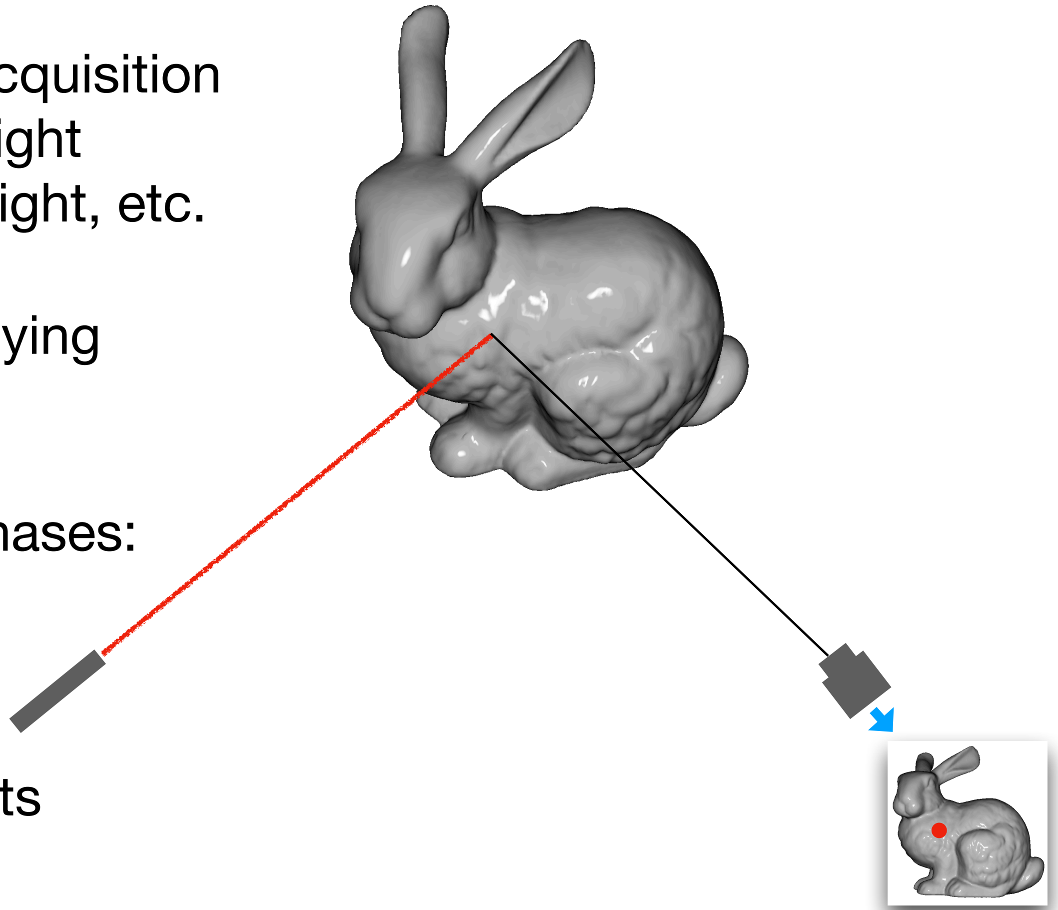
Subtle Deformations and Tiny Features

Andreas Bærentzen

Visual Computing : Department of Applied Mathematics and Computer Science : Technical University of Denmark

Optical Acquisition

- Numerous technologies exist for optical acquisition of 3D models: laser scanning, structured light scanning, structure from motion, time of flight, etc.
- Triangulation is the simple principle underlying many methods.
- Usually, reconstruction consists of two phases:
    - Reconstruction of points from images
    - Reconstruction of 3D surface from points

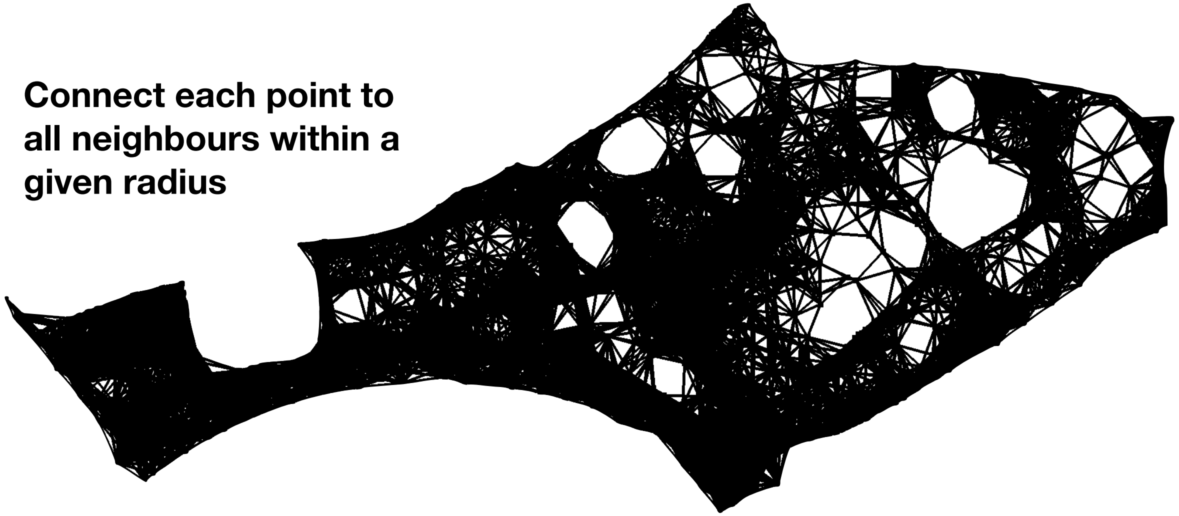


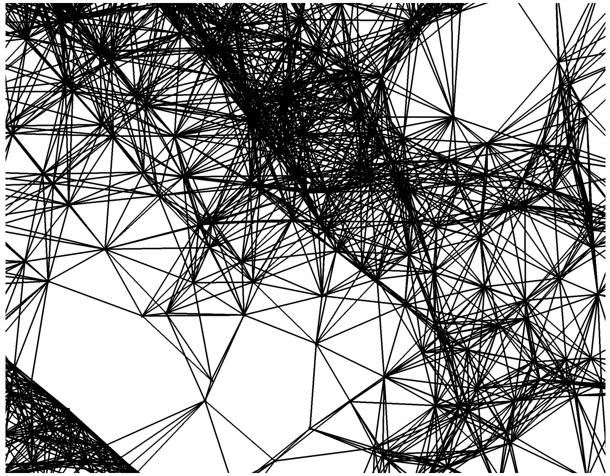
Overview

- 3D Reconstruction from point clouds
- Non-rigid registration through smoothing before (or after) reconstruction
- Using graph methods to capture very thin structures

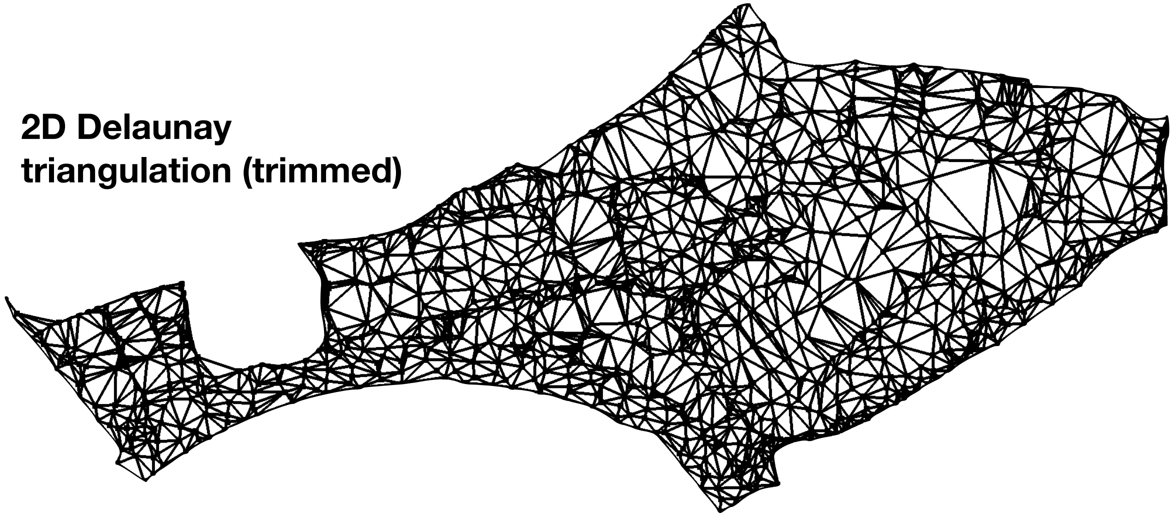
Neighborhood Graph vs 2D Delauny Triangulation

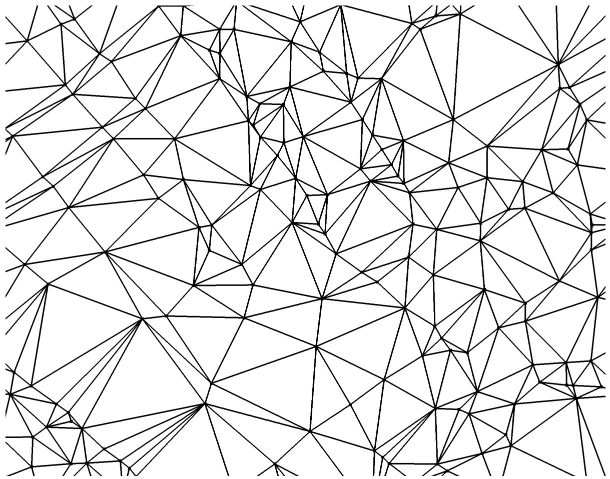
Connect each point to all neighbours within a given radius



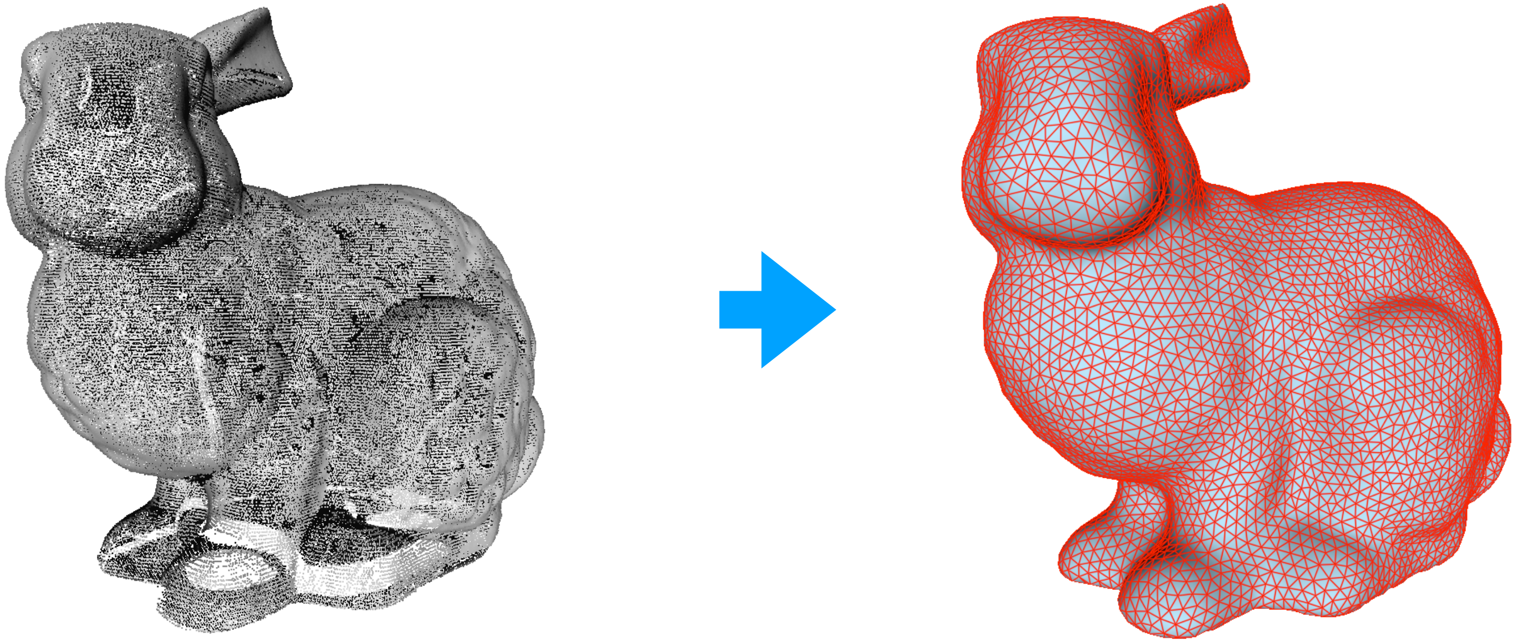


2D Delaunay triangulation (trimmed)





3D Reconstruction



Basically, this is the problem

Combinatorial 3D Reconstruction

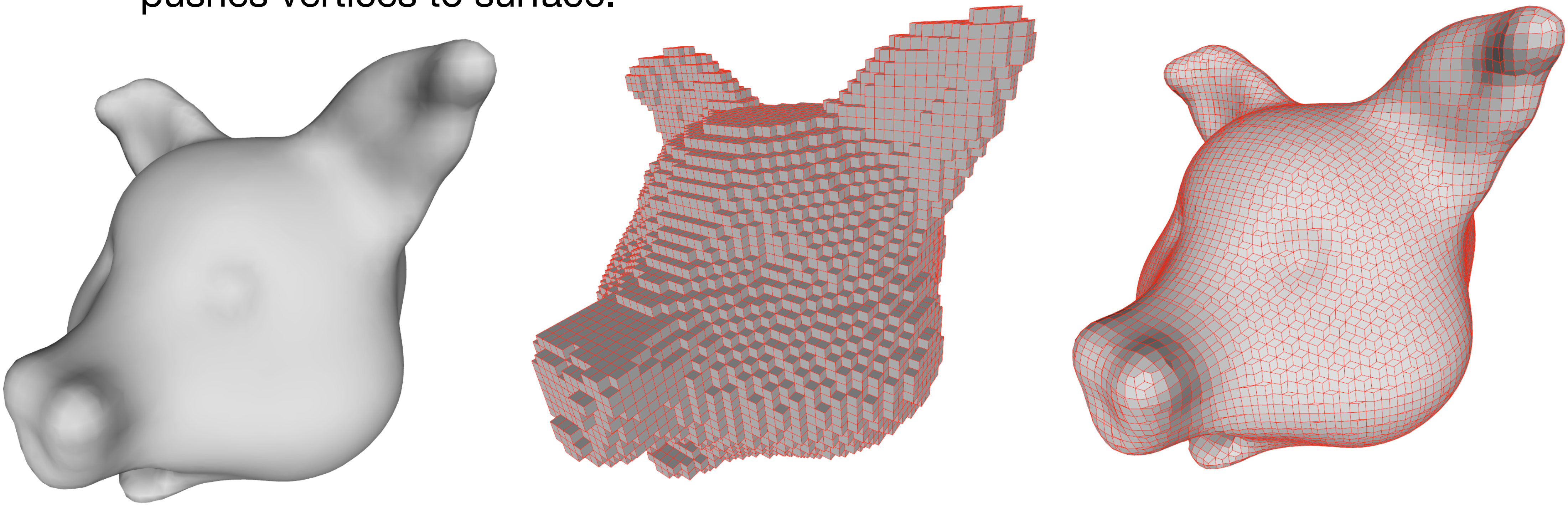
- Connect points to form triangles
- Caveat: we usually require a manifold triangle mesh:
    - All edges incident on at most two triangles
    - Triangles form a single loop around each vertex
- In practice, many methods start from the Delaunay Tetrahedralization:
    - [Cazals and Giesen 2006] provide a nice overview

Volumetric Reconstruction

- Find a characteristic function, $\chi : R^3 \rightarrow [0,1]$, such that for a point, $\mathbf{x}$,
  - $\chi(\mathbf{x}) < \tau$ if $\mathbf{x}$ is inside
  - $\chi(\mathbf{x}) > \tau$ if $\mathbf{x}$ is outside
  - $\chi(\mathbf{x}) = \tau$ if $\mathbf{x}$ is on the surface
- For some threshold, τ
- Usually, χ is stored in tabulated form in a volume, i.e. voxel grid.
- A manifold surface can now be found using iso-surface contouring

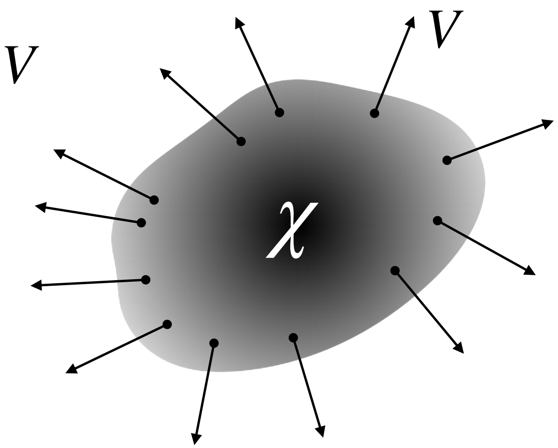
Iso-Surface Contouring

- Prefer dual contouring [Frisken 98, Nielson 04, AB et al. 12] to marching cubes.
- Dual contouring identifies faces separating inside from outside then pushes vertices to surface.

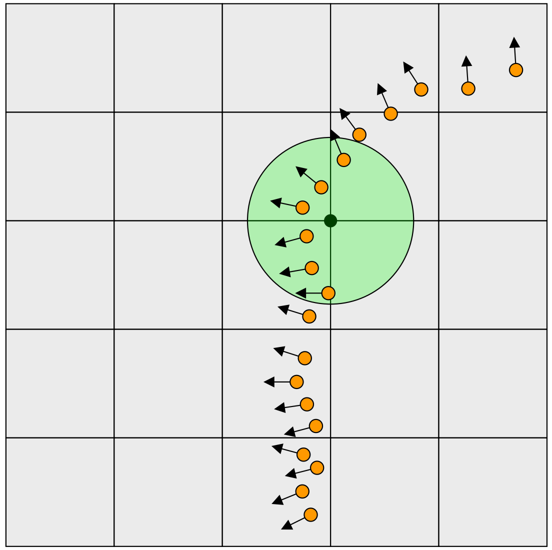


Poisson Reconstruction

- Estimated normals can be construed as a vector field V
- We want gradients of χ to match V, *i.e.* $\nabla\chi = V$
- Applying divergence, $\nabla \cdot \nabla\chi = \nabla \cdot V$, we arrive at
- a Poisson problem, $\Delta\chi = \nabla \cdot V$
- This insight led to the Poisson Reconstruction method [Kazhdan et al. 2006]. Several later improvements by the authors.

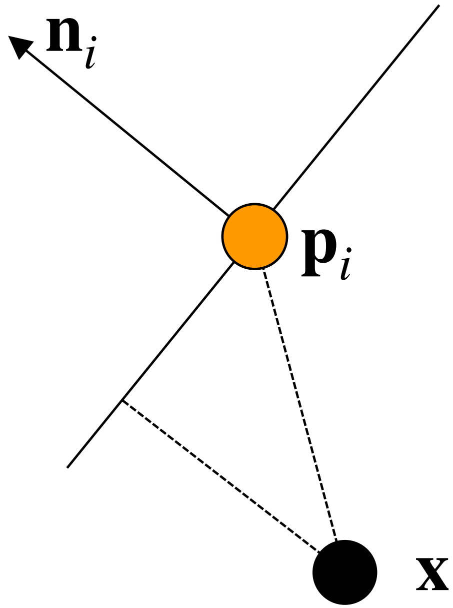


Volumetric Reconstruction



- χ does not need to be globally defined
- This leads to

$$\chi(\mathbf{x}) = \frac{\sum_i w_i(\mathbf{x})\phi_i(\mathbf{x})}{\sum_i w_i(\mathbf{x})}$$



- $w_i = \exp(-\|\mathbf{x} - \mathbf{p}_i\|^2/\sigma^2)$ clamped to limited support, and $\phi_i = \mathbf{n}_i \cdot (\mathbf{x} - \mathbf{p}_i)$
- Similar to FSSR [Fuhrmann and Goesele 2014]

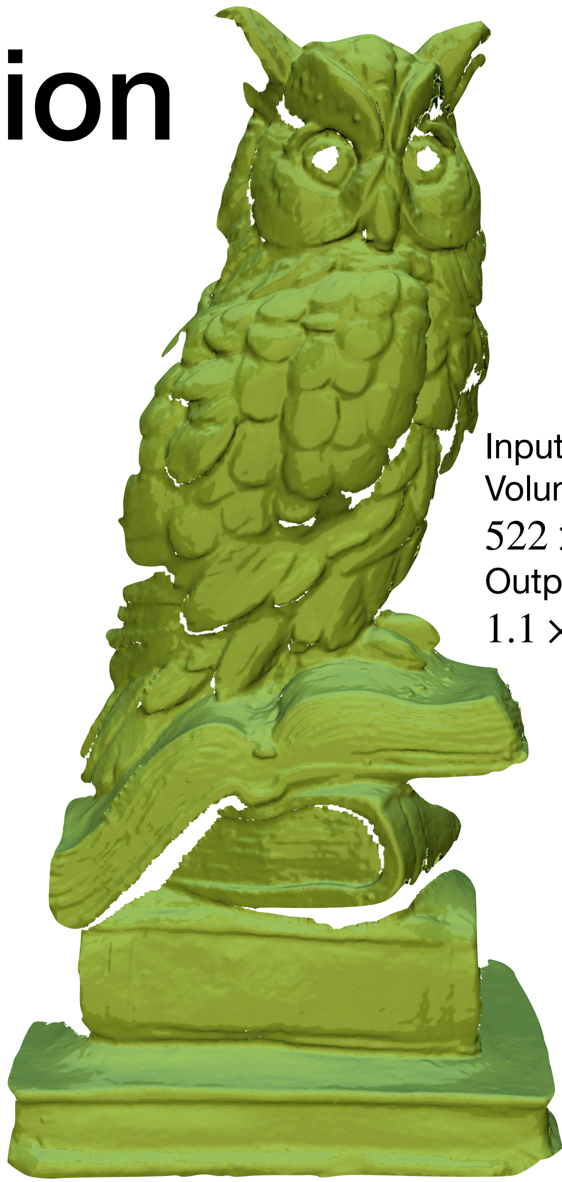
Volumetric Reconstruction

Input `points` **and estimate** `norms` **(usually by local plane fitting)**
Create voxel grid, `V`**, and weight grid,** `V_w`
Execute code below:

```
for i in range(len(points)):
    p,n = points[i], norms[i]
    for vxl in support(dim, world2grid(p)):
        x = grid2world(vxl)
        d = n @ (x-p)
        w = Gaussian(x-p)
        V[vxl] += d * w
        V_w[vxl] += w

for vxl in np.ndindex(dim):
    V[vxl] = V[vxl] / V_w[vxl]
```

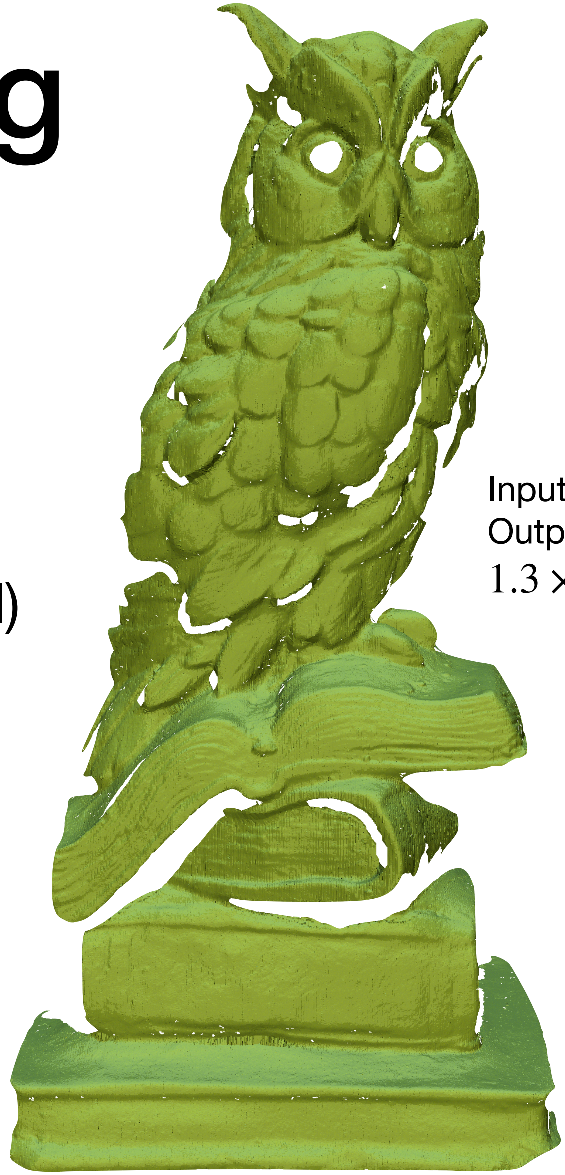
Now, compute the mesh by iso-surface contouring.



Input: 666750 points
Volume dimensions:
$522 \times 1023 \times 751$
Output:
1.1×10^6 triangles

Scale Space Meshing

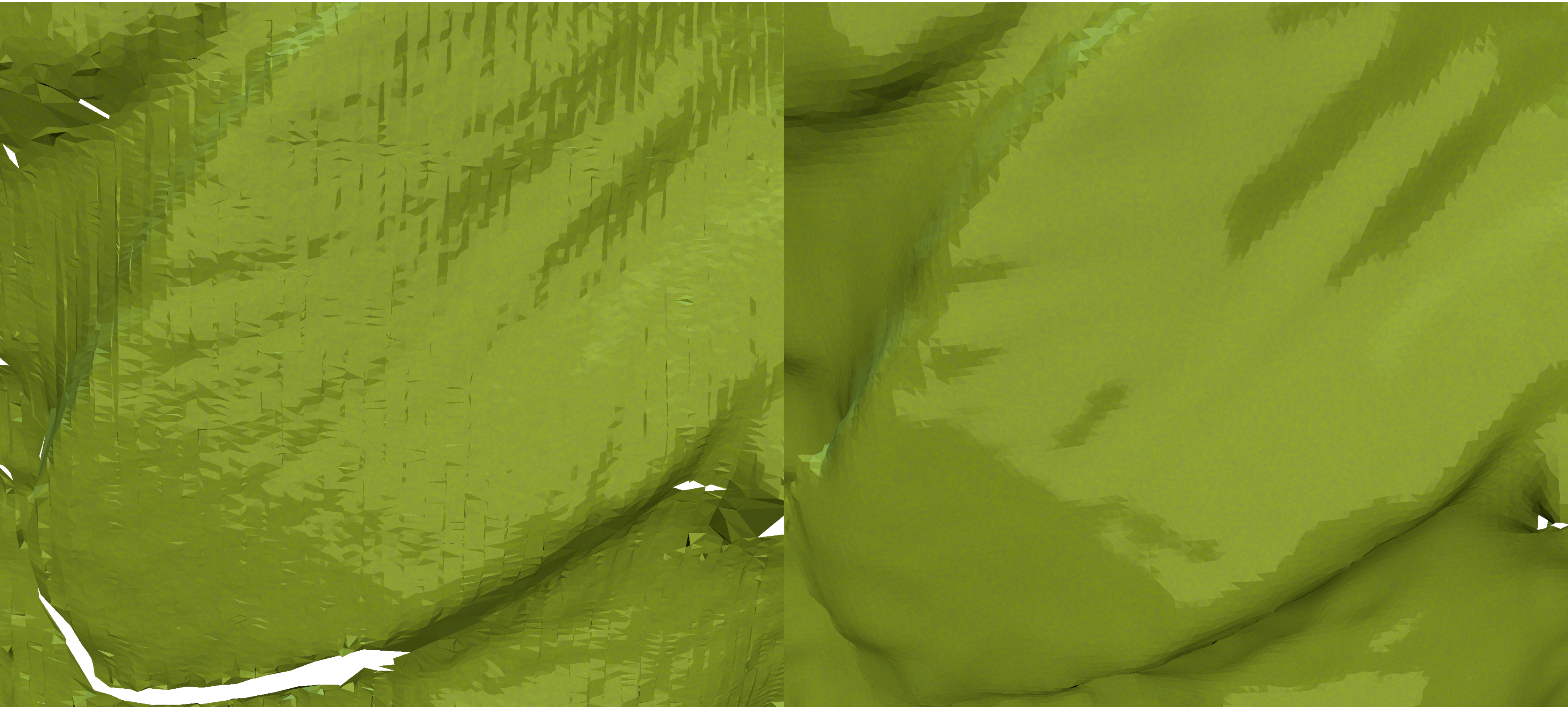
- Method due to Digne et al. [2011]
- Observation: smoothing point cloud leads to easier reconstruction problem (more points used)
- Reconstruction using Ball Pivot Algorithm [Bernardini et al. 1999].
- After reconstruction, we can put the points back



Input: 666750 points
Output:
1.3×10^6 triangles

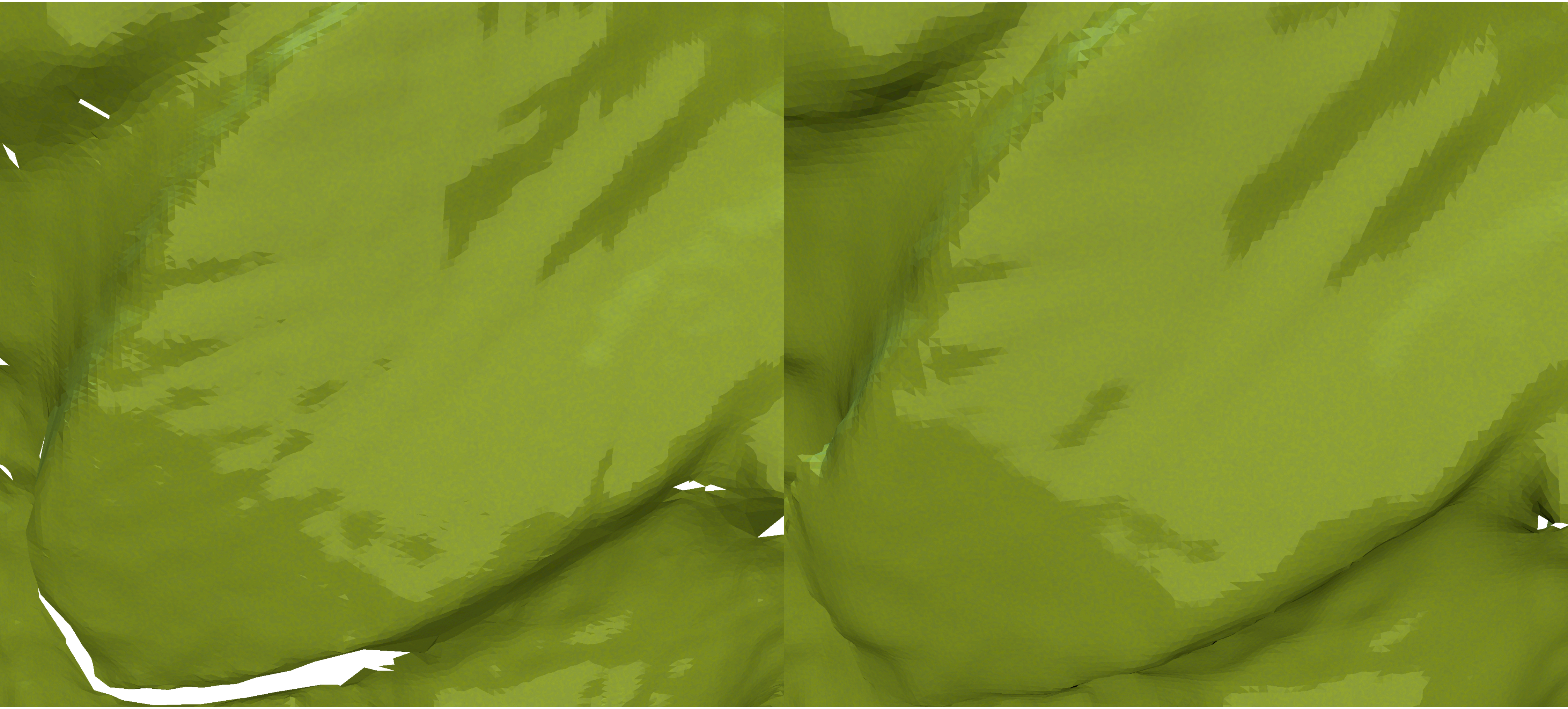
Mesh Comparison

Mesh Comparison

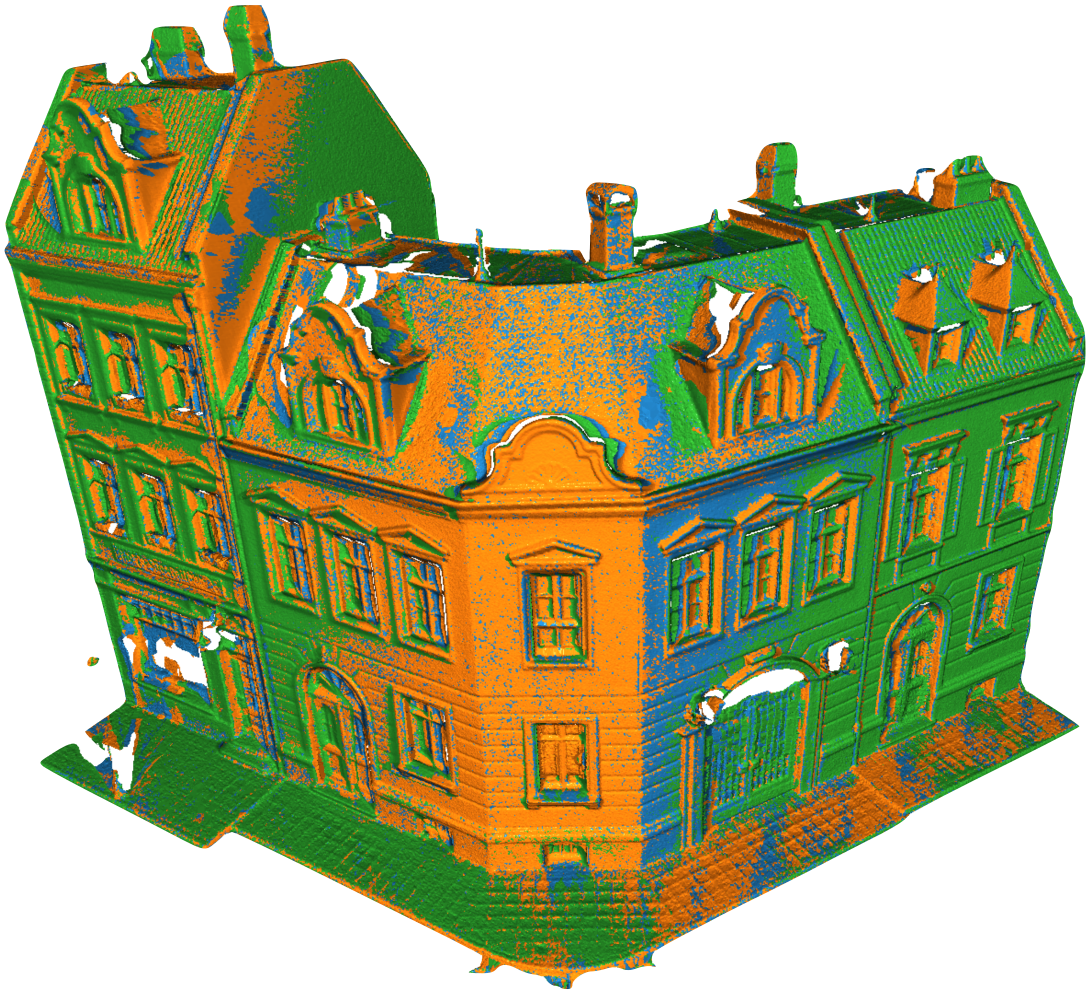


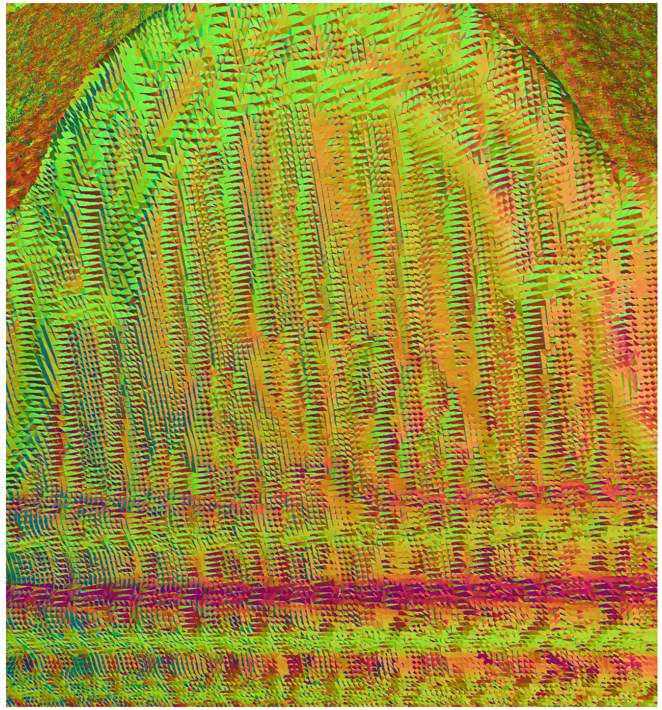
Mesh Comparison

10 iterations of Taubin smoothing



The Need for Non-Rigid Registration





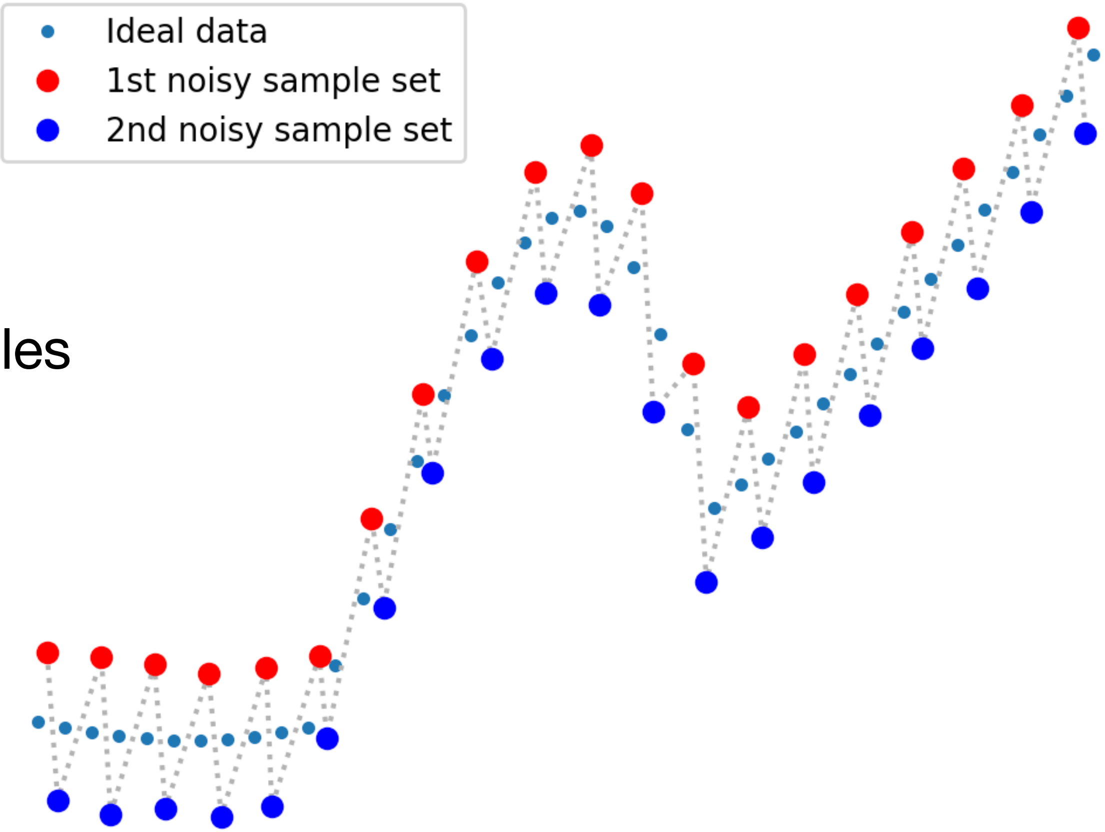
Mesh reconstructed using Co3Ne
[Boltcheva, D. and Lévy, B. 2017]

Non-Rigid Registration through Smoothing

- Optical scans of the same object usually differ by a slight deformation:
  - Our camera lens is not a pinhole: some lens distortion unmodeled
  - Many materials are slightly translucent
  - Some objects are deformable
- We could remove the noise by smoothing, but that would wash out features
- Our scheme: smooth the reconstructed mesh, but keeping subscans as rigid as possible

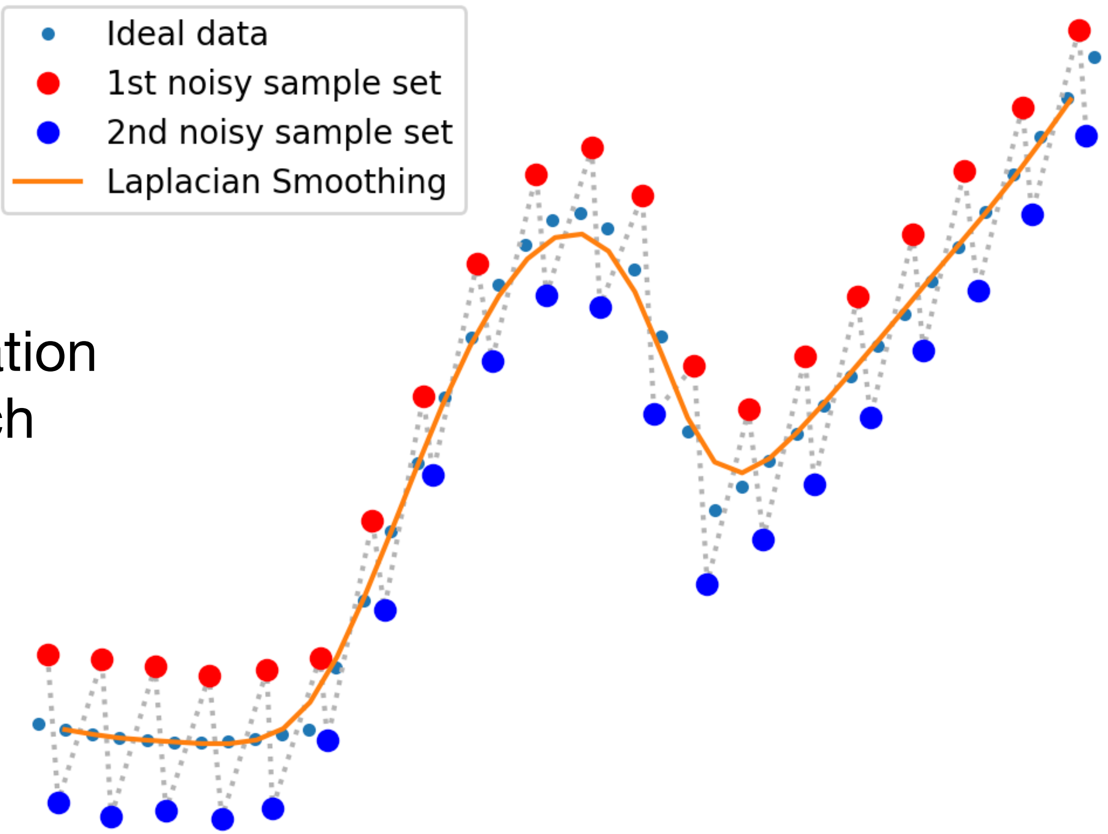
Toy Example

- Dotted line connects samples from two data sets: hence the high frequency noise



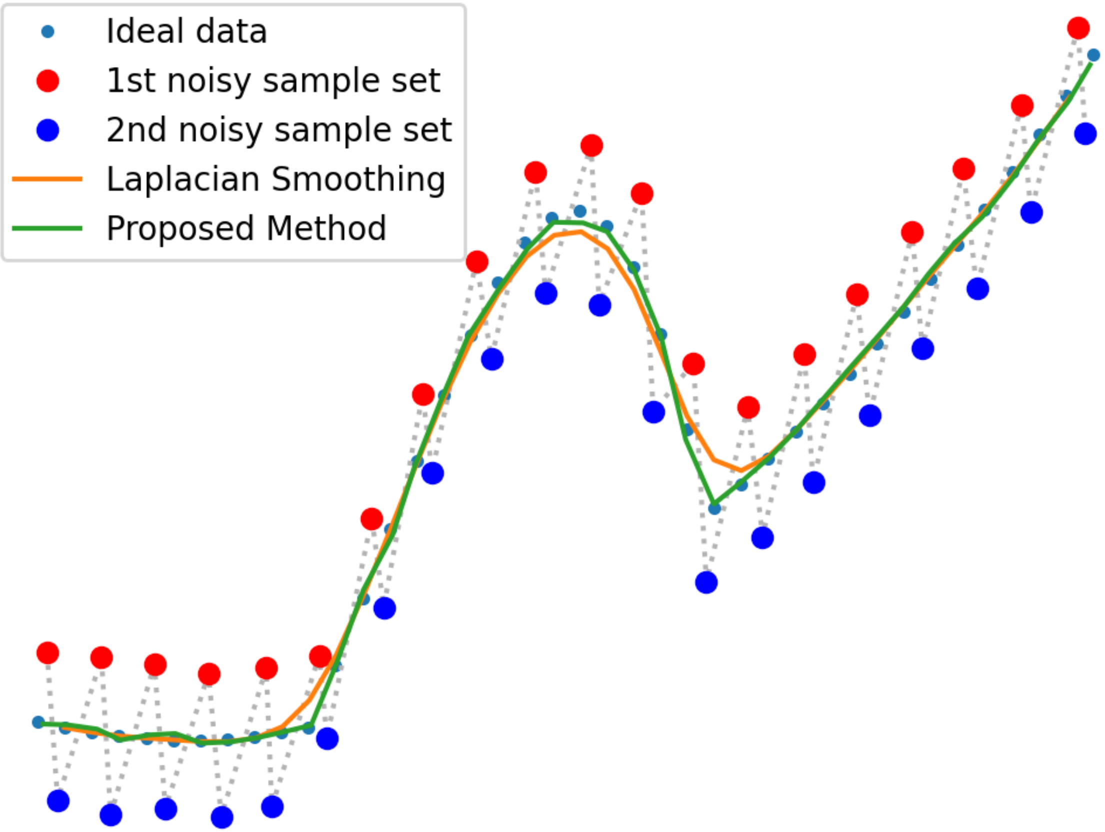
Toy Example

- The Laplacian is a deformation vector we can apply to each point



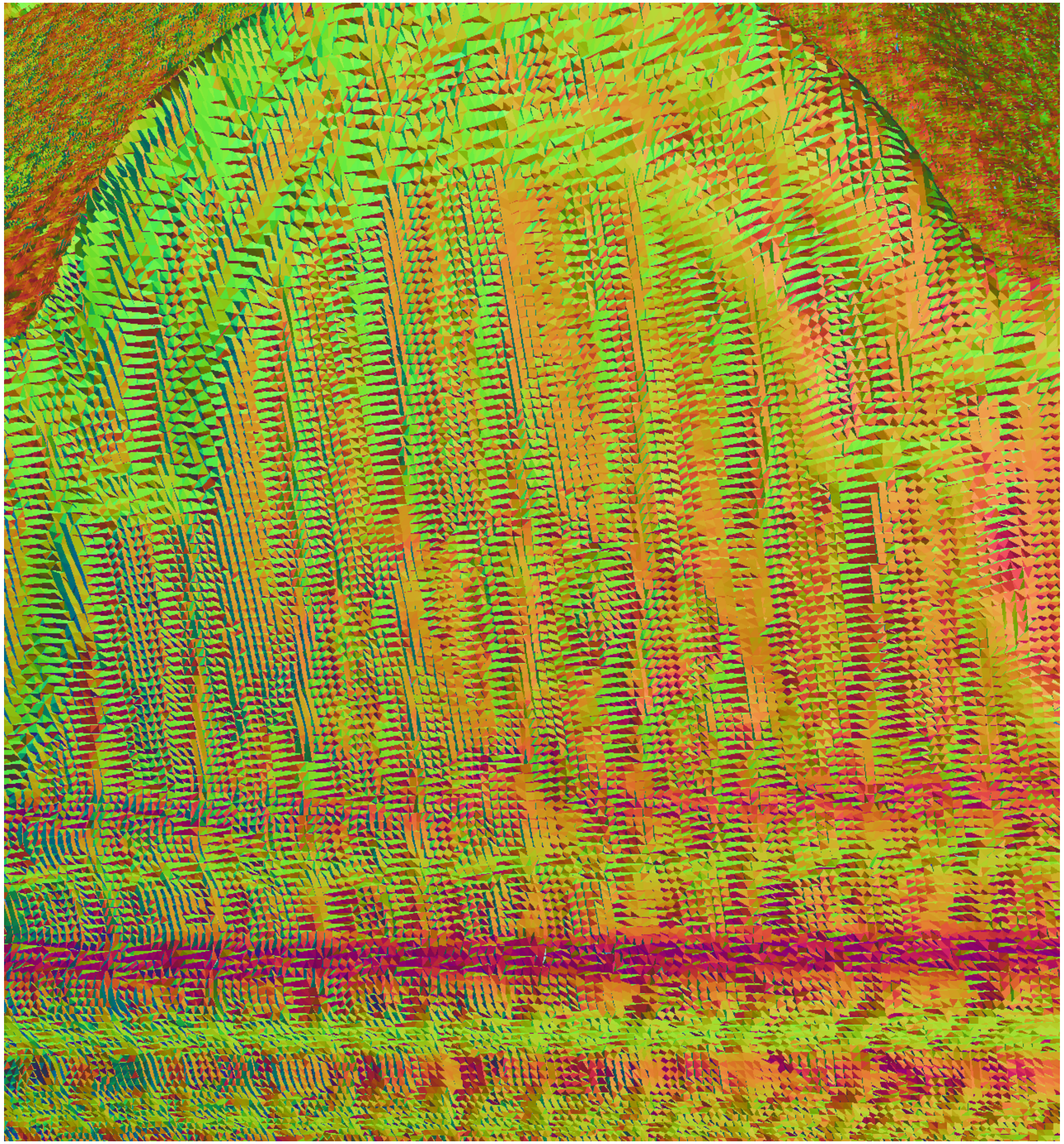
Toy Example

- Smooth the deformation before applying it, yields green curve

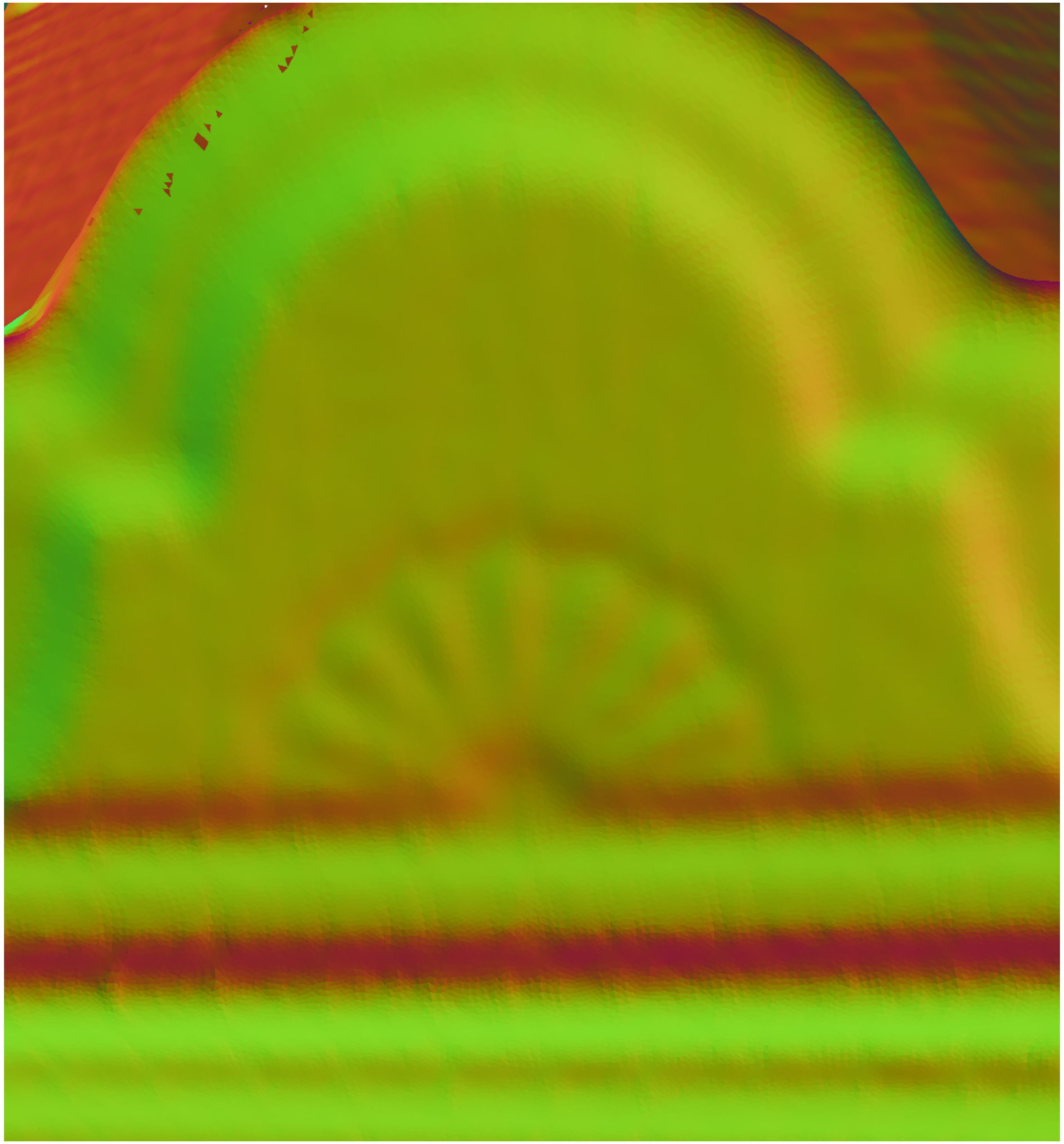


Non-Rigid Registration through Smoothing

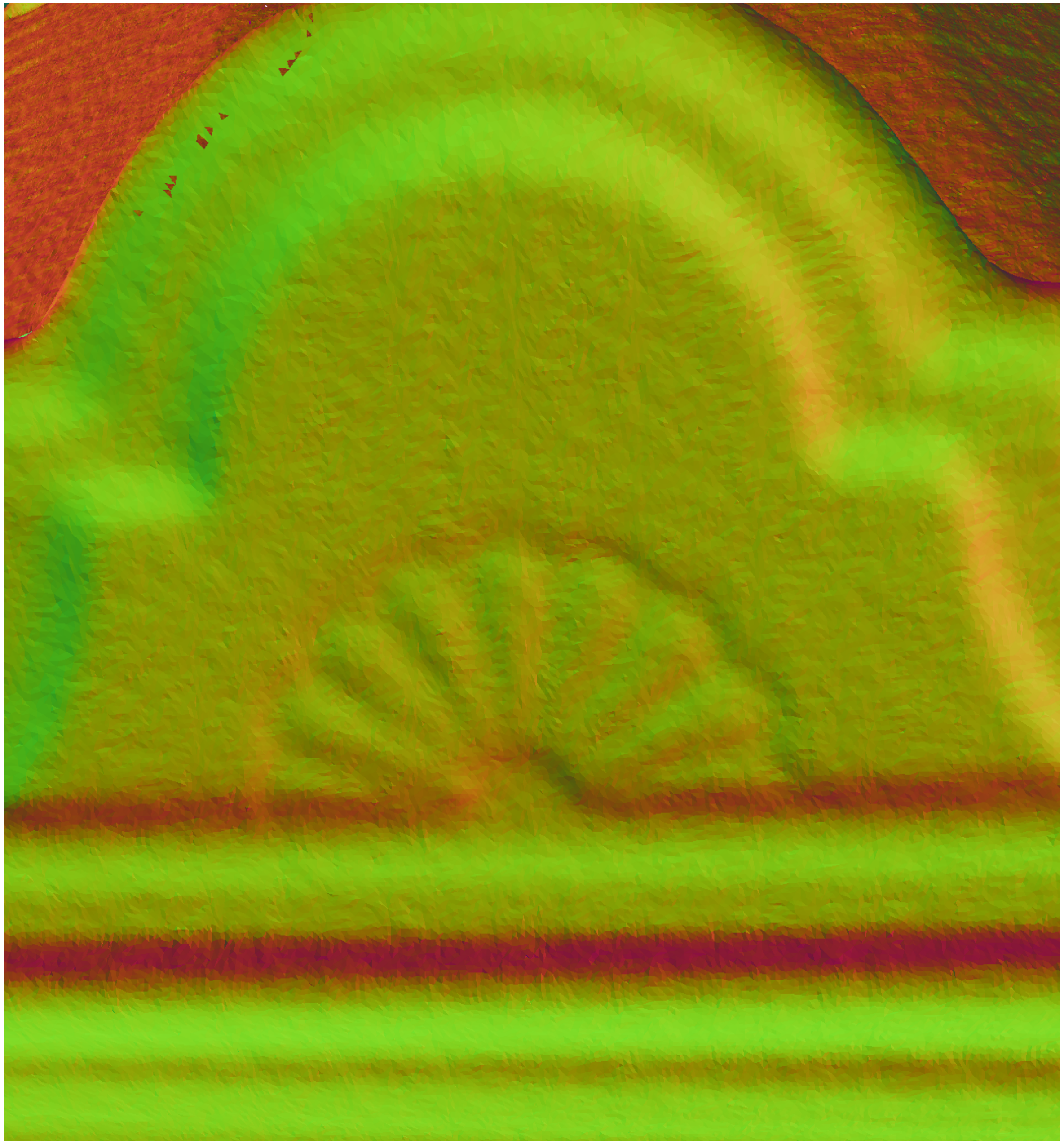
- We developed a method for Non-Rigid Alignment (NRA) which allows for different sources of displacement [Gawrilowicz and AB 2014]
- The energy is $E = \|\mathbf{D} - \mathbf{T}\|^2 + \alpha\|\mathbf{MD}\|^2$, where
  - $\mathbf{T}$ are target displacements
  - $\mathbf{D}$ the displacements we seek
  - $\mathbf{M}$ incidence matrix for sub scans
  - α is a parameter controlling stiffness

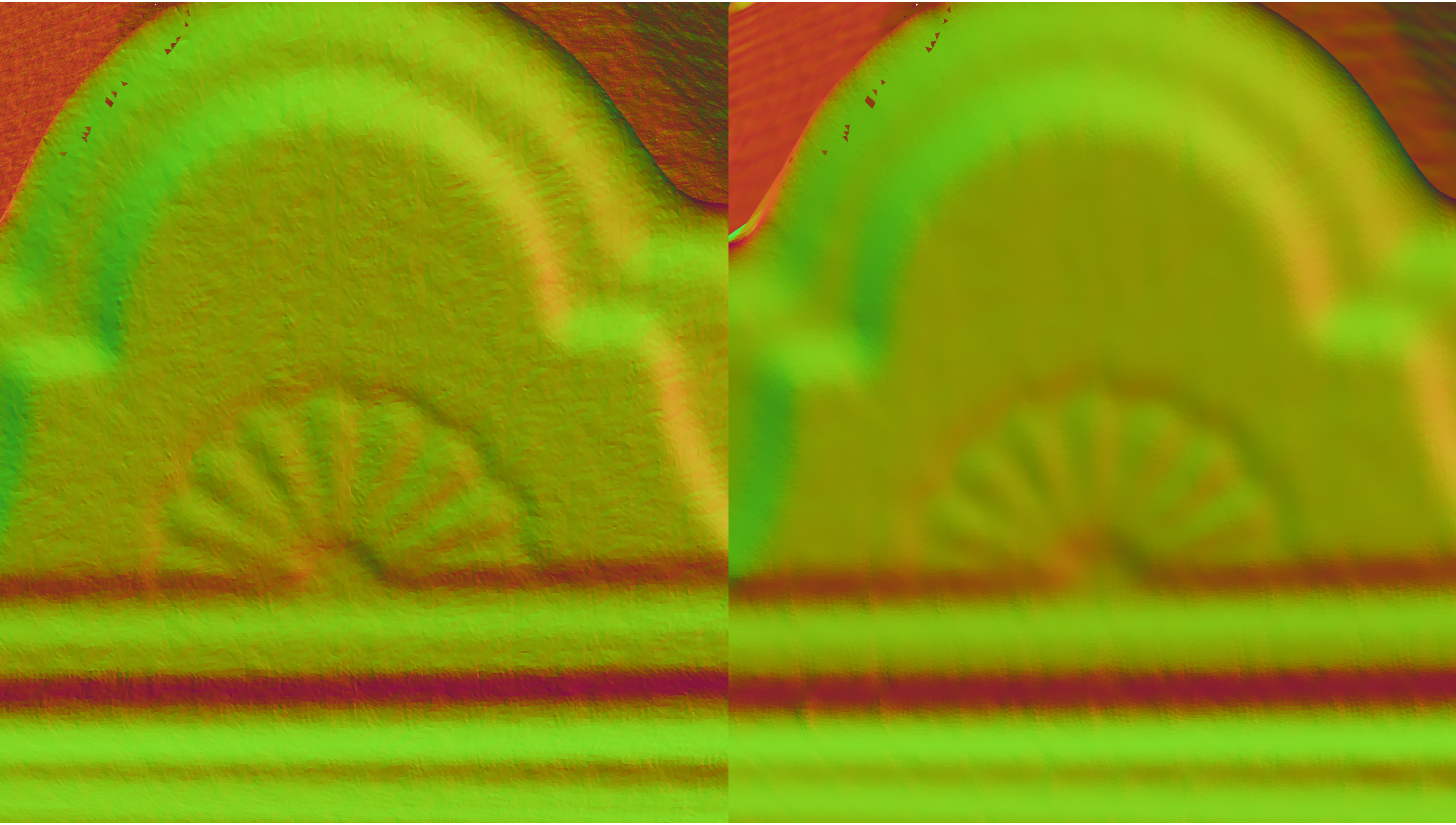


Smoothing

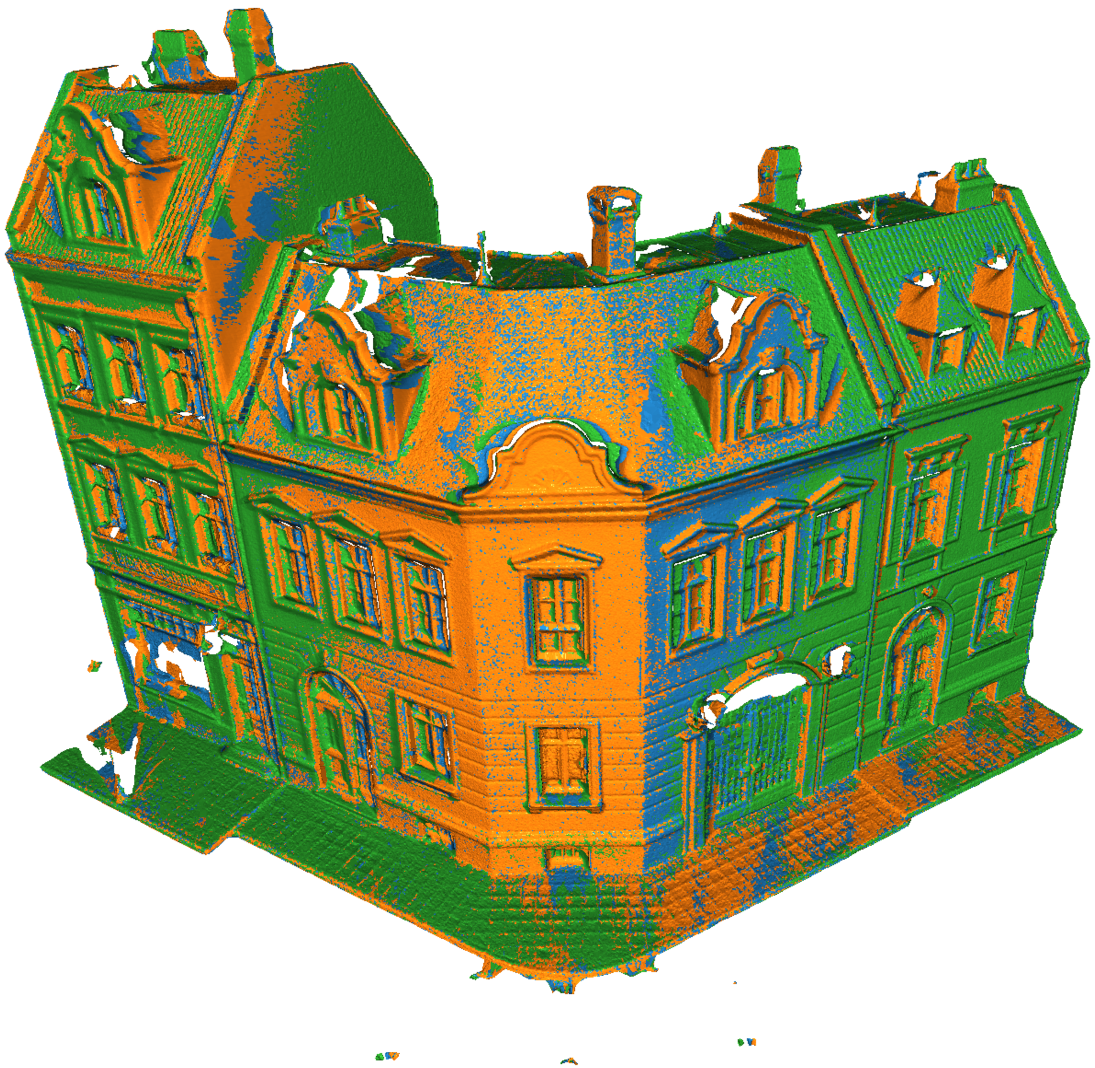


NRA

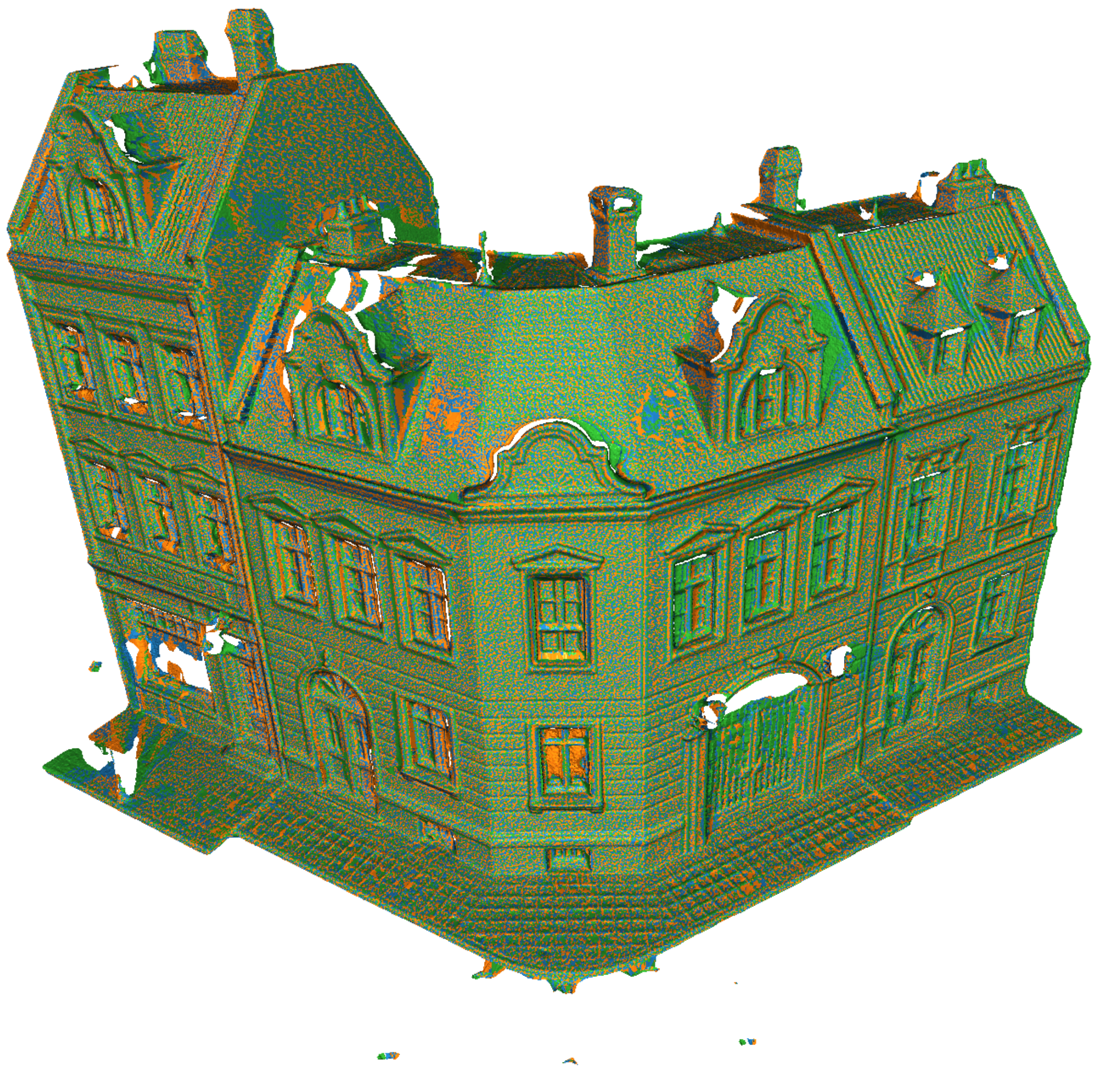




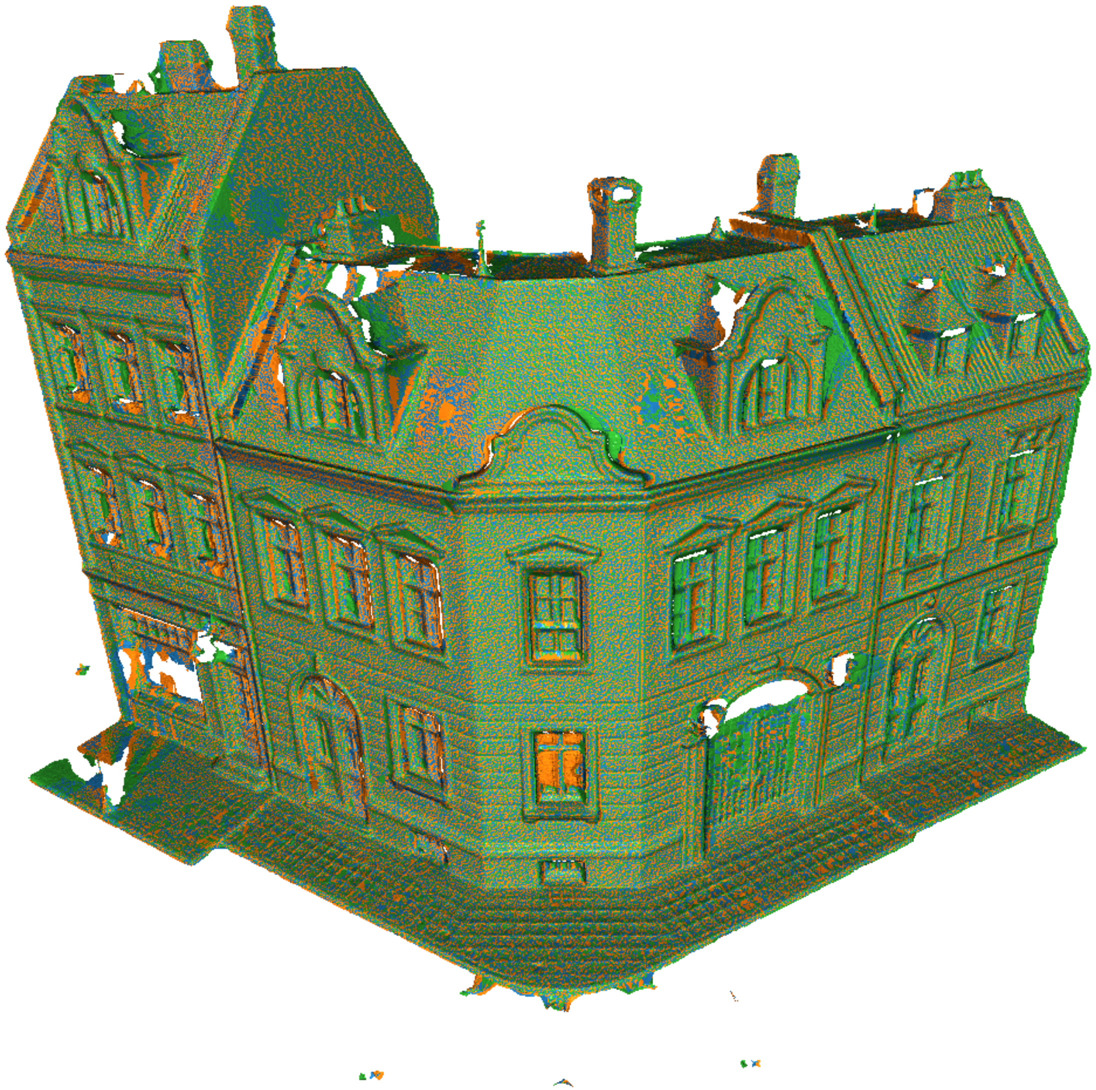
- Original scans, color coded



- NRA applied with $\mathbf{T}$ = graph Laplacians computed from Co3Ne reconstruction



- NRA applied with $\mathbf{T}$ = centroids of k-nearest neighbours.

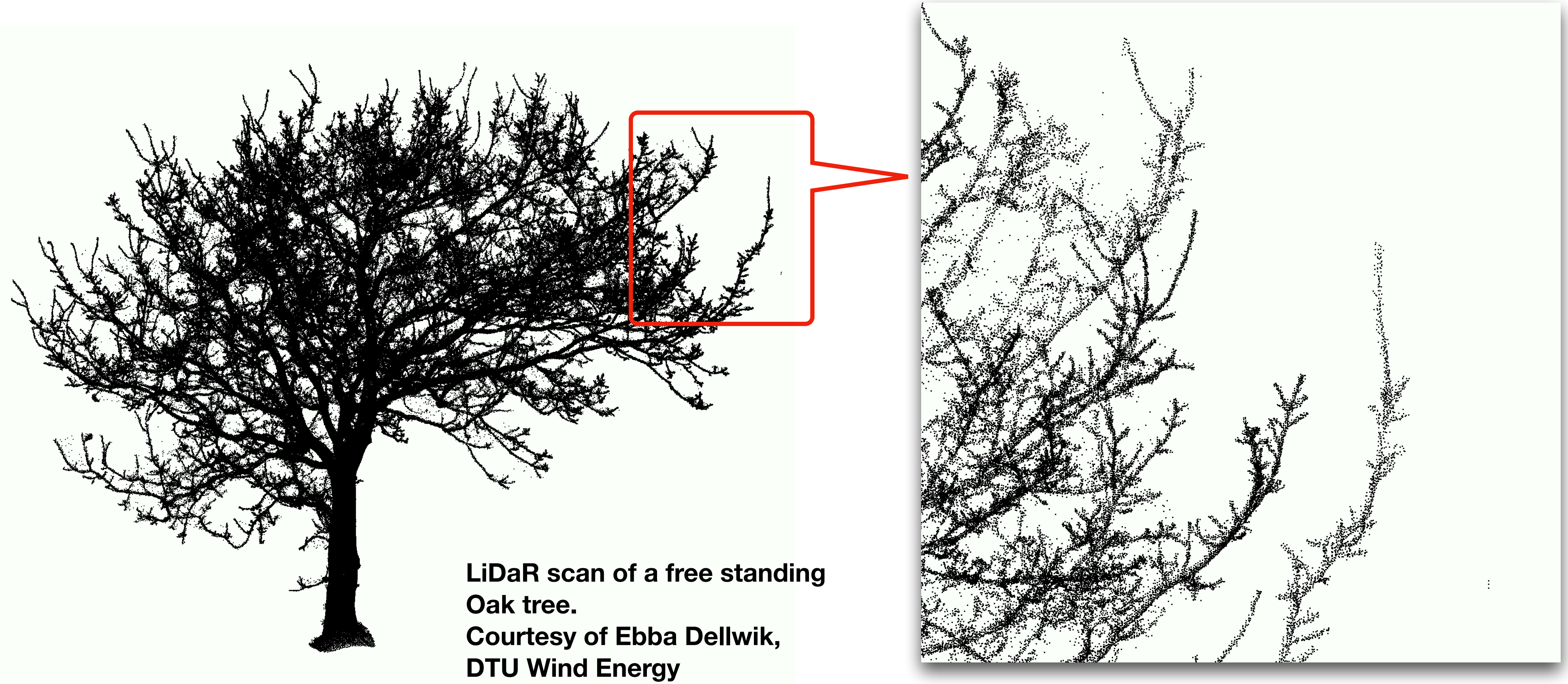


- Owl before (left) and after (right) NRA with graph Laplacian displacements

Comments and Caveats

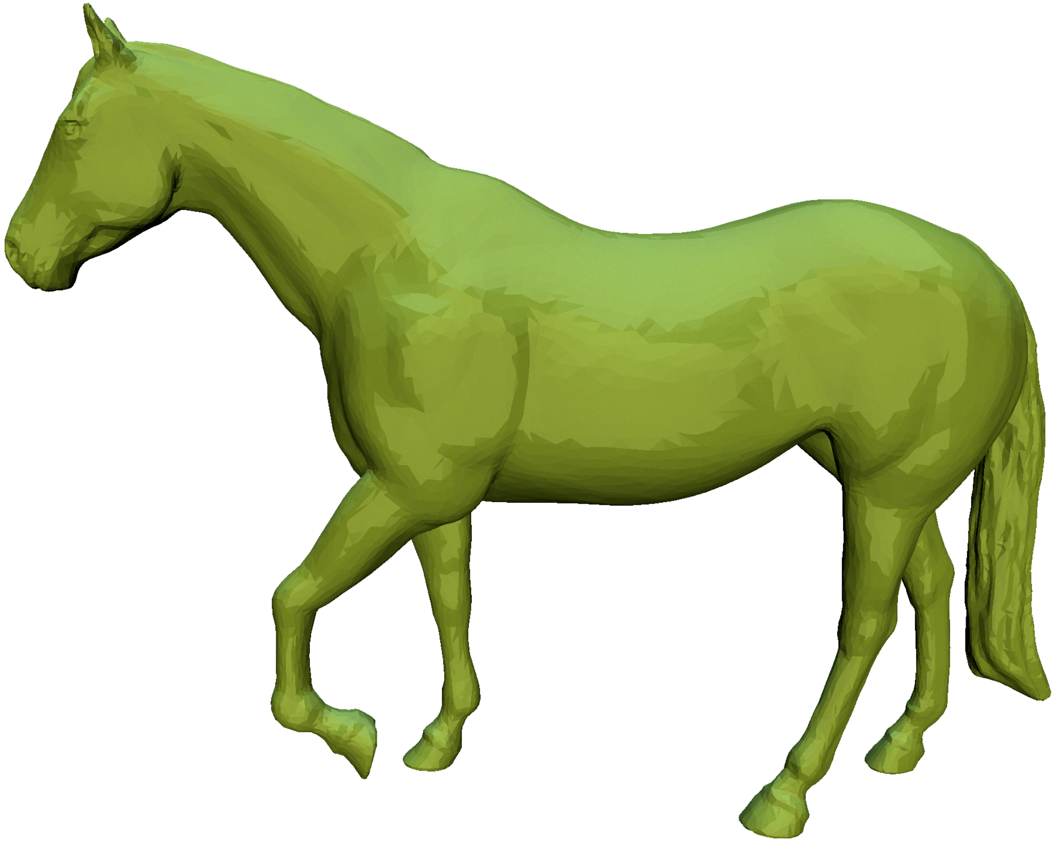
- This NRA method assumes a good rigid registration and only limited deformation
- You can apply NRA even after combinatorial reconstruction, but also using the neighborhood graph of the combined point cloud.
- The main benefit of graph Laplacians computed from mesh is speed.

Features not handled so far ...



LiDaR scan of a free standing Oak tree.
Courtesy of Ebba Dellwik,
DTU Wind Energy

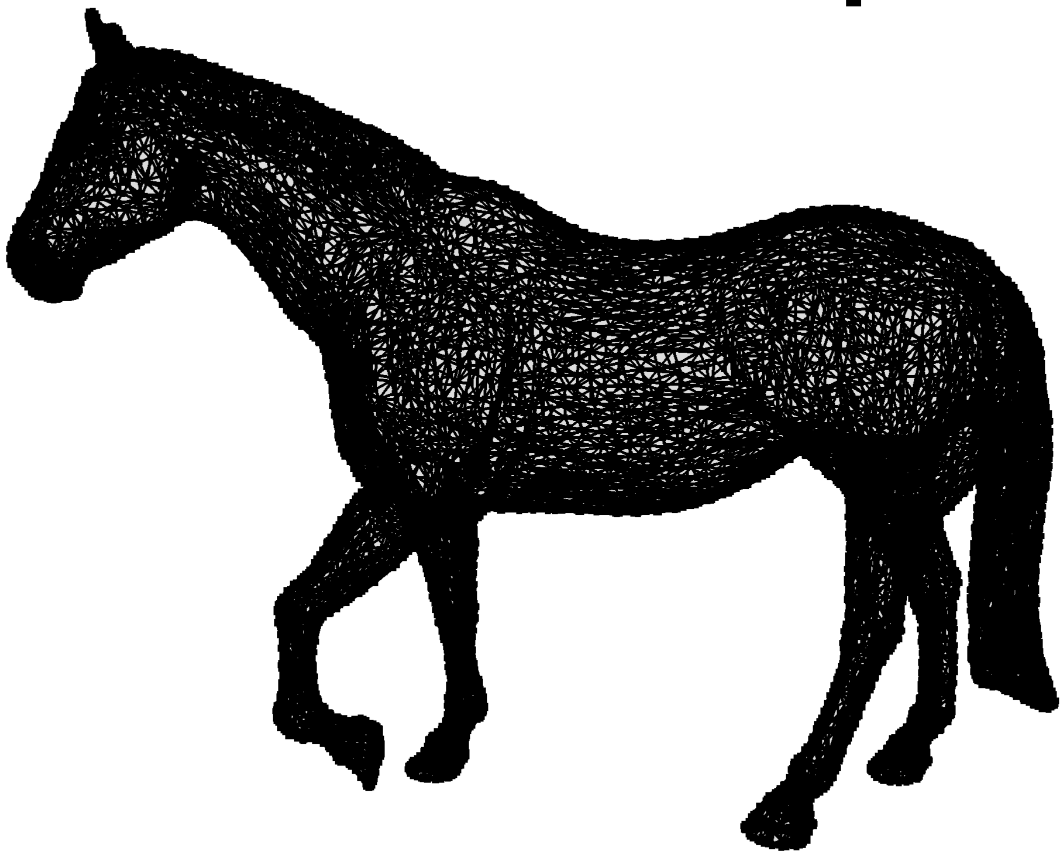
3D Model



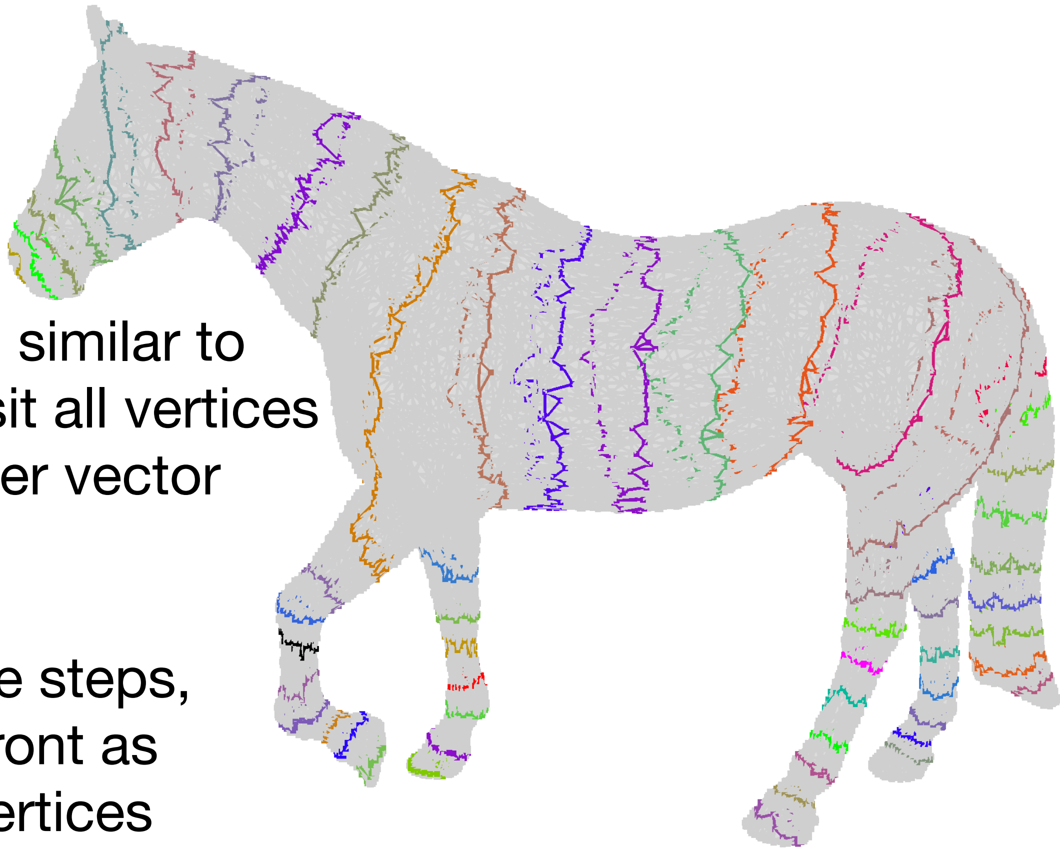
Fiedler Vector

Aka first non-constant eigenvector of the Laplace-Beltrami Opreator

Mesh as Graph



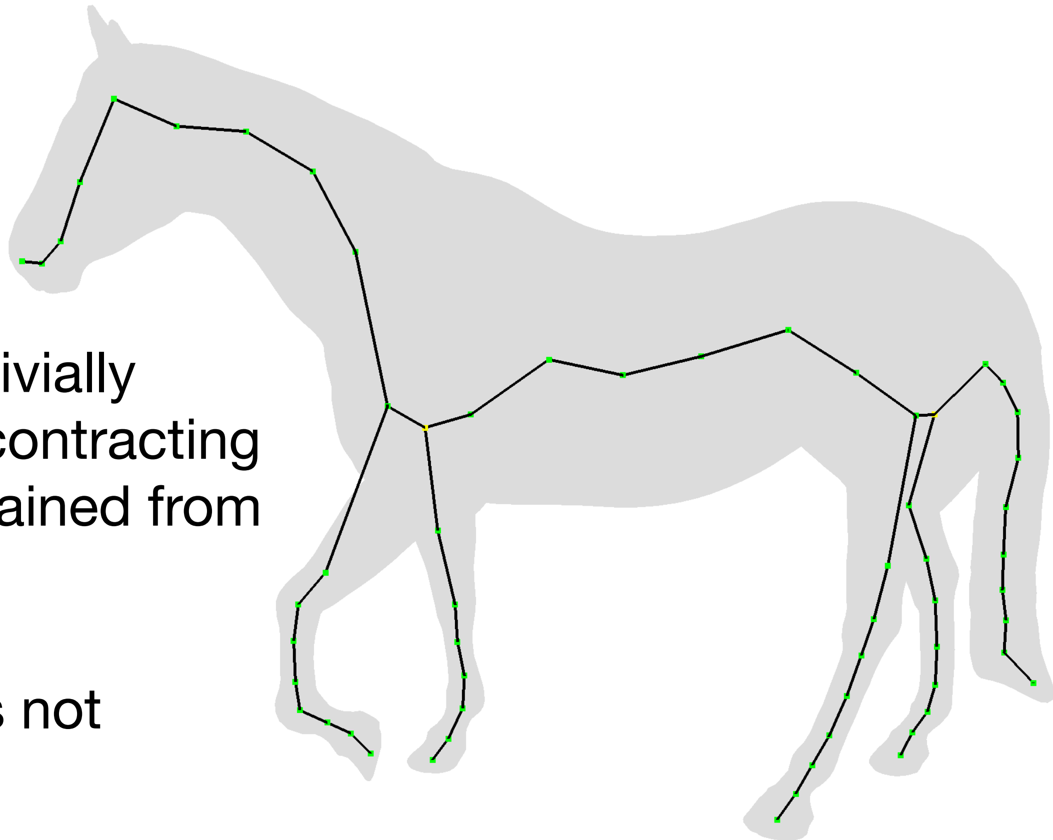
Separators from Fiedler vector



- Using algorithm similar to Dijkstra's we visit all vertices in order of Fiedler vector value

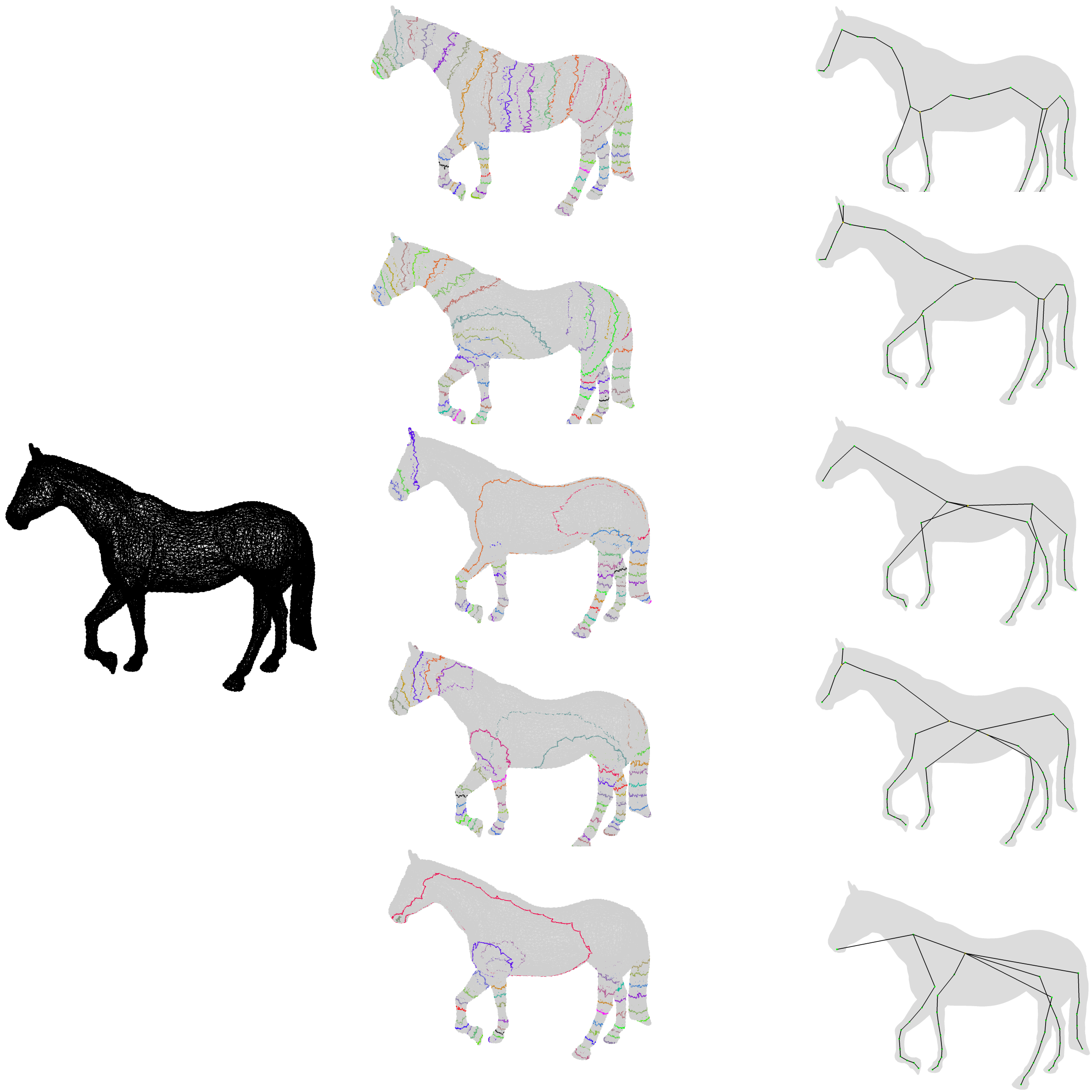
- For specific time steps, we output the front as a selection of vertices (color coded)

Skeleton



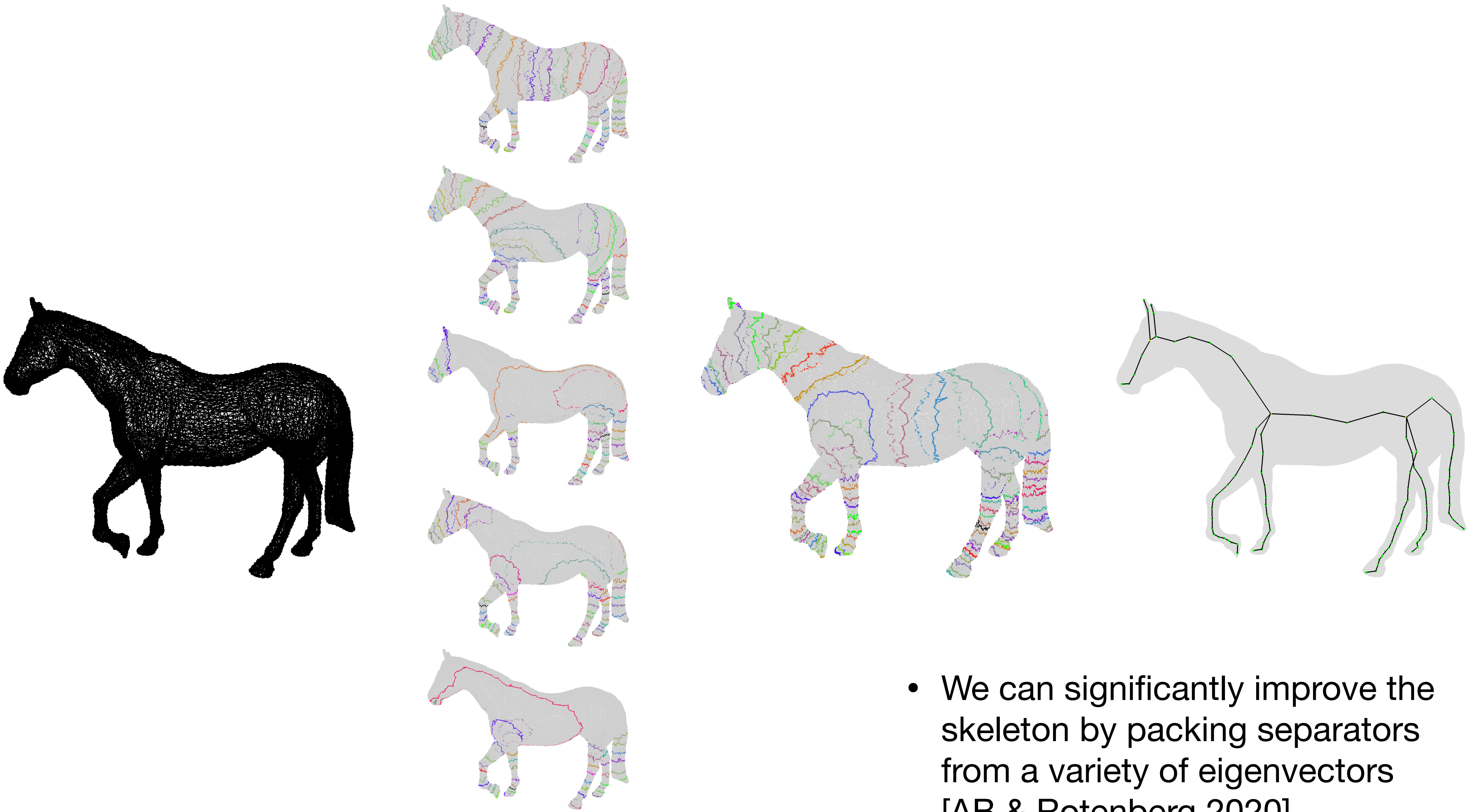
- A skeleton is trivially computed by contracting separators obtained from front sets.

- The skeleton is not satisfactory



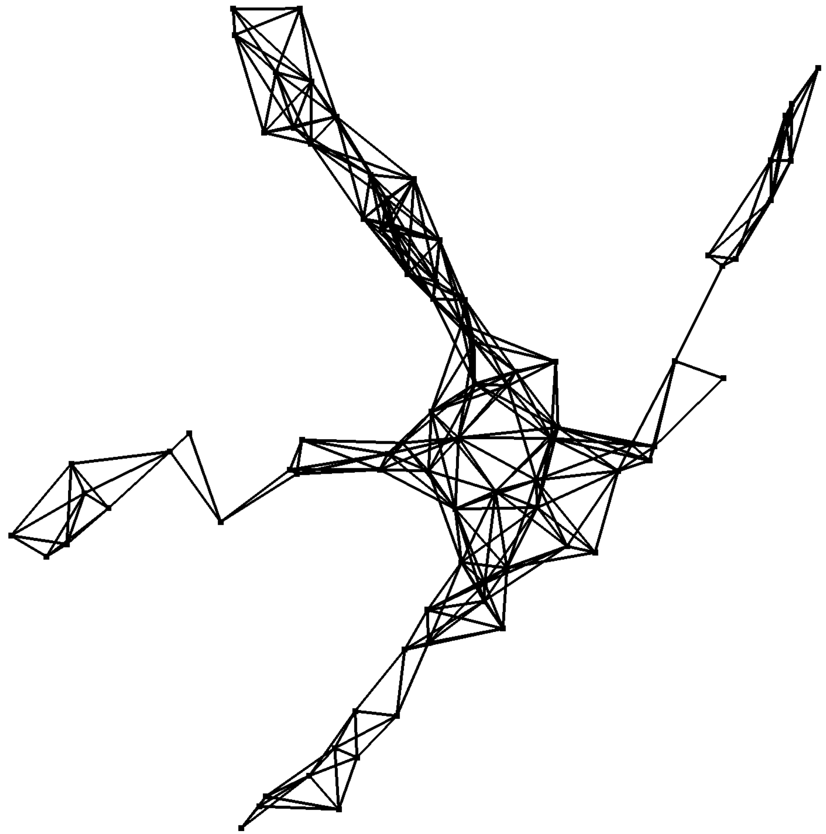
- Repeat the process for several eigenvectors of the Laplace-Beltrami eigenvector

- Results are increasingly hopeless …

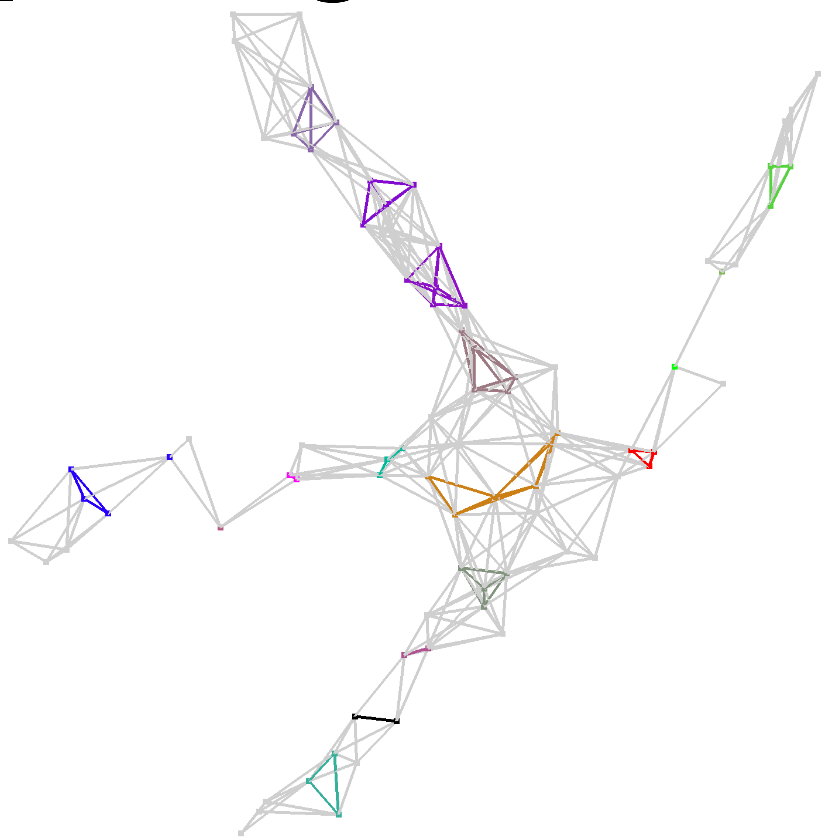


- We can significantly improve the skeleton by packing separators from a variety of eigenvectors [AB & Rotenberg 2020]

Computing Local Separators



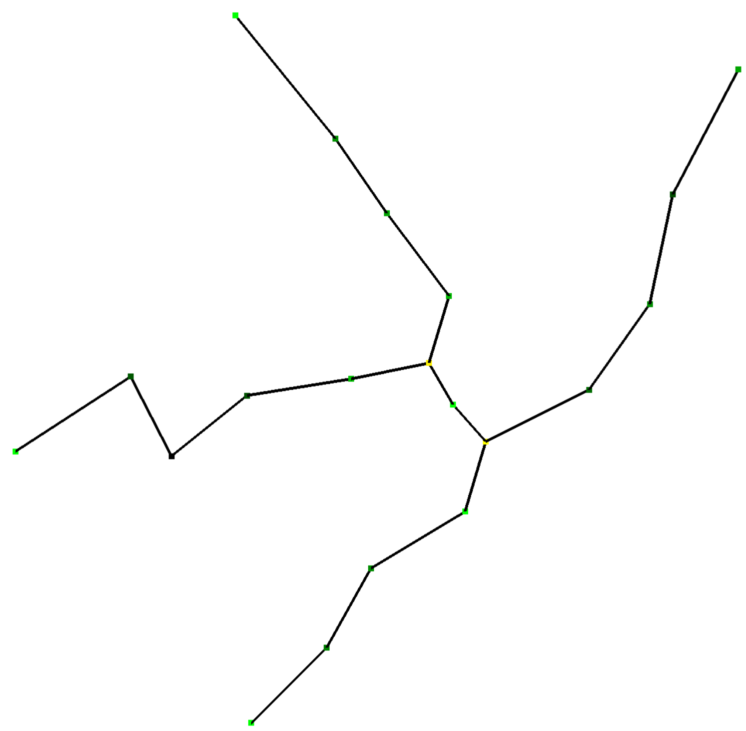
Computing Local Separators



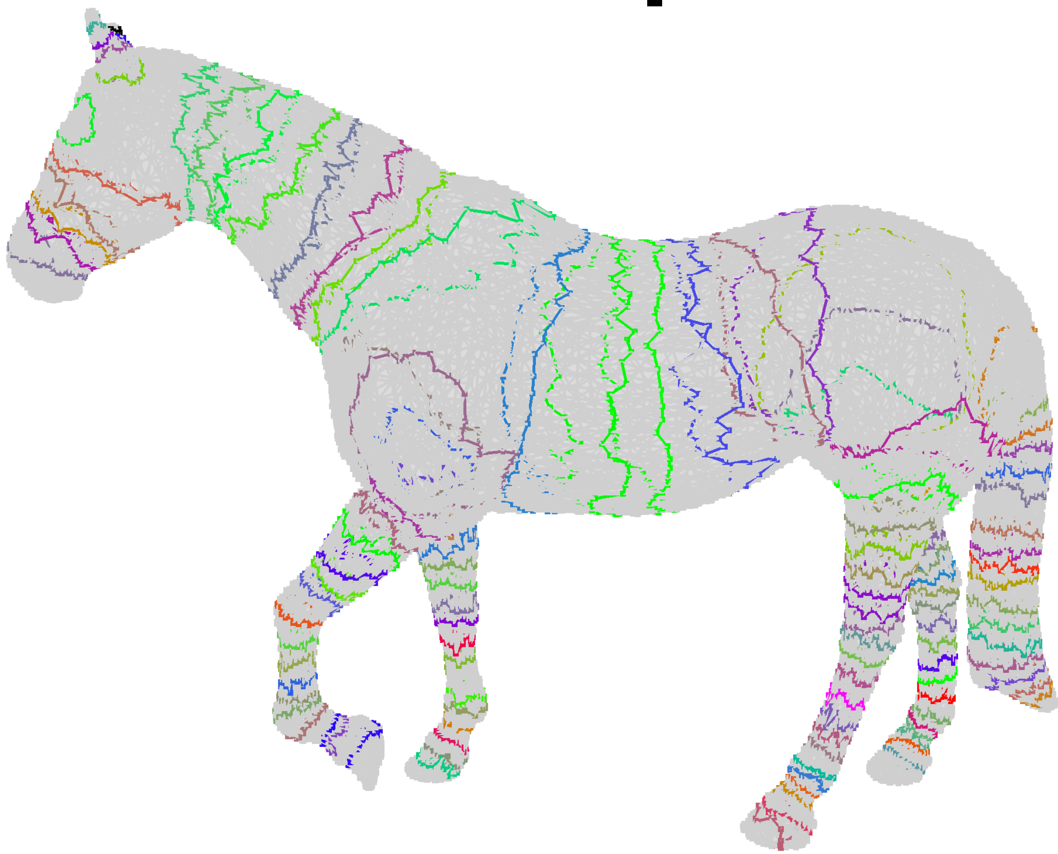
Local separators are separators of a subgraph. In practice, we grow a cluster of vertices and a separator is found when the front breaks into two components

[AB & Rotenberg 2020]

Skeleton from Local Separators

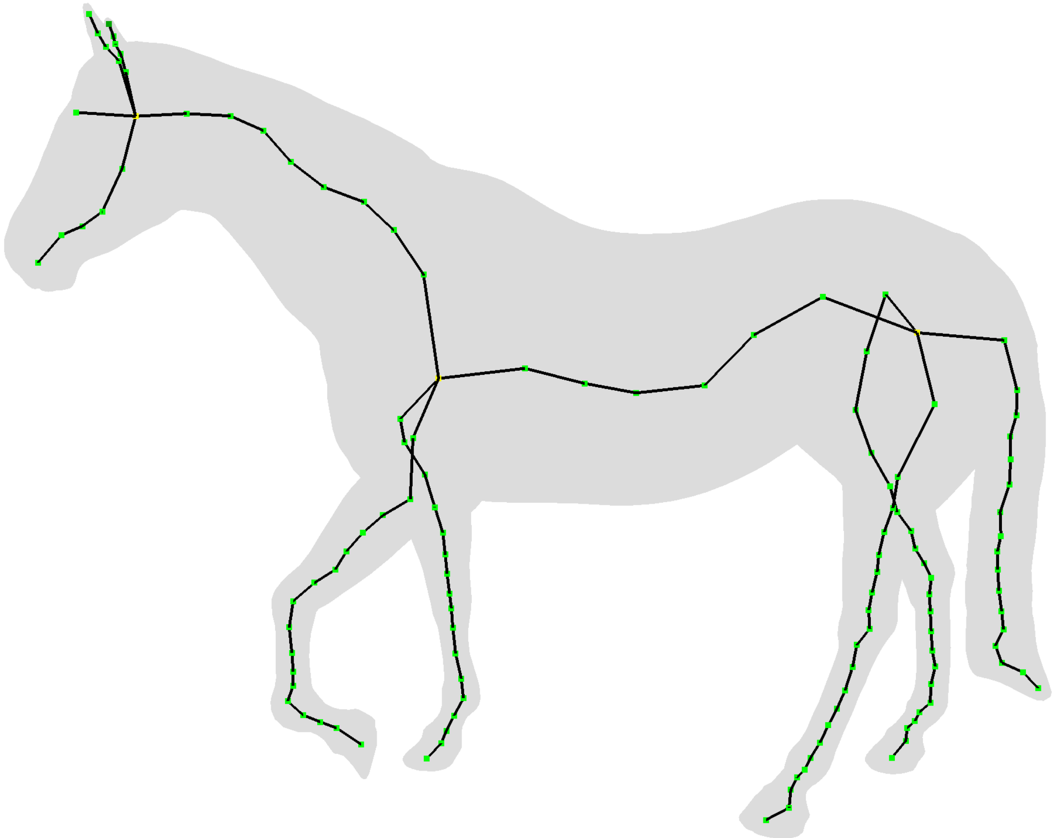


Local Separators



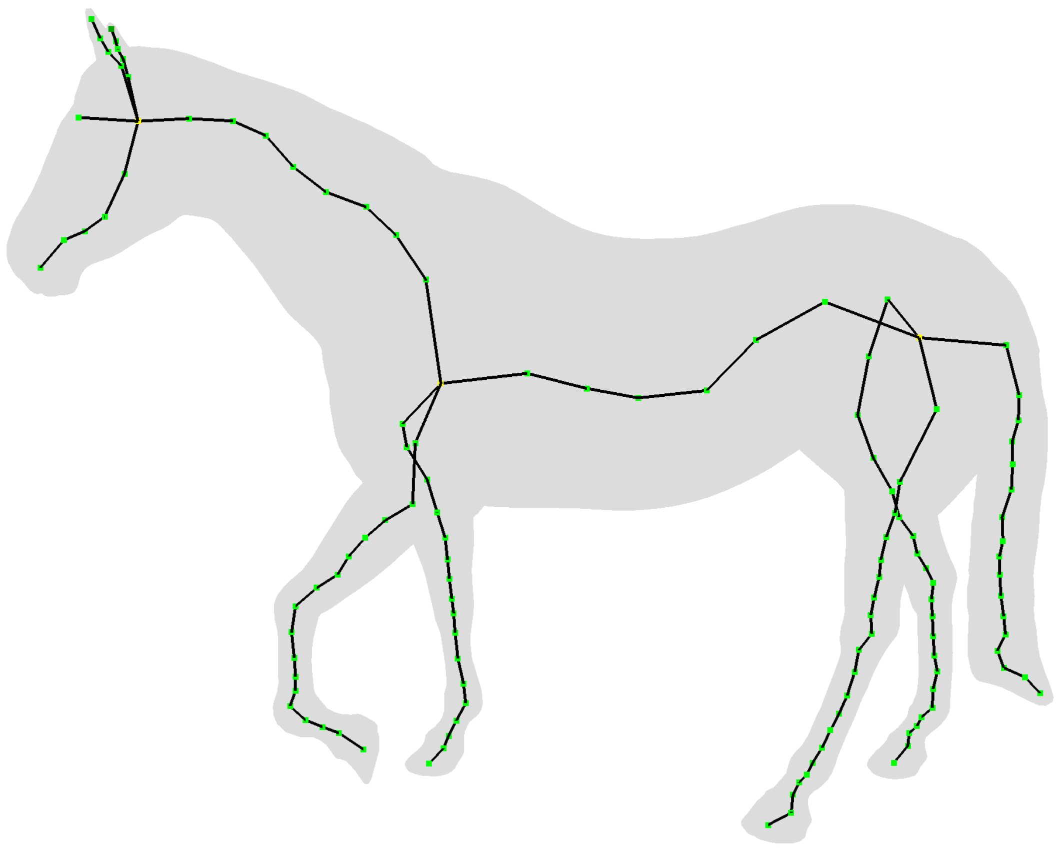
[AB & Rotenberg 2020]

Skeleton from Local Separators

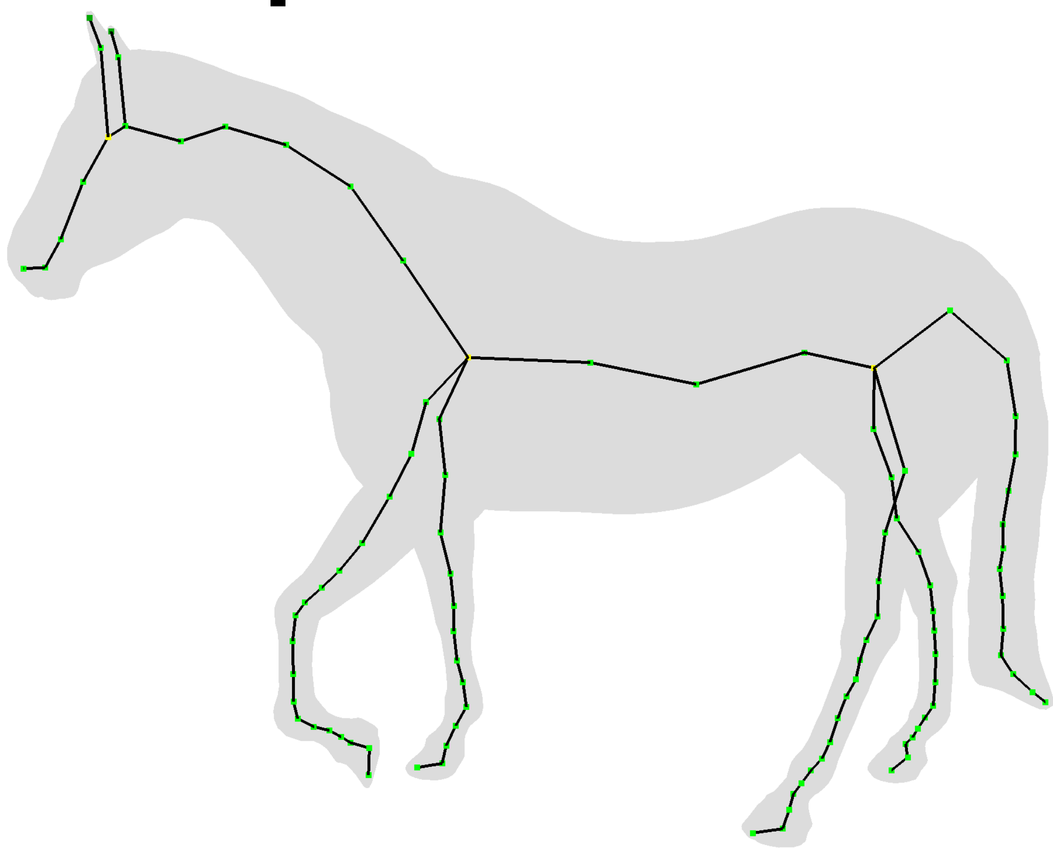


[AB & Rotenberg 2020]

Skeletons Compared



Local separators



LBO Eigen vector skeleton

Skeleton of Tree

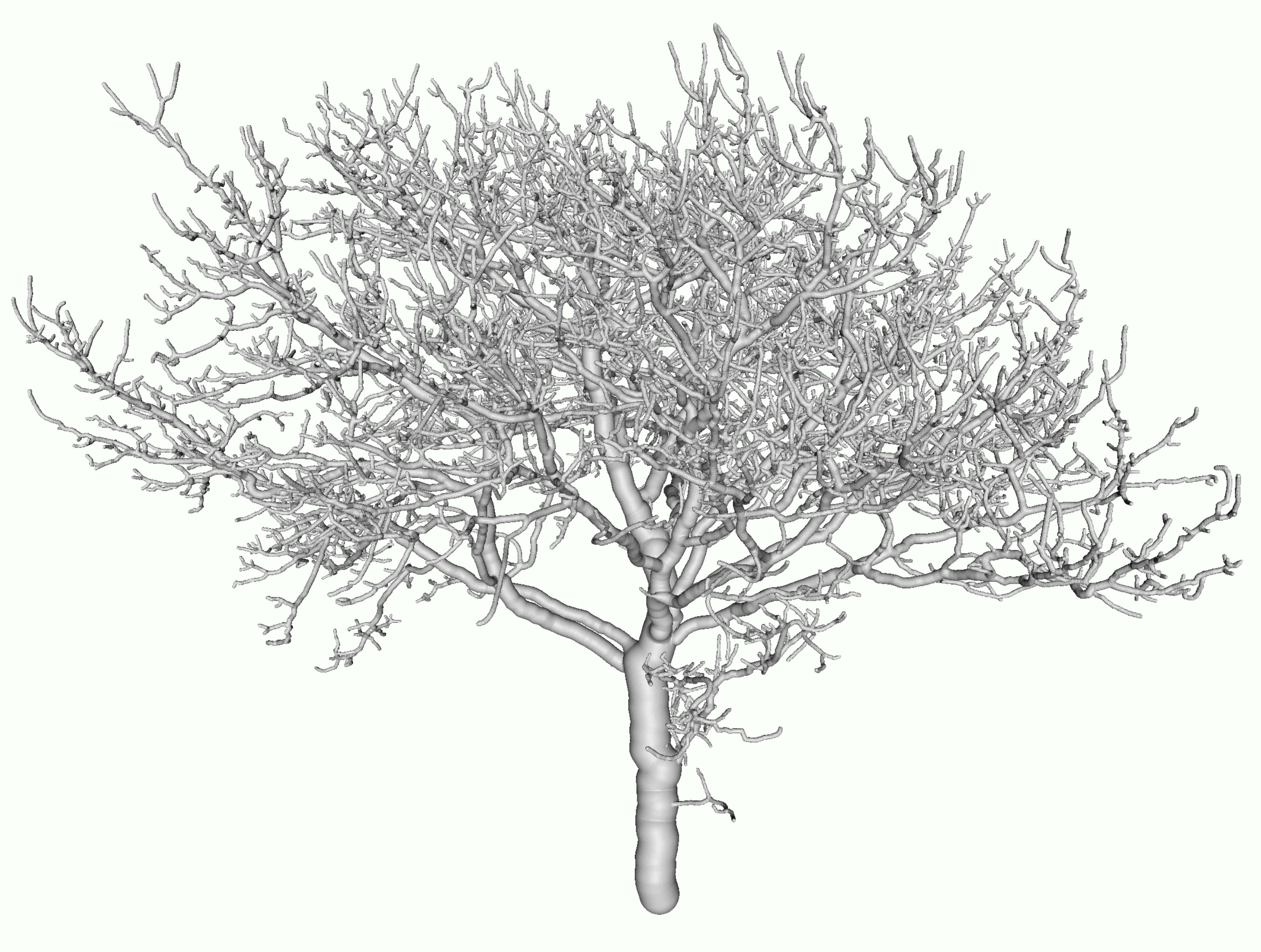
Reconstructing the Tree

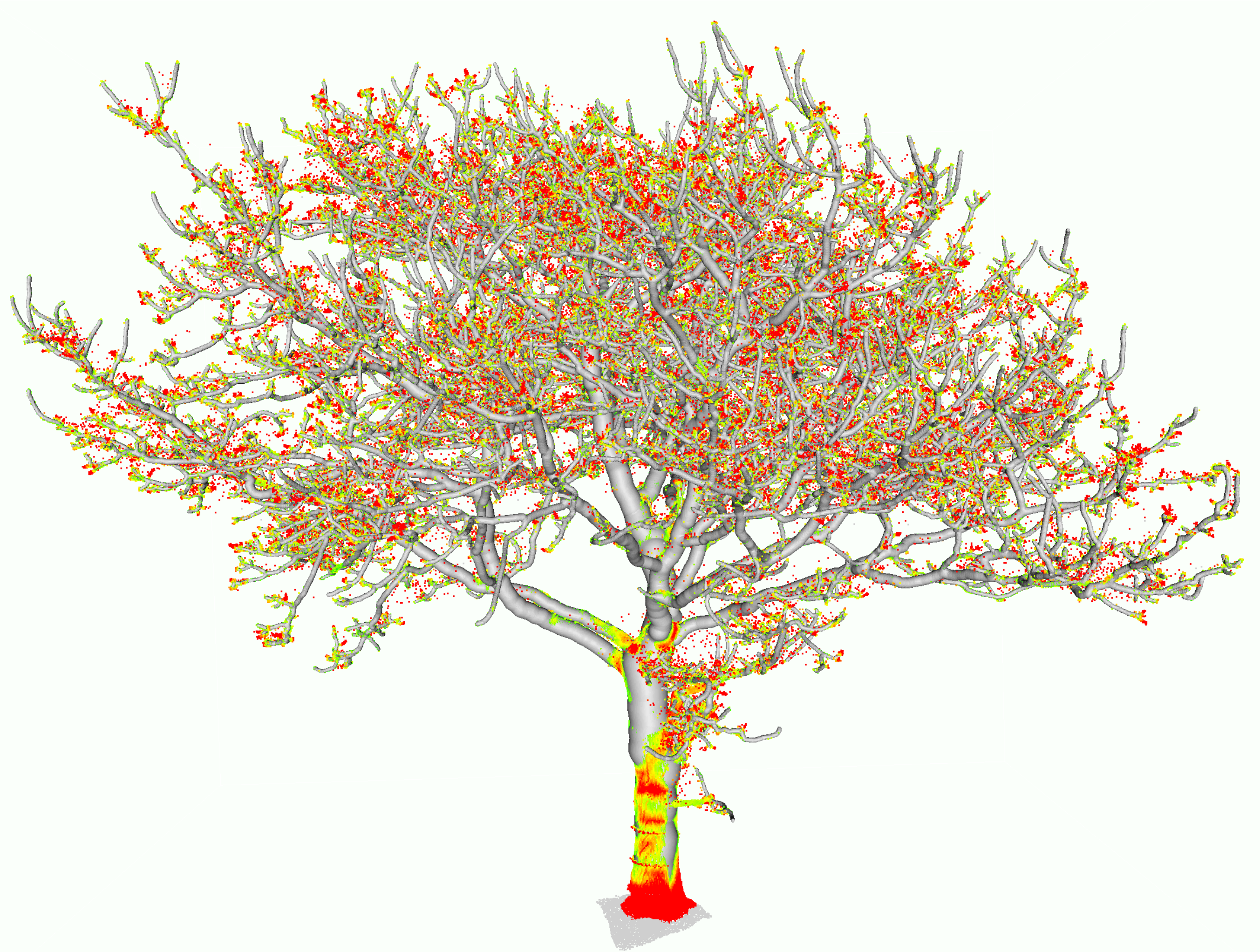
- Leonardo da Vinci estimated that in branches, the cross-sectional area of daughter branches sum to the area of the mother branch [Villesen 2020]:

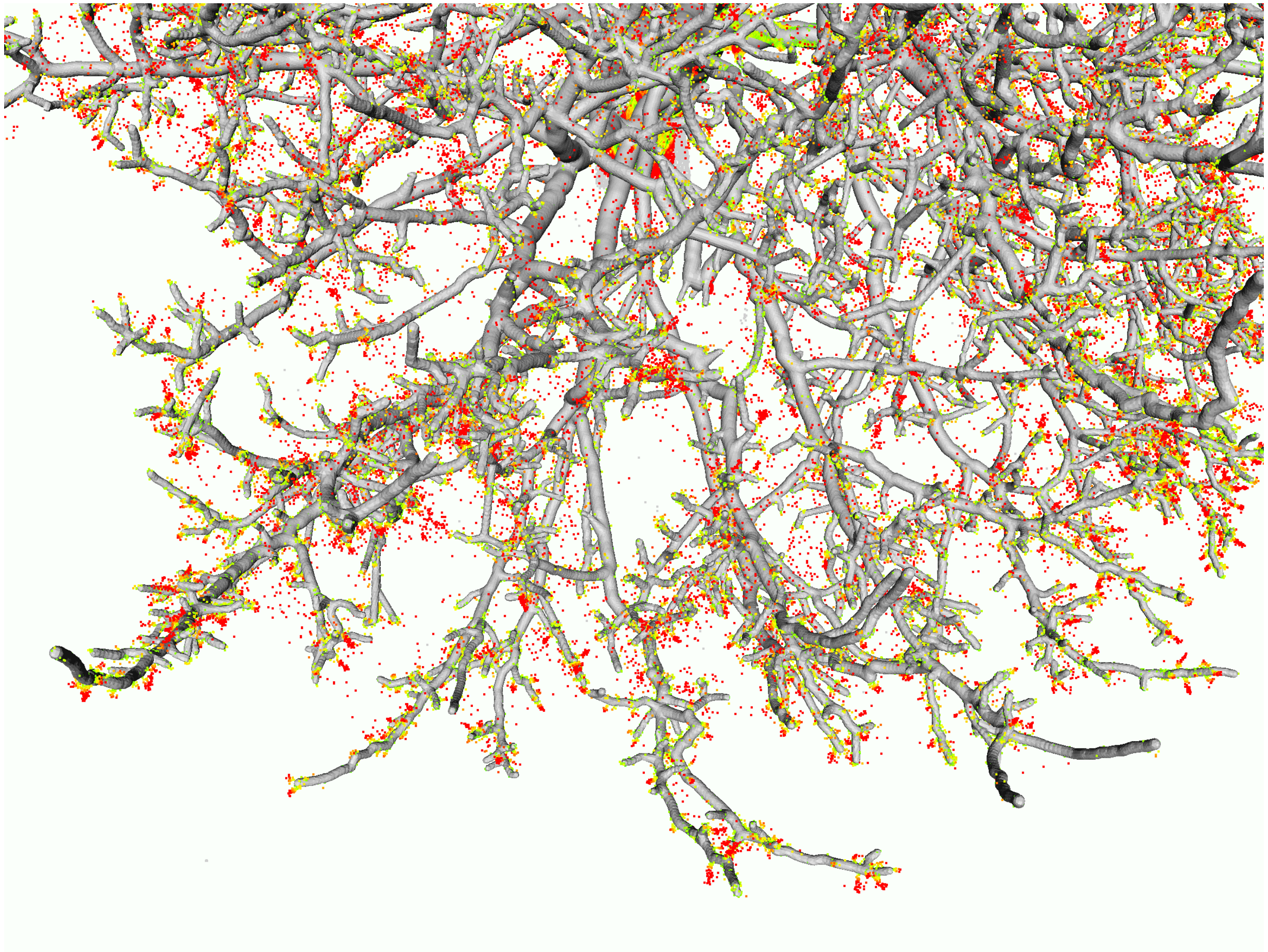
$$d^{\Delta} = \sum_{i=1}^{N} d_i^{\Delta}$$

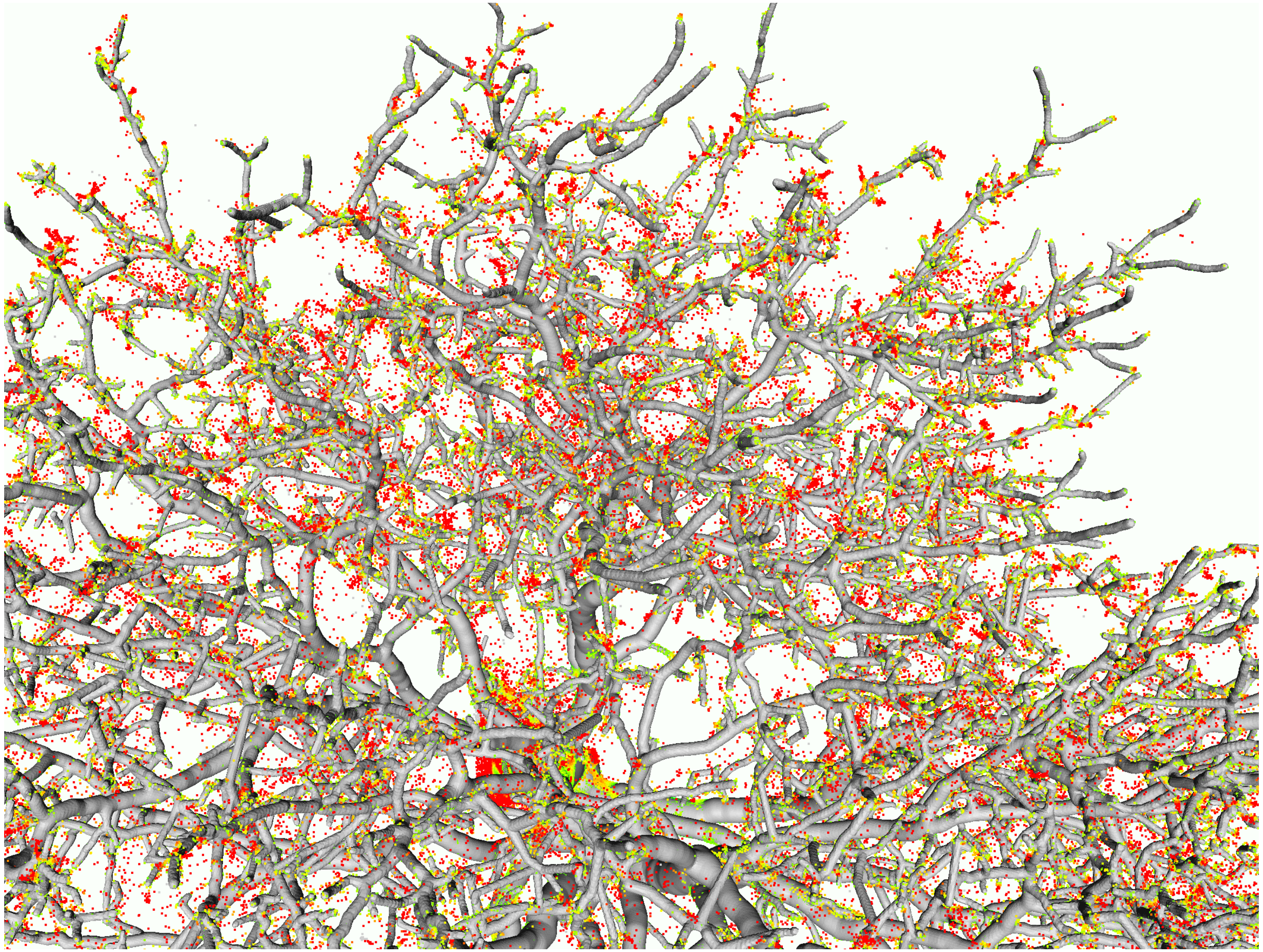
- This appears to be a good model near the trunk, less so close to the smallest twigs. Tree was reconstructed using convolution surfaces with radius given by equation above using model fitted Δ

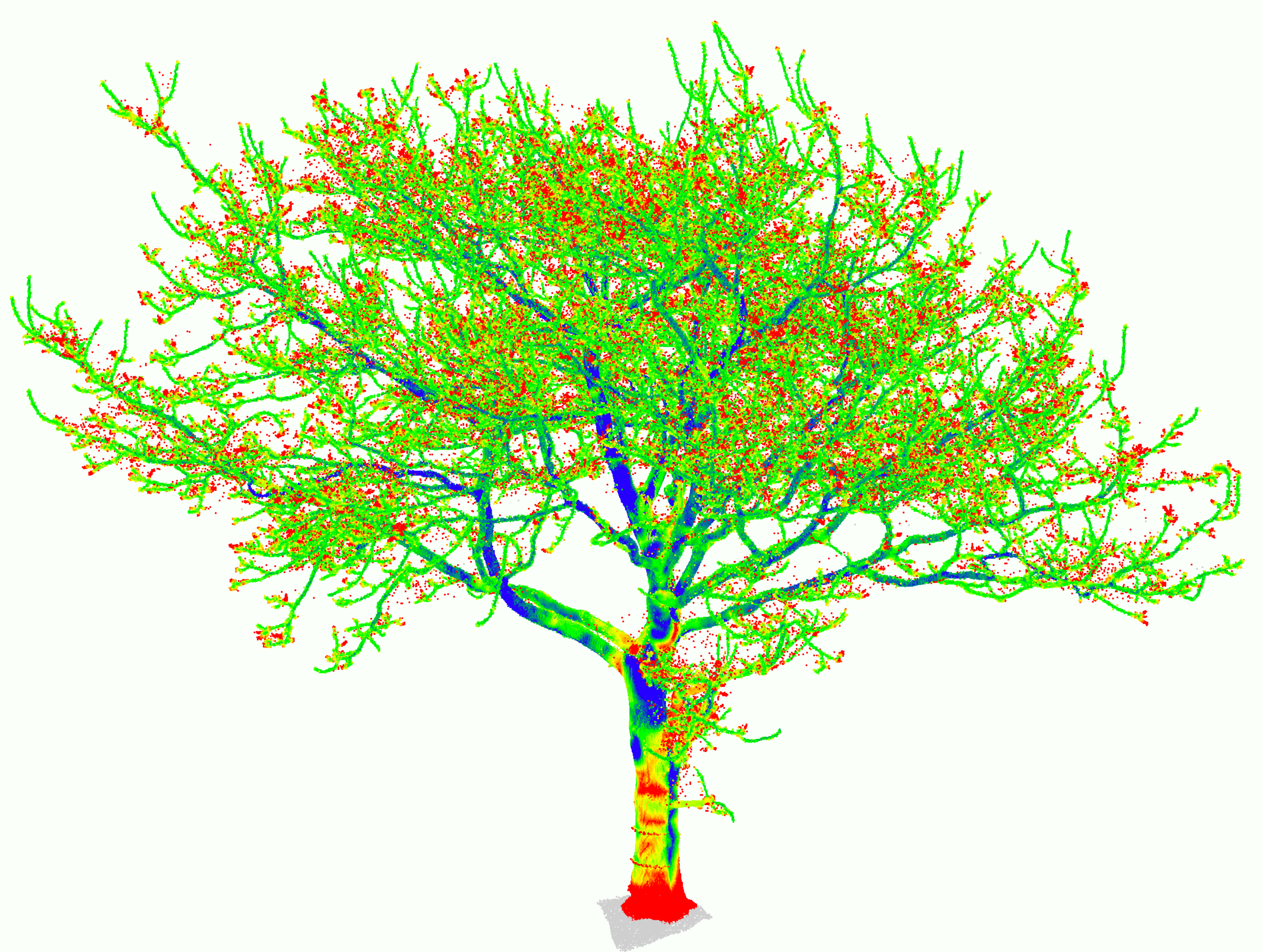
- The skeleton graph was also processed to attach loose branches [Villesen 2020]











Acknowledgements

- Ebba Dellwik
- Ida Bukh Villesen
- Eva Rotenberg
- Florian Gawrilowicz

Looking Forward

- Surface reconstruction is both trivial and extremely hard!
- Combinatorial methods and graph structures point to one future direction.
- Another direction is optimising shapes directly to match input images [Jensen 2021]

Thanks! Questions? References…

- AB, and Rotenberg, E. *Skeletonization via Local Separators*. arXiv preprint arXiv:2007.03483 (2020).
- Gawrilowicz, Florian and AB. Optimal, *Non-Rigid Alignment for Feature-Preserving Mesh Denoising*. 2019 International Conference on 3D Vision (3DV). IEEE, 2019
- AB; Gravesen, J.; Anton, F.; & Aanæs, H. *Guide to computational geometry processing: foundations, algorithms, and methods*. Springer Science & Business Media. 2012
- Cazals, F.; and Giesen, J. *Delaunay triangulation based surface reconstruction*. Effective computational geometry for curves and surfaces. Springer, Berlin, Heidelberg, 2006. pp. 231-276
- Digne, J.; Morel, J. M.; Souzani, C. M.; & Lartigue, C. *Scale space meshing of raw data point sets*. In Computer Graphics Forum (Vol. 30, No. 6, pp. 1630-1642, September 2011)
- Kazhdan, M.; Bolitho M.; and Hoppe, H. *Poisson surface reconstruction*. 4th EG Symposium on Geometry processing (SGP 2006).
- Nielson, G.M. *Dual marching cubes.* IEEE Visualization 2004. IEEE, 2004.
- Amenta, N.; Bern, M.; & Kamvysselis, M. *A new Voronoi-based surface reconstruction algorithm*. Proceedings of the 25th annual conference on Computer graphics and interactive techniques. 1998.
- Fuhrmann, S. and Goesele, M. *Floating scale surface reconstruction*. ACM Transactions on Graphics (ToG) 33.4 (2014): 1-11.
- Bernardini, F.; Mittleman, J.; Rushmeier, H.; Silva, C.; & Taubin, G. *The ball-pivoting algorithm for surface reconstruction*. IEEE transactions on visualization and computer graphics, 5(4), 349-359, 1999.
- Jensen, J. N.; Hannemose, M.; AB; Wilm, J.; Frisvad, J.R.; Dahl, A. *Surface Reconstruction from Structured Light Images Using Differentiable Rendering,* Sensors, 21(4), MDPI 2021.
- Frisken (Gibson), S.F. *Using distance maps for accurate surface representation in sampled volumes*. IEEE Symposium on Volume Visualization (Cat. No. 989EX300). IEEE, 1998.
- Boltcheva, D.; and Lévy, B. *Surface reconstruction by computing restricted Voronoi cells in parallel*. Computer-Aided Design 90 (2017): 123-134.
- Ida Bukh Villesen, *Reconstruction of Botanical Trees from Optically Acquired Point Clouds*, B.Sc. Thesis, Technical University of Denmark.