Accurate, Detailed, and Automatic Tree Modelling from Point Clouds

Shenglan Du

Mentor #1: Liangliang Nan

Mentor #2: Roderik Lindenbergh

TU Delft

CONTENT

- Introduction
- Related Work
- Methodology
- Results & Discussion
- Conclusion & Future Work

TU Delft

1. Introduction

1.1 Motivation

- Why do we want to model trees?

1.1 Motivation

1.1 Motivation

- Why do we want accurate tree models?

1.1 Motivation

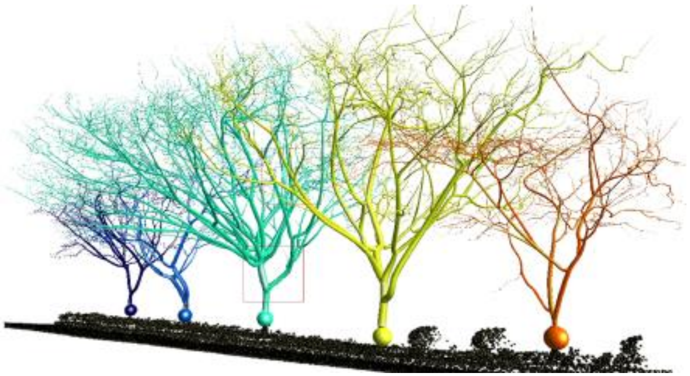
1.2 Research Objective

- 3D tree modeling from point clouds
  - Accurate (geometrically correct)
  - Detailed (topologically faithful)
  - Automatic

1.3 Research Scope

- Focus on branch reconstruction

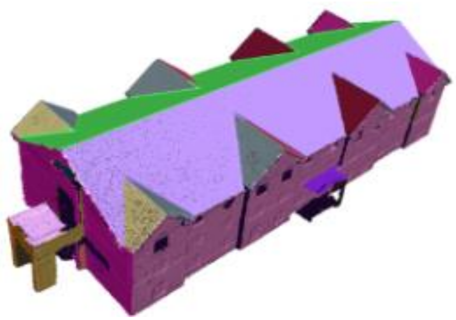
- Focus on individual tree reconstruction



TU Delft

1.4 Challenges

- Trees are complex


- Data is incomplete

Static scanner

Mobile scanner

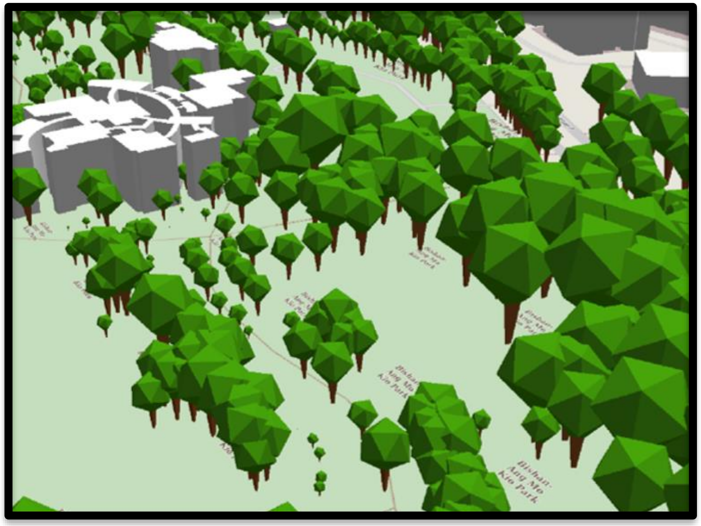
AHN3 data

TU Delft

2. Related Work

2.1 Modelling “icon” trees

- **Modelling 3D trees for CityGML Singapore** (Soon et al. 2017)



LOD1

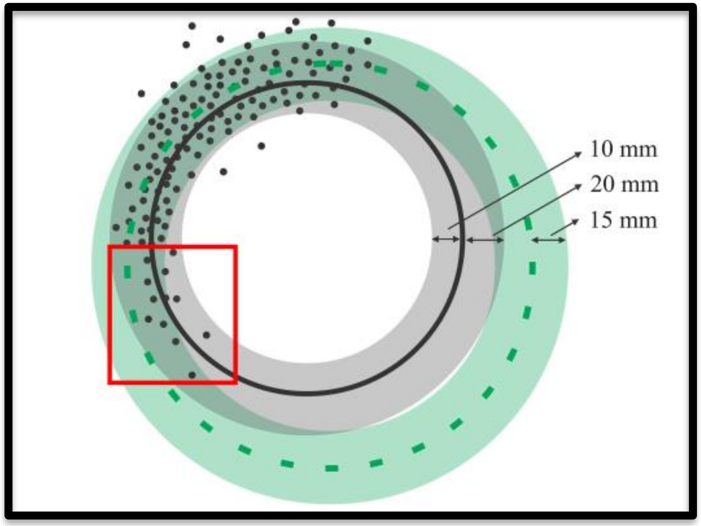
LOD2

  – Can model trees on a large region
  – Tree model is not accurate or detailed

TU Delft

2.2 Cylinder-fitting approach

- **Automatic trunk reconstruction** (Wang et al. 2016)



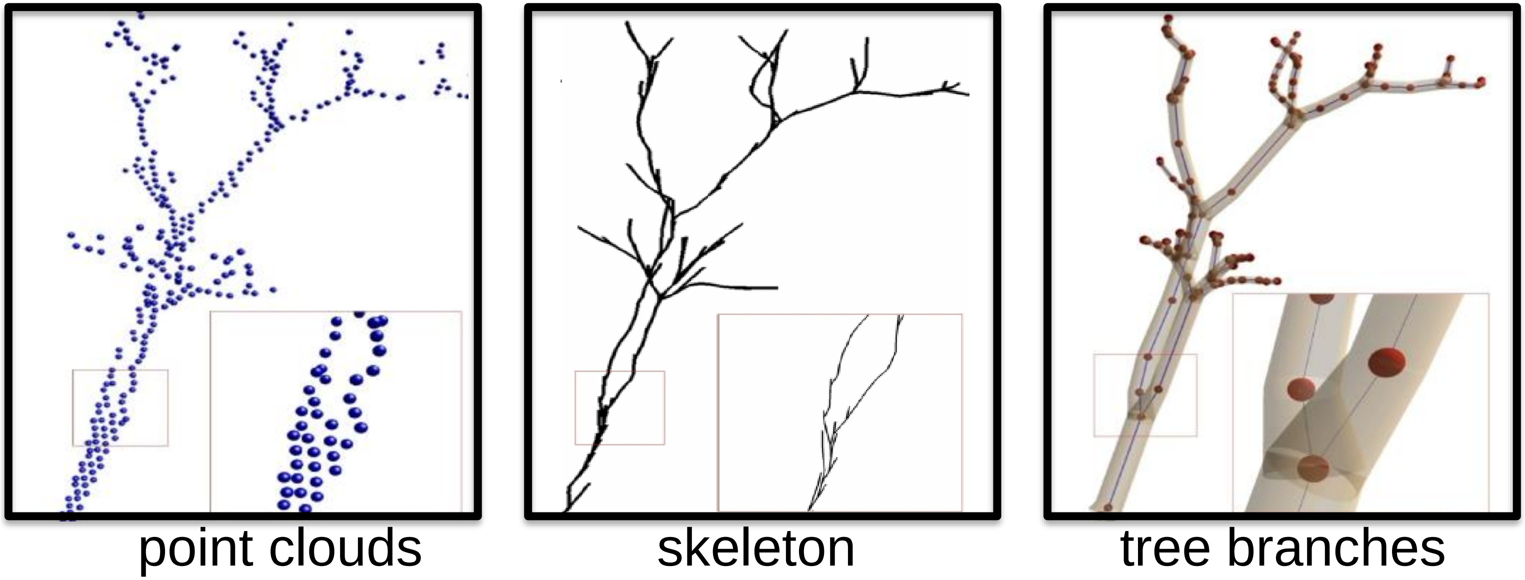
fitting process

reconstructed trunk

  - Can model the trunk accurately
  - Doesn’t consider other small tree branches

2.3 Skeleton-based approach

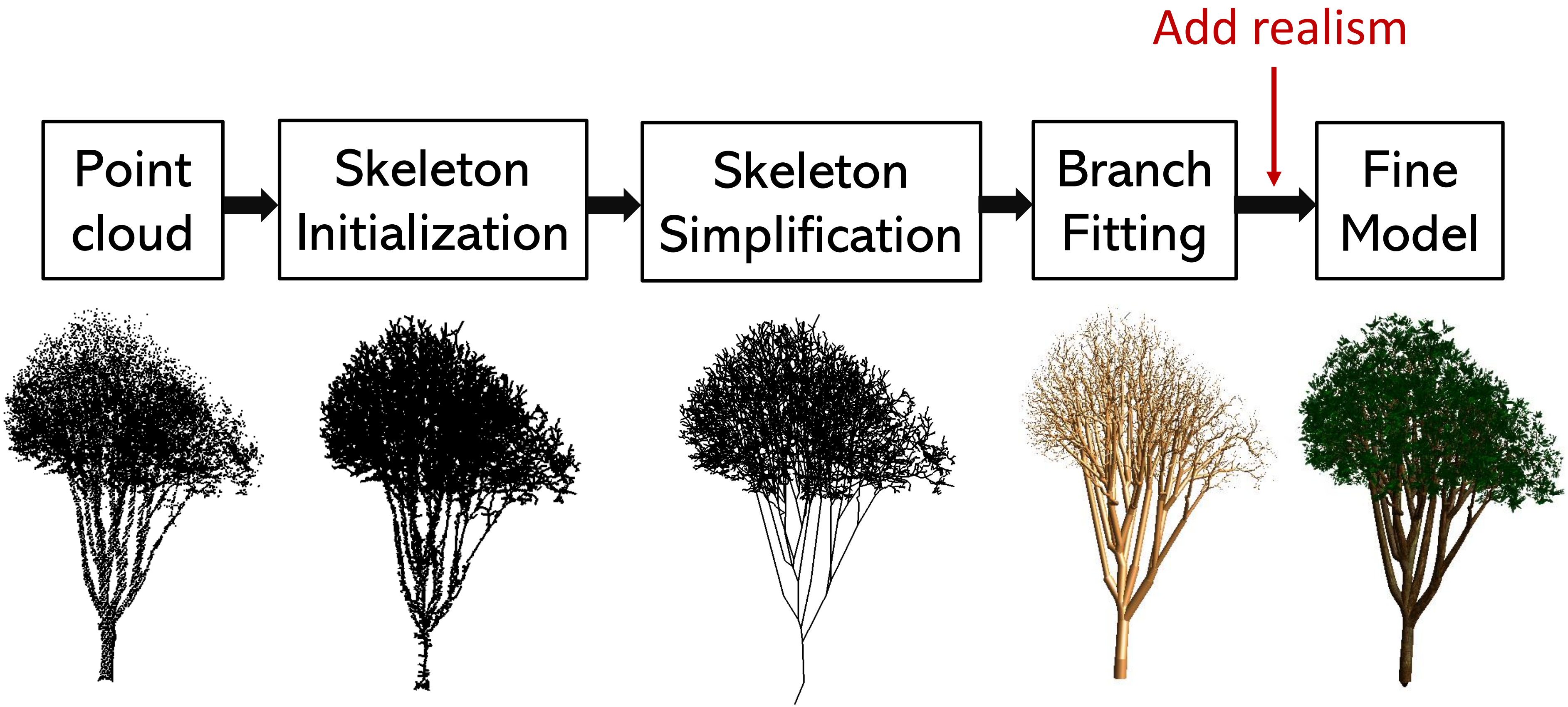
- **Automatic reconstruction of tree skeleton** (Livny et al. 2010)



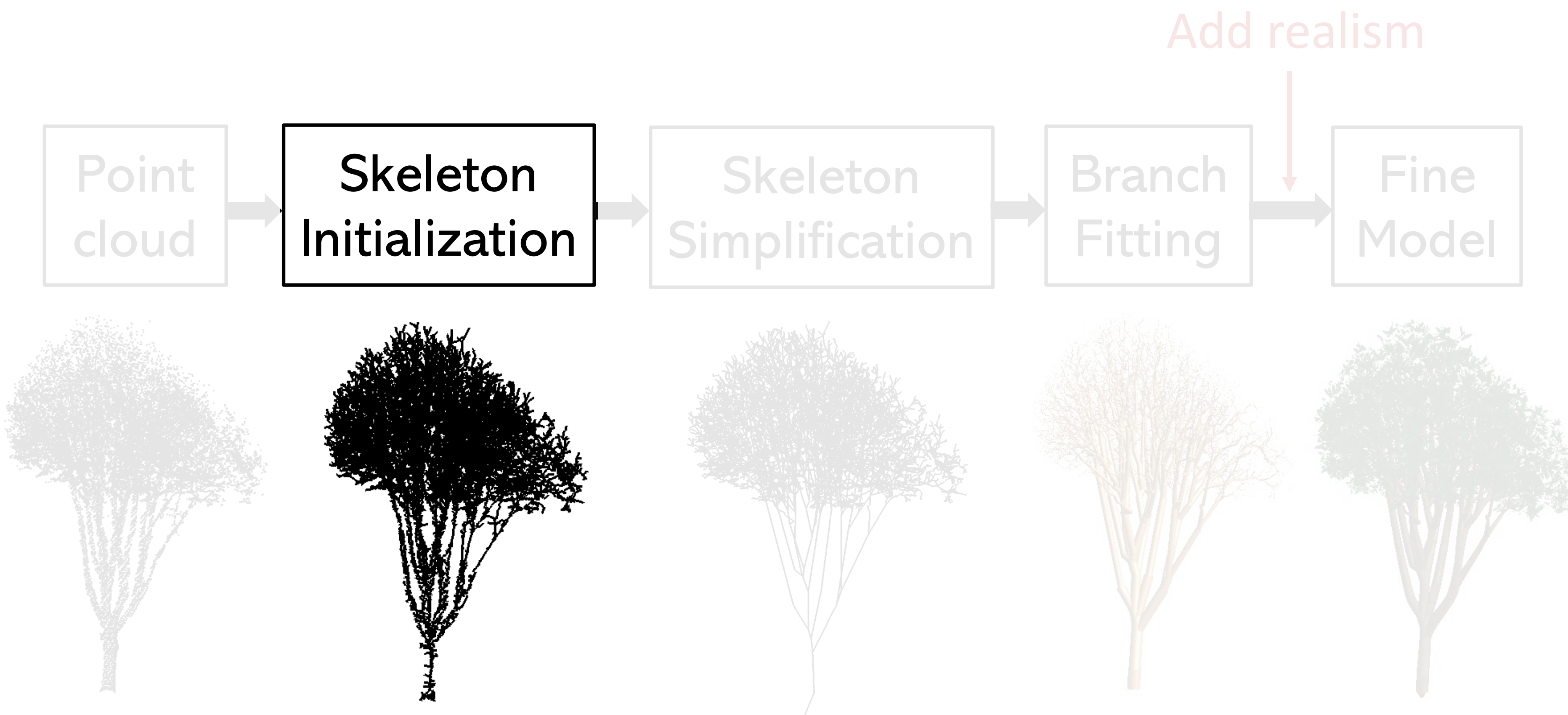
  - Can model the complicated skeletal structure of the tree
  - Doesn’t fit enough to the input points

3. Methodology

3. Methodology Overview

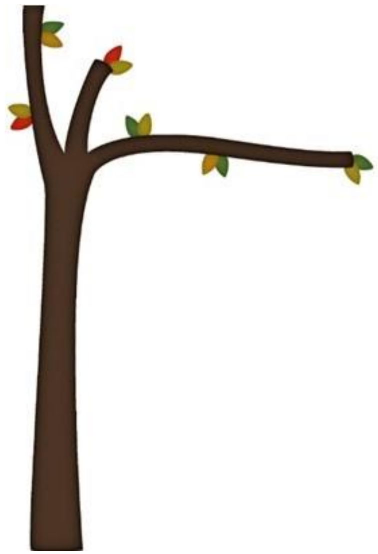


3. Methodology Overview

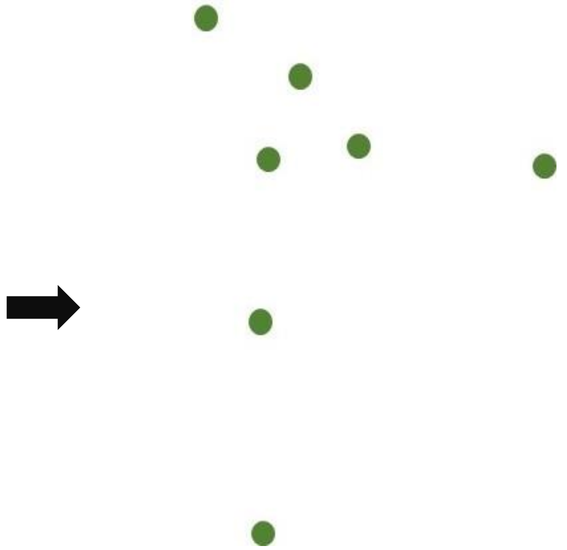


3.1 Skeleton Initialization

- We obtain initial tree skeleton from the minimum spanning tree
  - Read input points



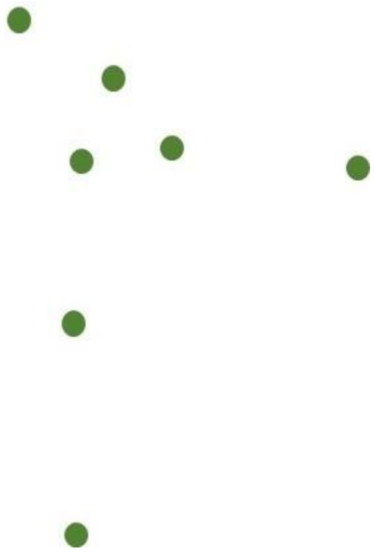
Real tree



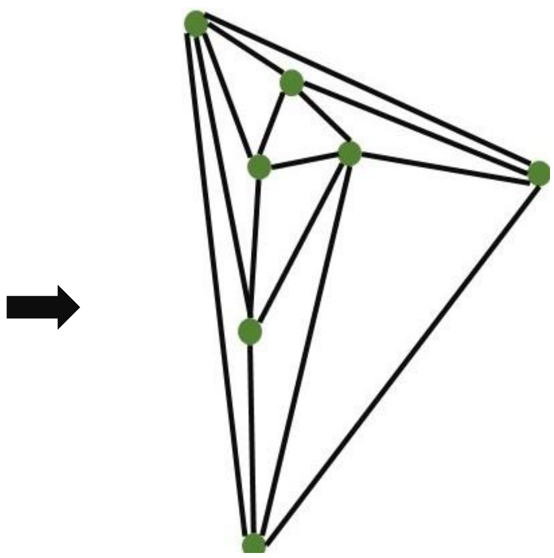
Point cloud

3.1 Skeleton Initialization

- We obtain initial tree skeleton from the minimum spanning tree
  - Construct Delaunay triangulation



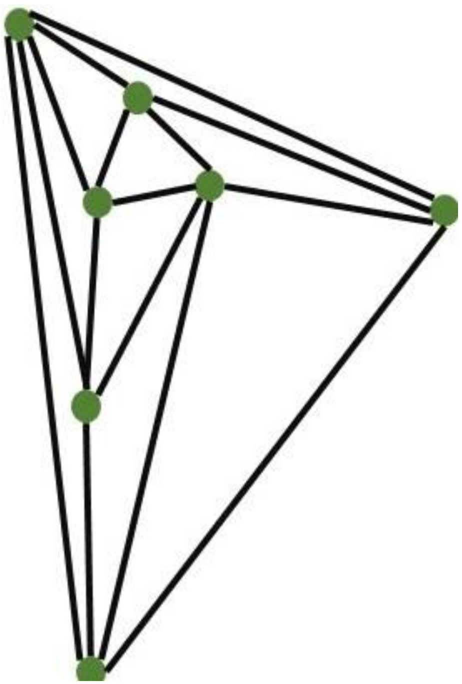
Point cloud



Delaunay triangulation

3.1 Skeleton Initialization

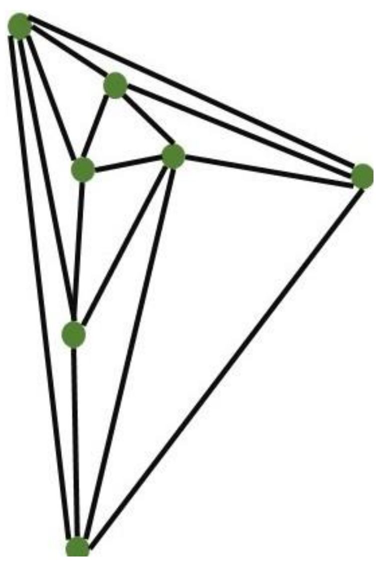
- We obtain initial tree skeleton from the minimum spanning tree



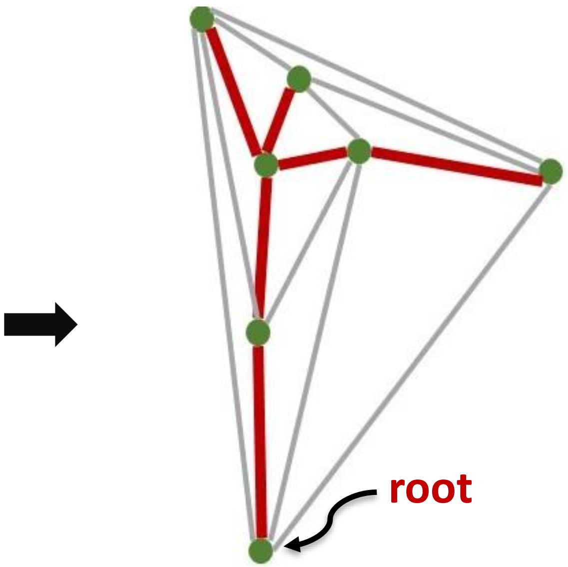
  - It provides a very initial graph
  - It completes missing parts of data

3.1 Skeleton Initialization

- We obtain initial tree skeleton from the minimum spanning tree
  - Compute minimum spanning tree from shortest path



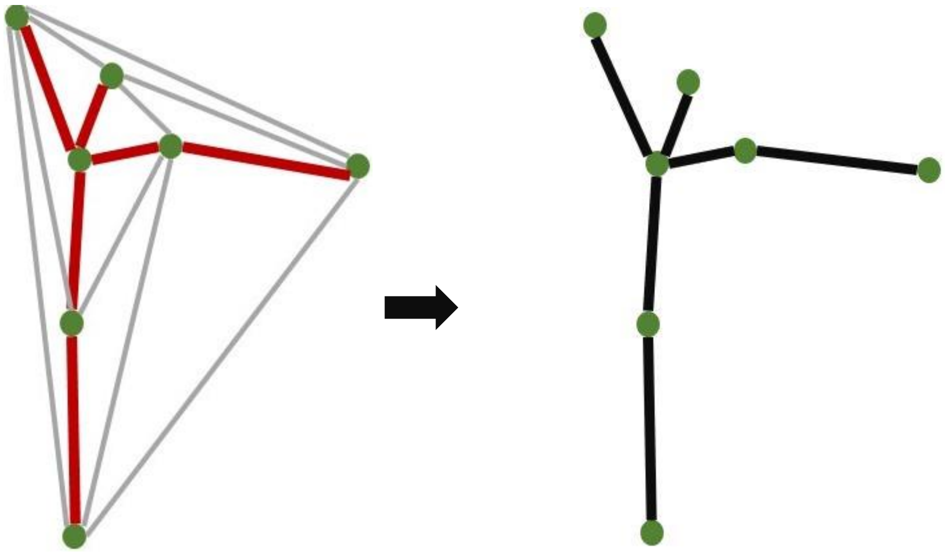
Delaunay triangulation



Minimum spanning tree

3.1 Skeleton Initialization

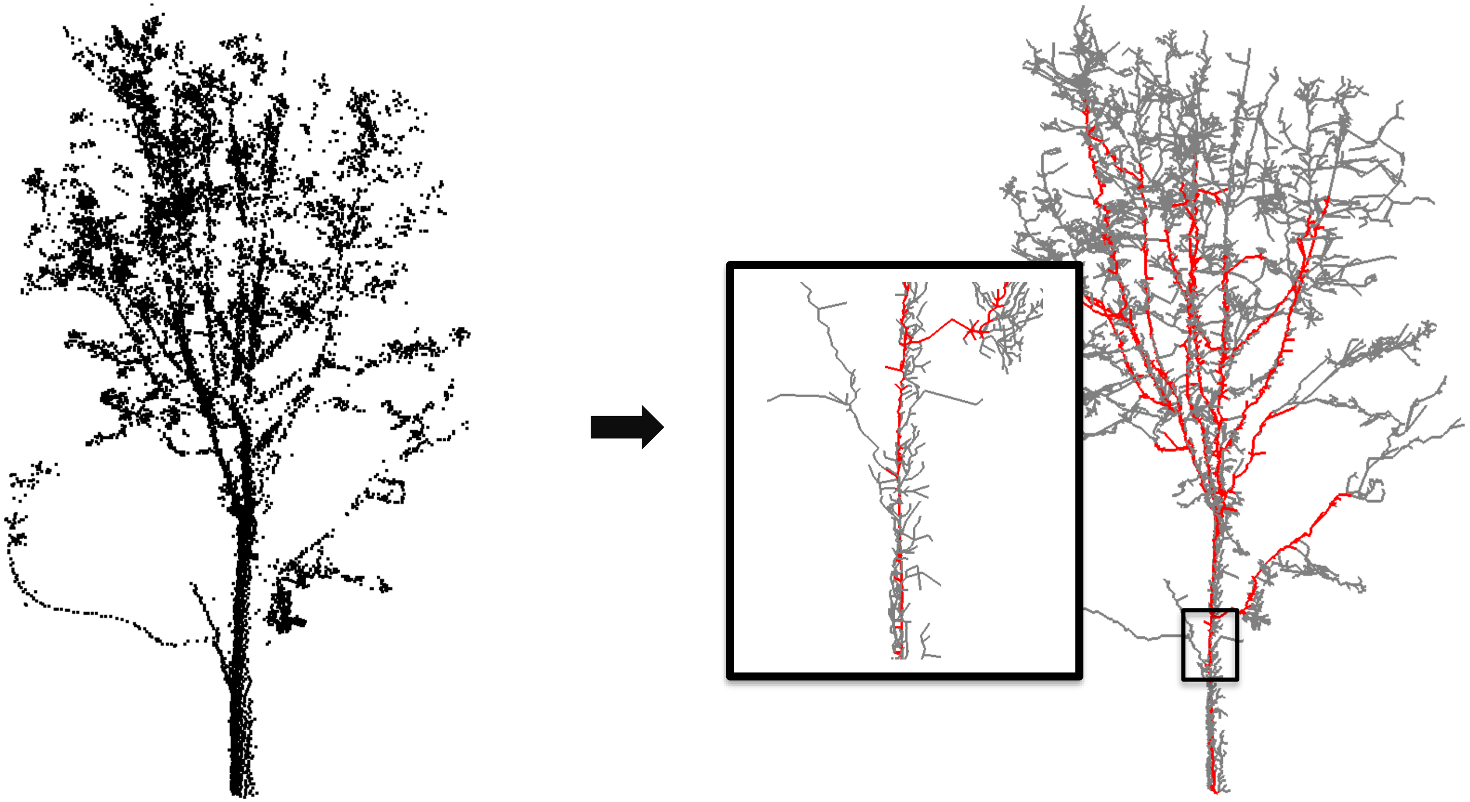
- We obtain initial tree skeleton from the minimum spanning tree
  - Obtain the initial skeleton



Minimum spanning tree   Initial skeleton

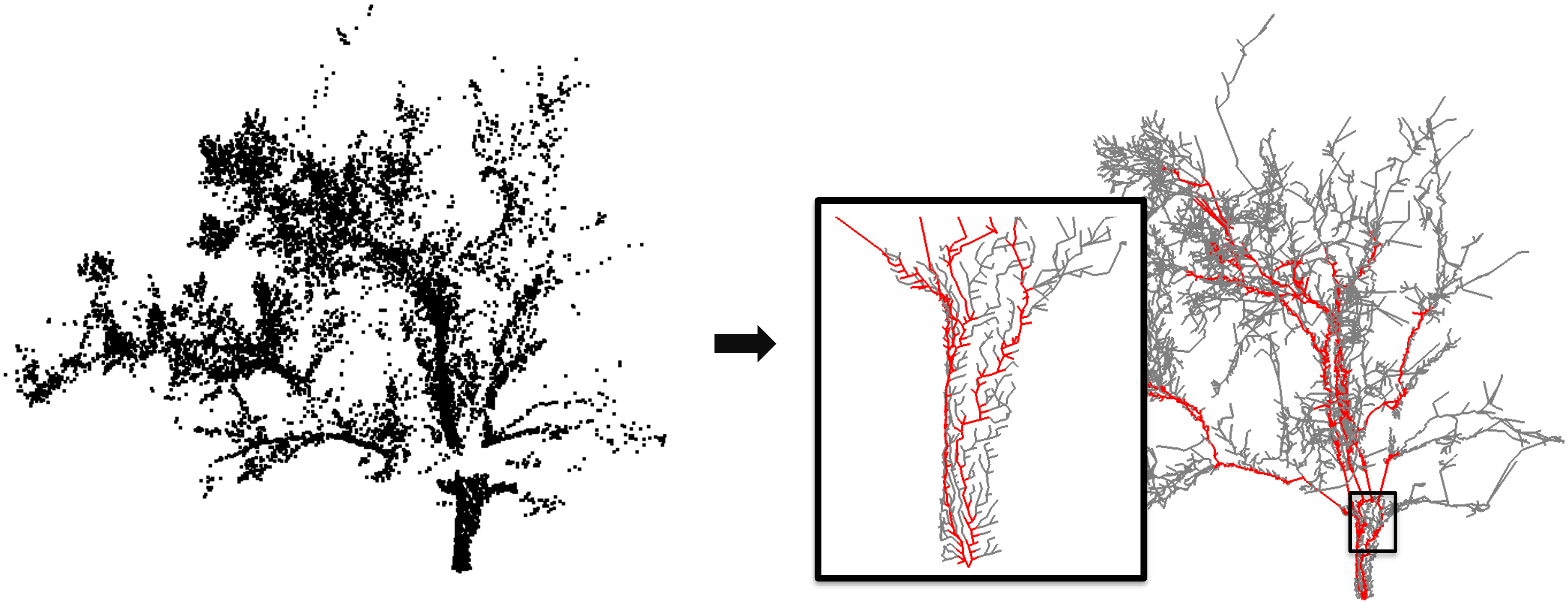
3.1 Skeleton Initialization

- An example of a well-extracted initial skeleton



3.1 Skeleton Initialization

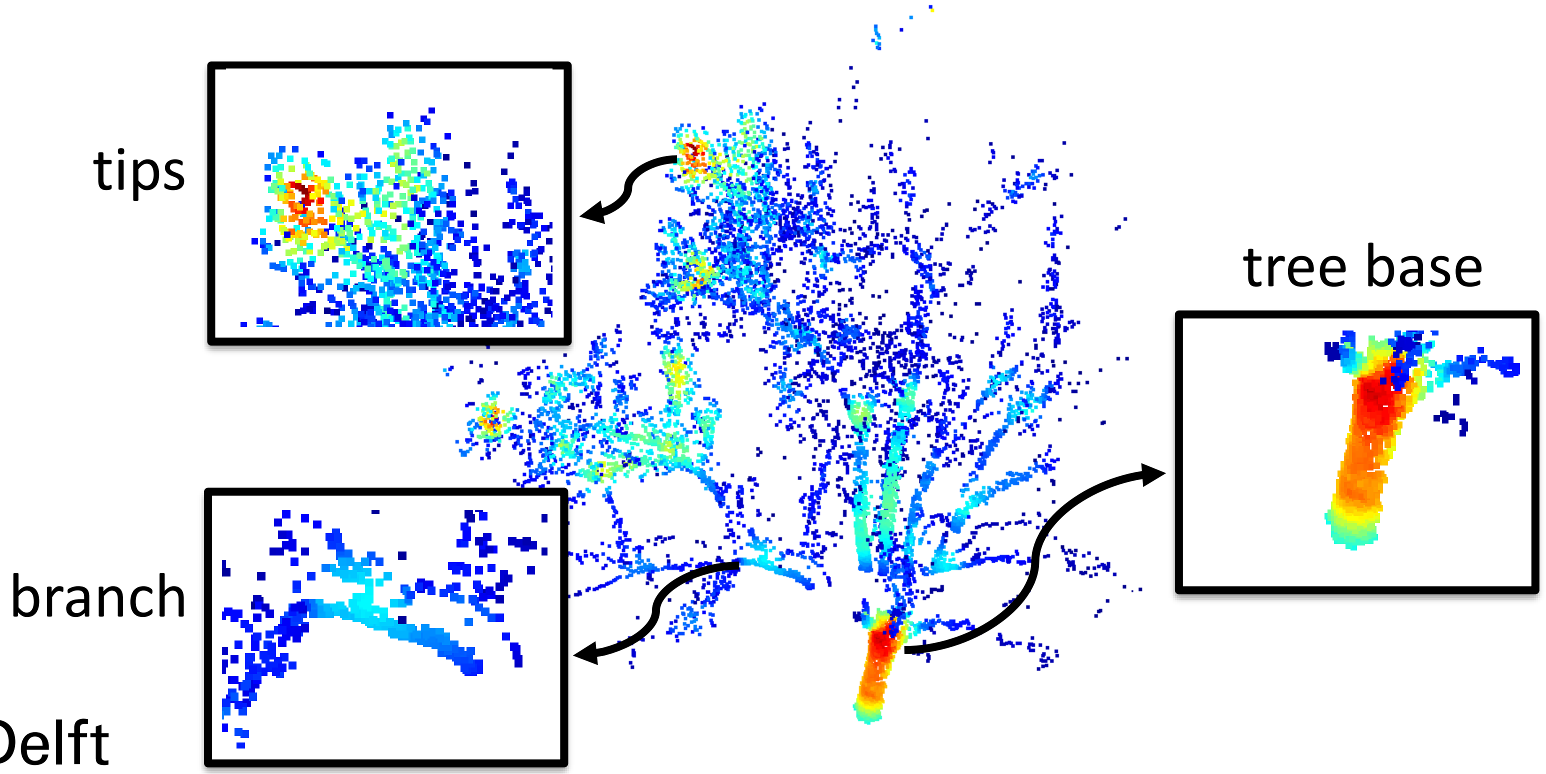
- An example of a badly-extracted initial skeleton



TUDelft

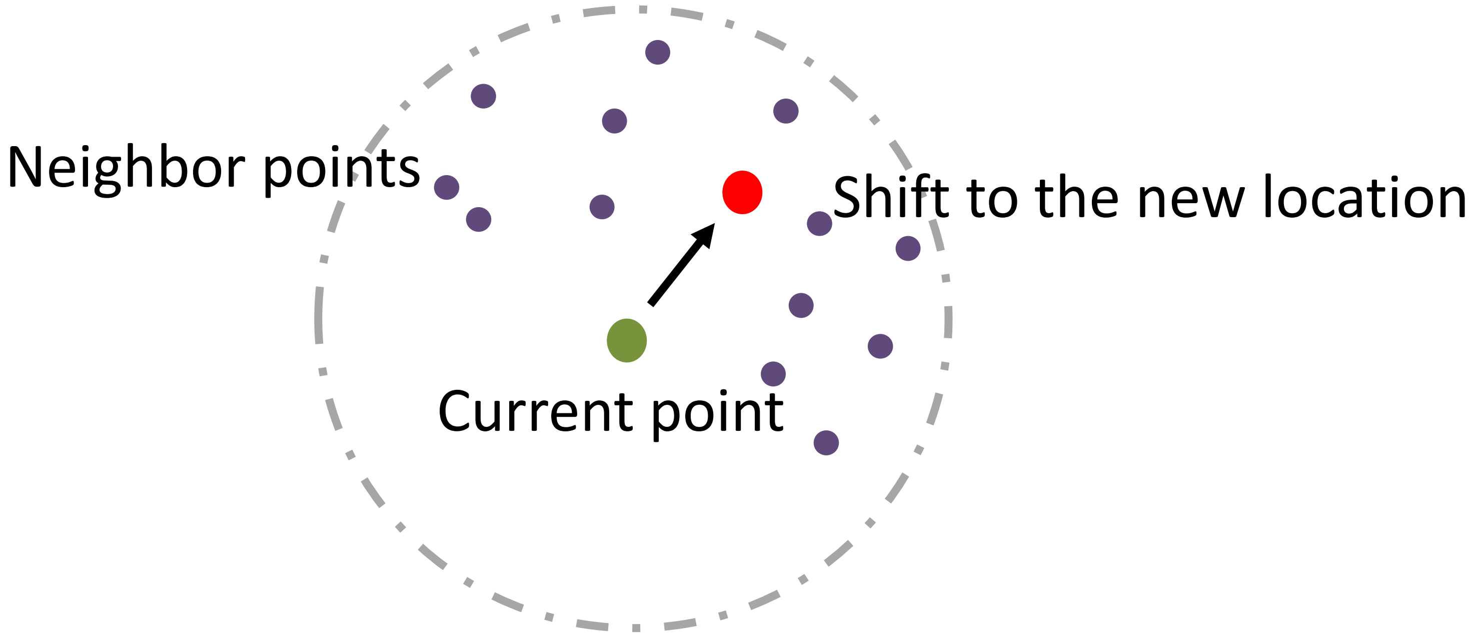
3.1 Skeleton Initialization

- We address the problem by centralizing main-branch points, to generate condensed branches
  - Identify main-branch points



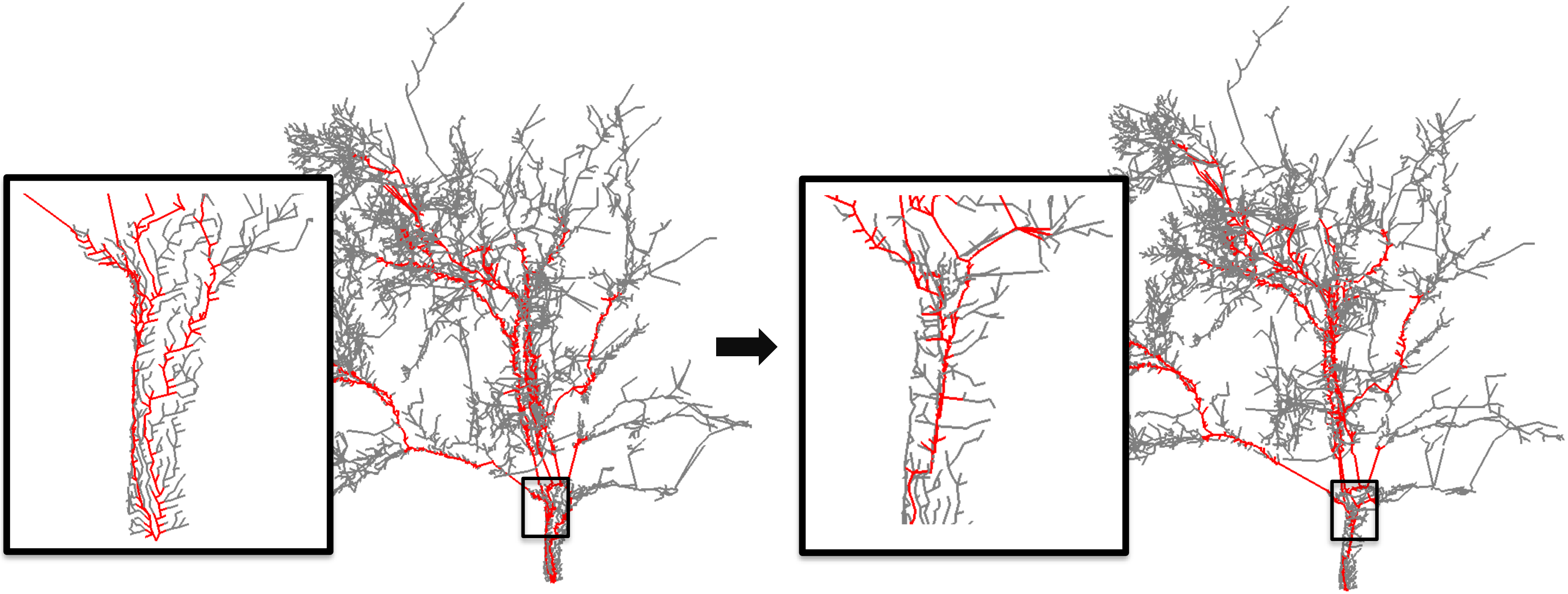
3.1 Skeleton Initialization

- We address the problem by centralizing main-branch points, to generate condensed branches
  - centralize main-branch points

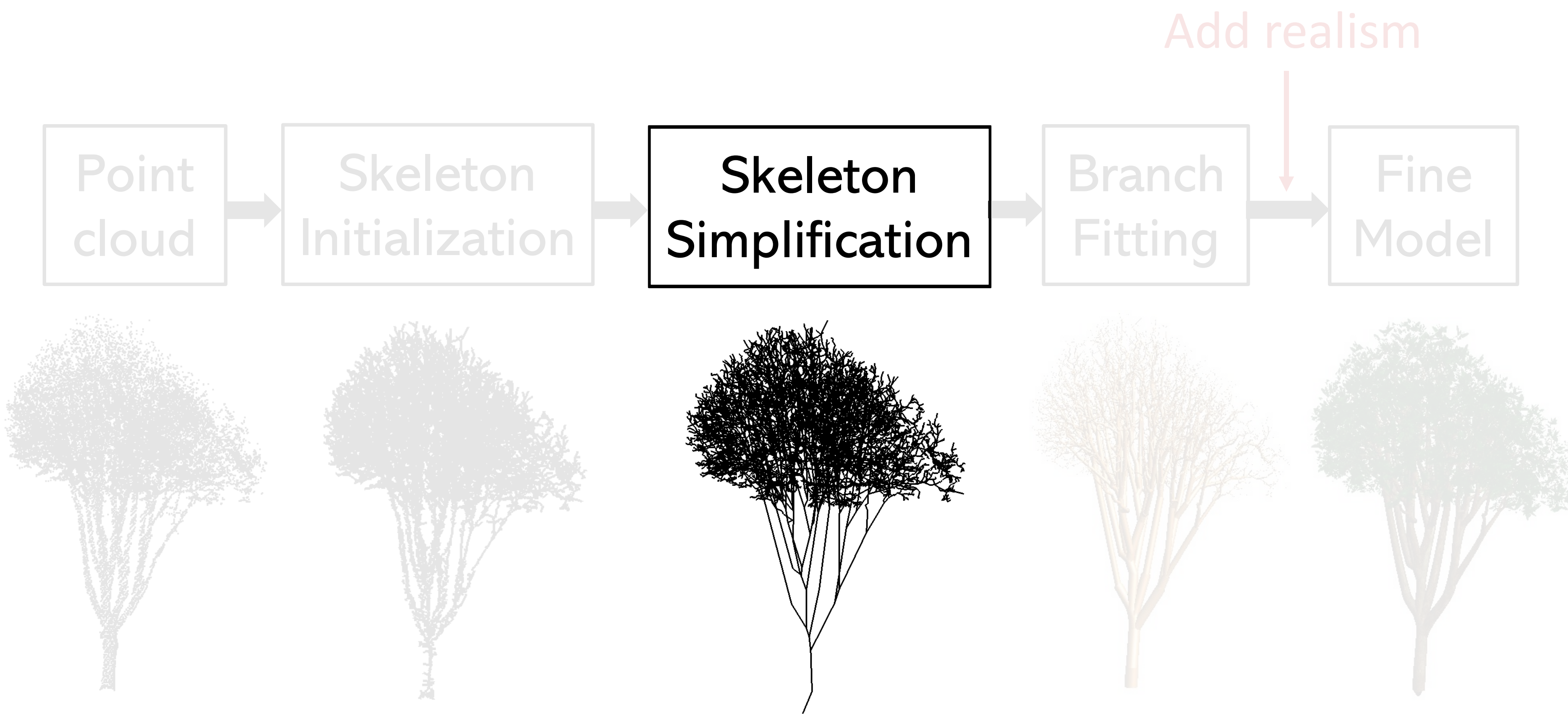


3.1 Skeleton Initialization

- We address the problem by centralizing main-branch points, to generate condensed branches

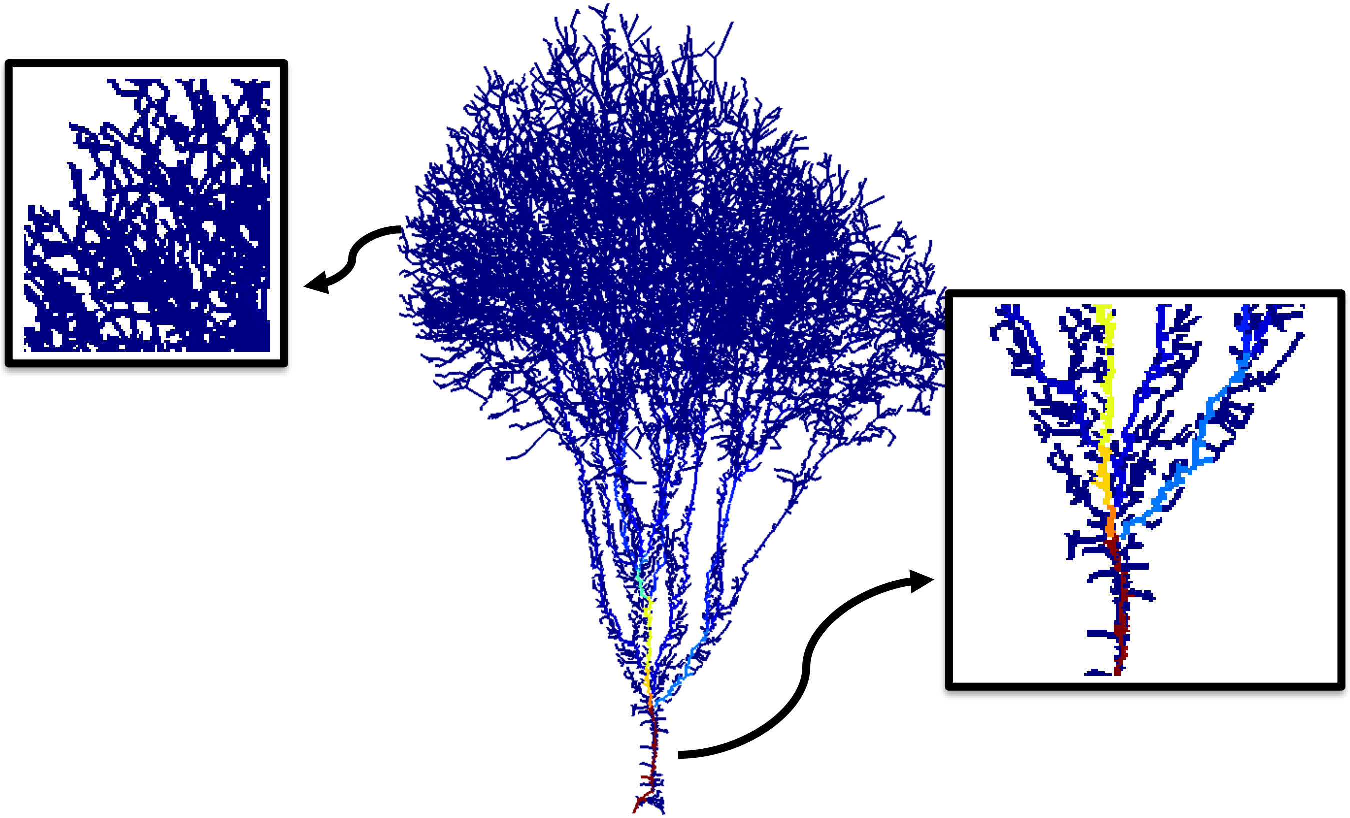


3. Methodology Overview



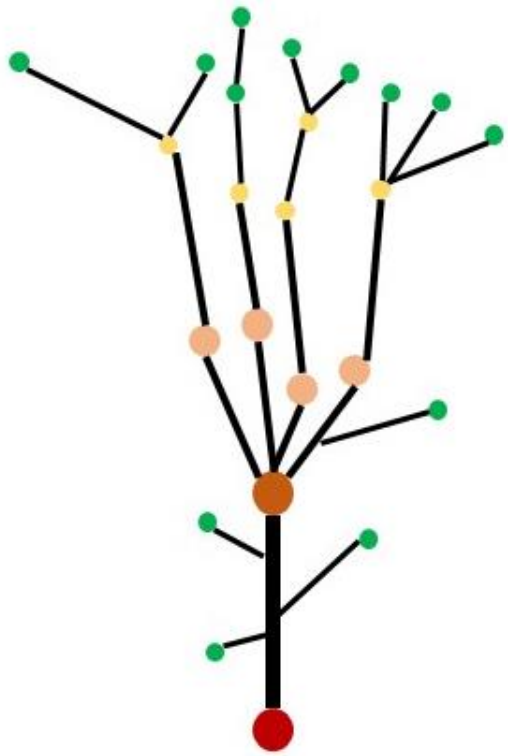
3.2 Skeleton Simplification

- Weight vertices and edges with subtree lengths



3.2 Skeleton Simplification

- Eliminate small noisy branches from weights

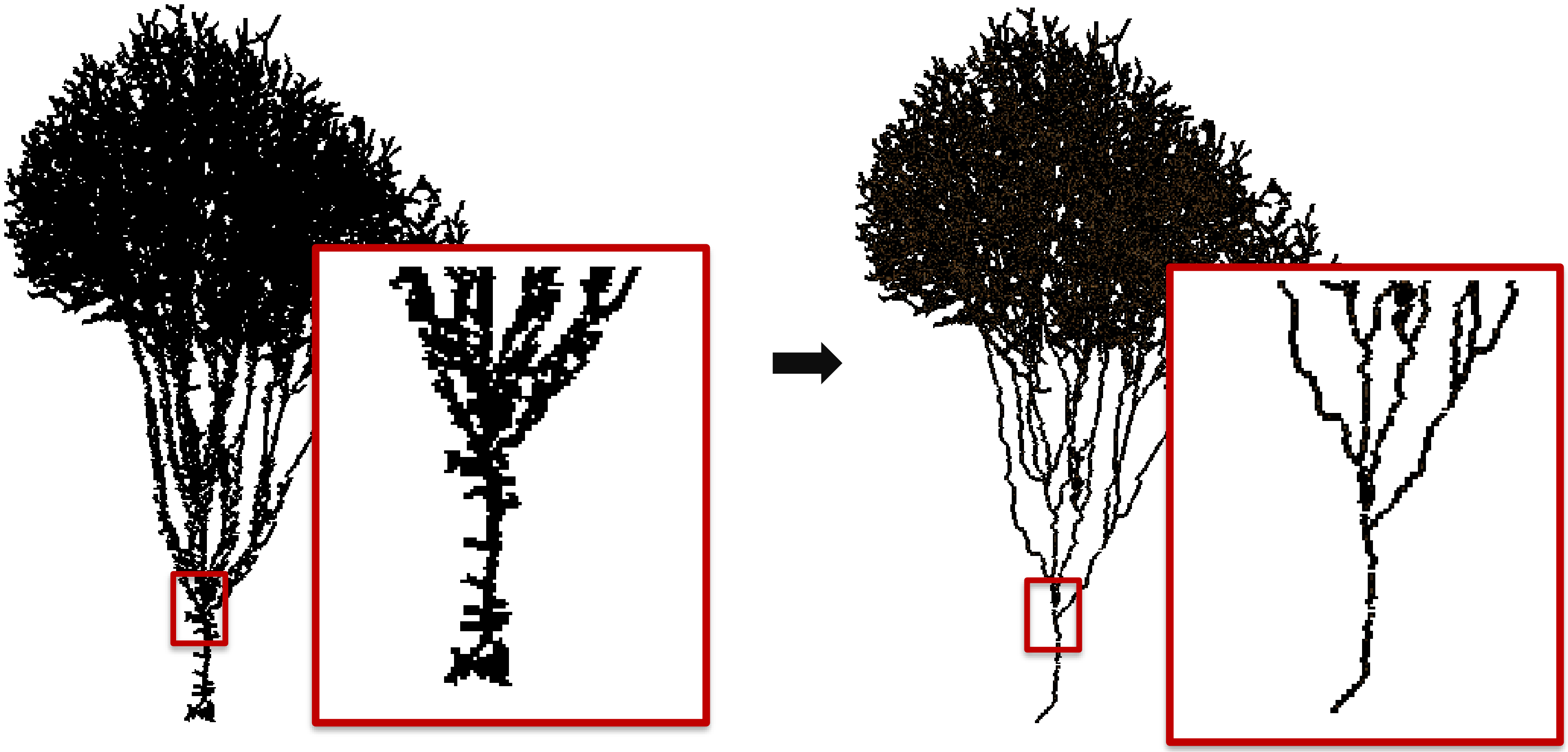


- $\delta w = {}^{w_i}/_{w_p}$
- For the i^{th} vertex:
  - If $\delta w < \tau$:
  - Then remove i^{th} vertex and its subtree

TU Delft

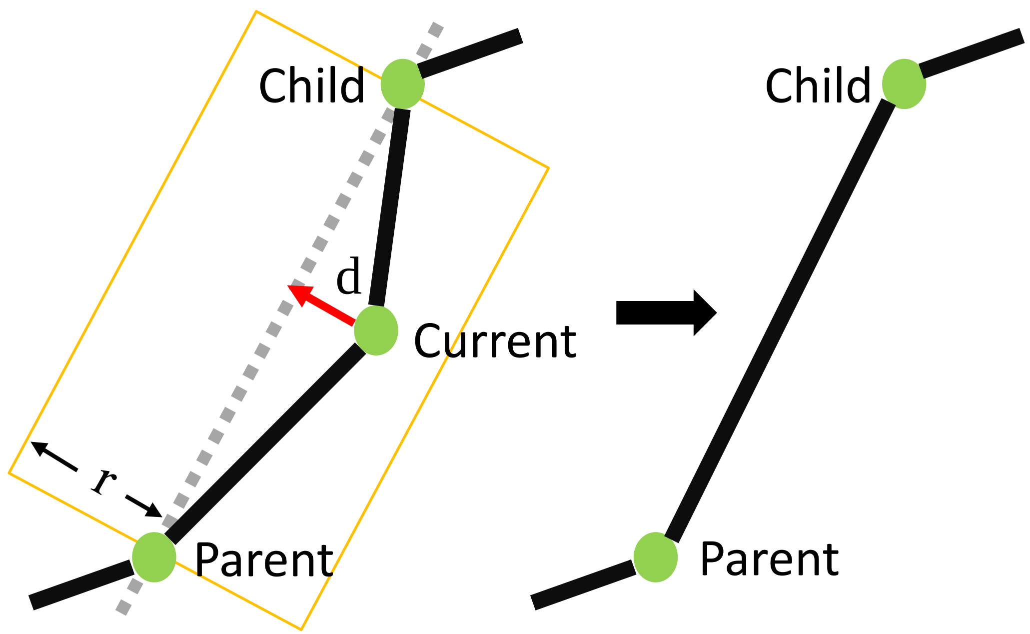
3.2 Skeleton Simplification

- Eliminate small noisy branches from weights



3.2 Skeleton Simplification

- Iterative simplification
  - Single-child vertex simplification

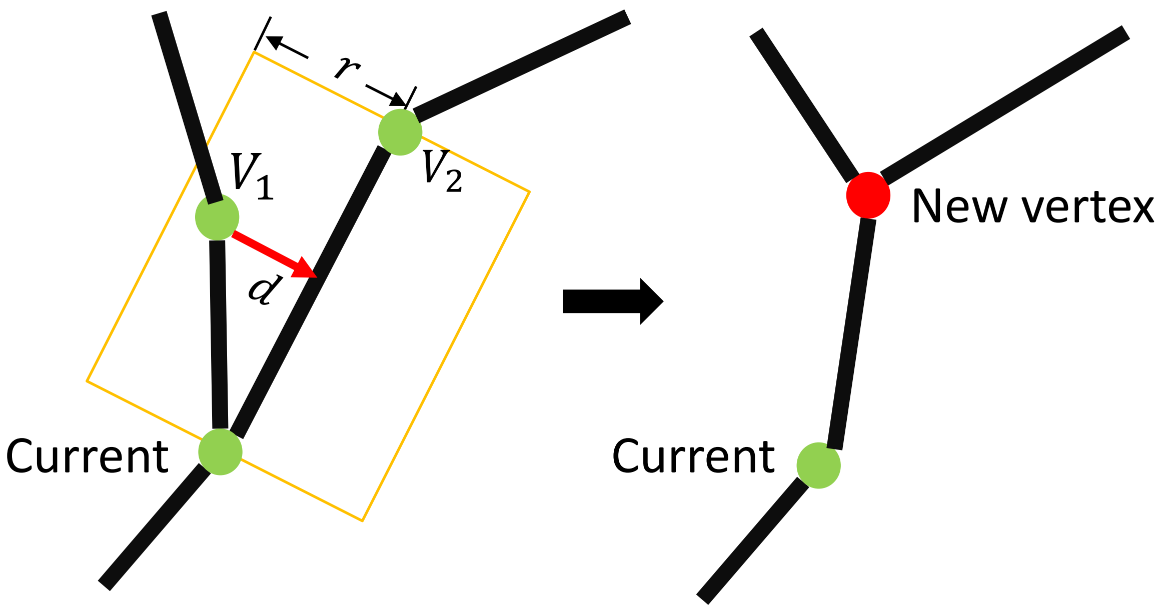


- If $\frac{d}{r} < \sigma$:
- Then clear current vertex

TUDelft

3.2 Skeleton Simplification

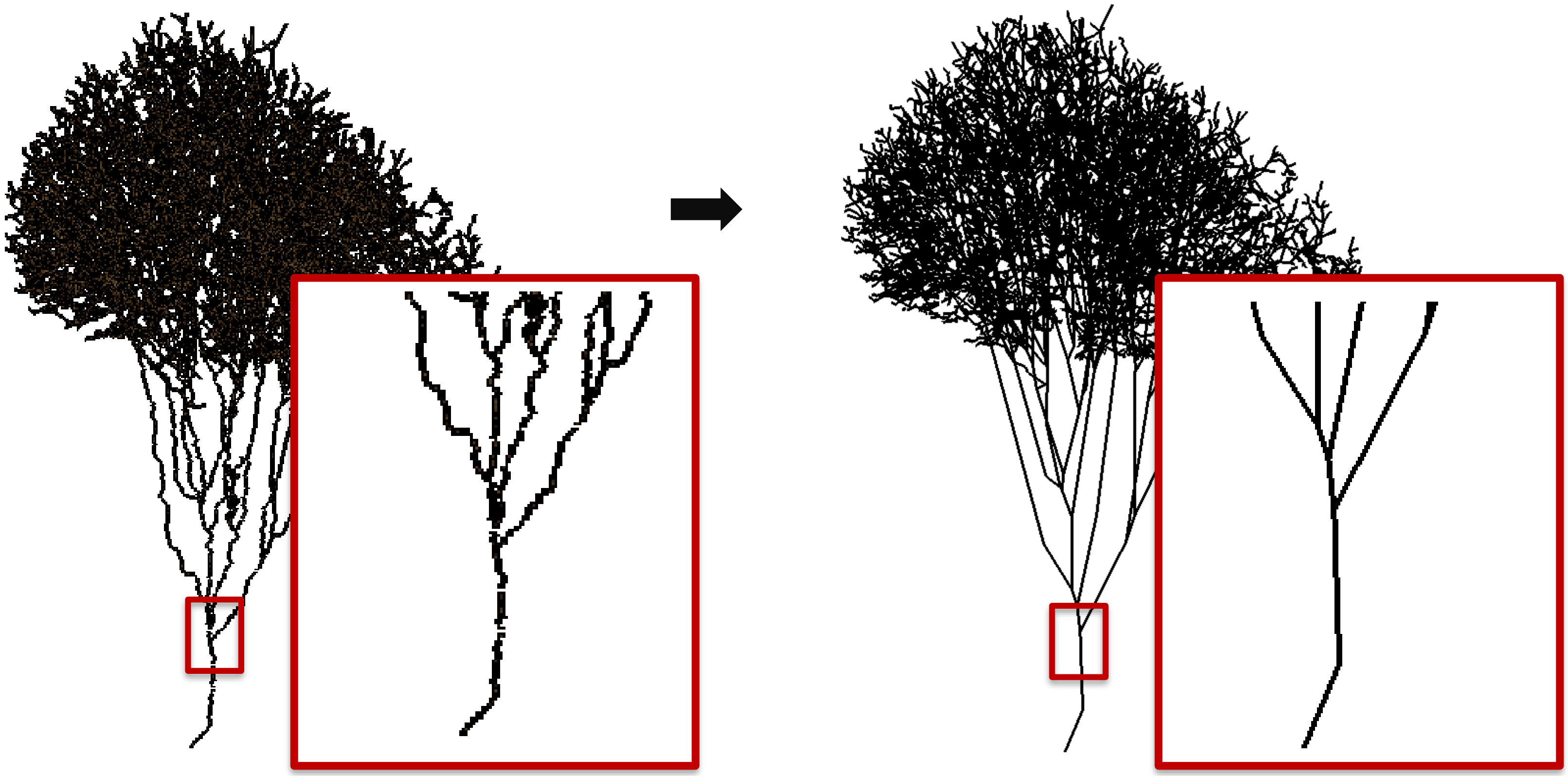
- Iterative simplification
  – Multiple-children vertex simplification



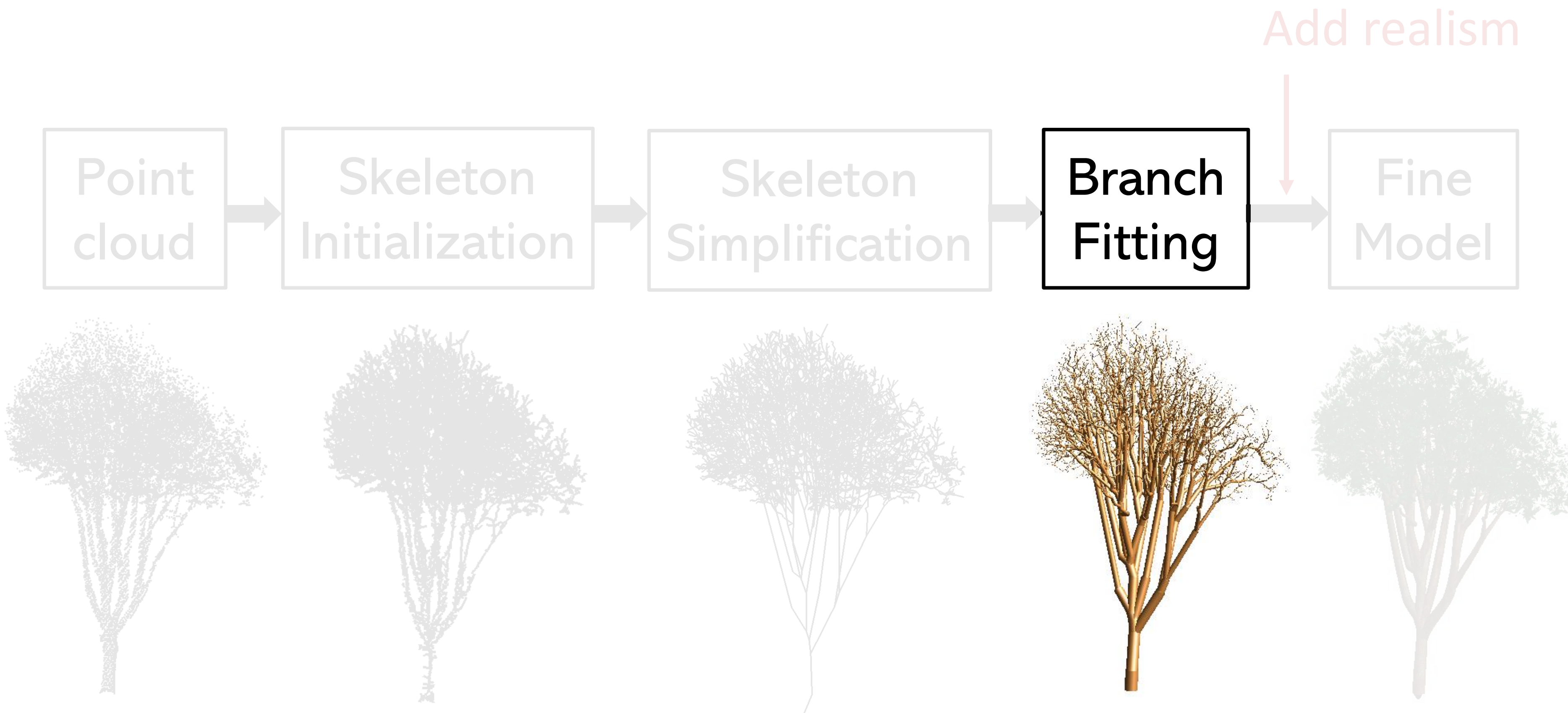
- If $\frac{d}{r} < \sigma$:
- Then merge current children

3.2 Skeleton Simplification

- Iterative simplification



3. Methodology Overview

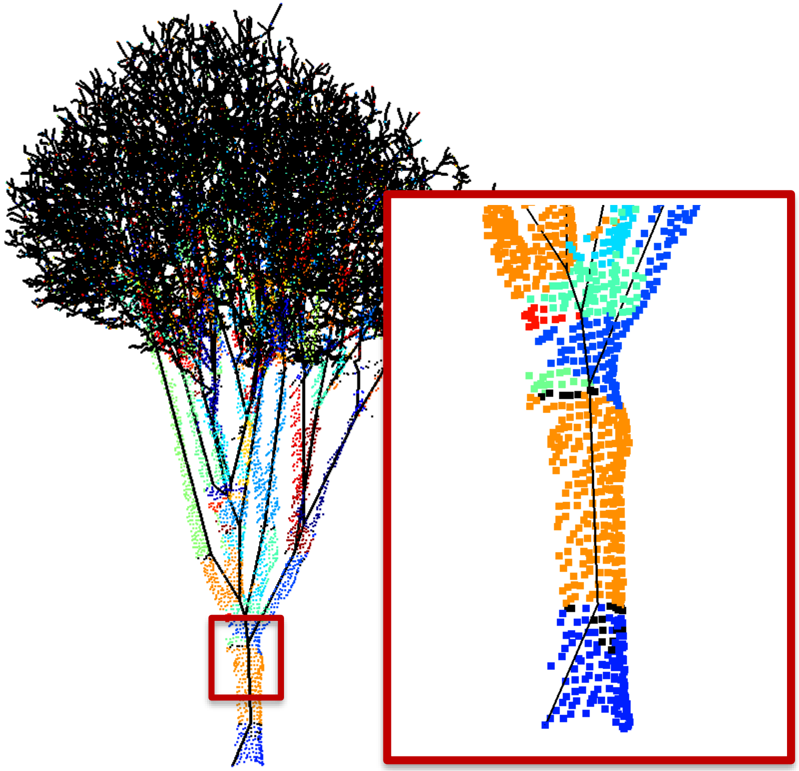


3.3 Branch fitting

- Fit cylinders to approximate the branch geometry

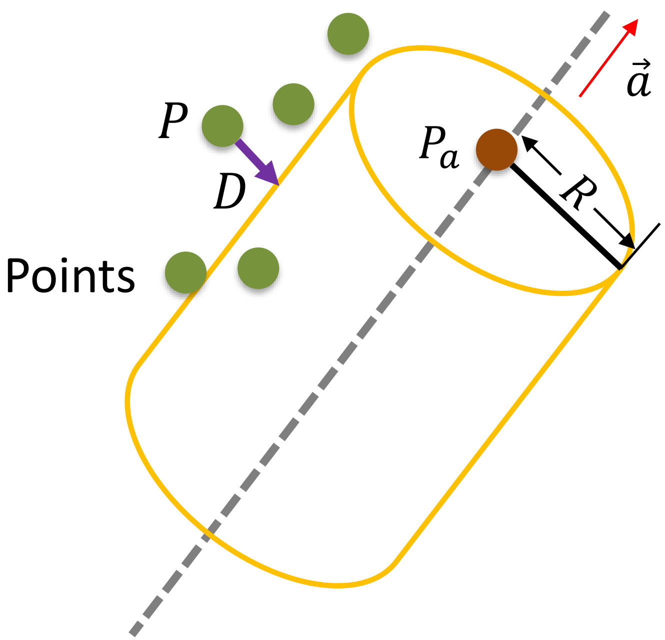
3.3 Branch fitting

- Fit cylinders to approximate the branch geometry
  - Assign each branch with corresponding points



3.3 Branch fitting

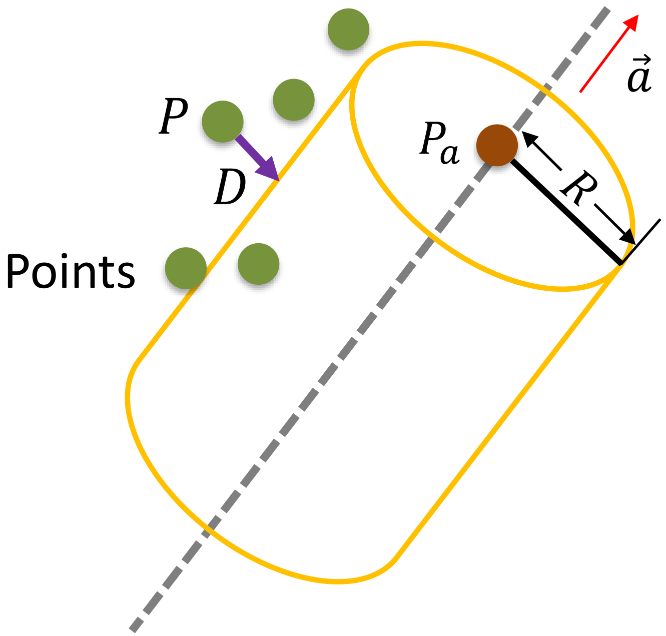
- Fit cylinders to approximate the branch geometry
  - Non-linear least squares problem (Levenberg–Marquardt algorithm)



- **Parameters to solve:**
  - P_a, R, $\vec{a}$
- **Input data:**
  - Position of P
- **Objective function:**
  - min $\sum D$

3.3 Branch fitting

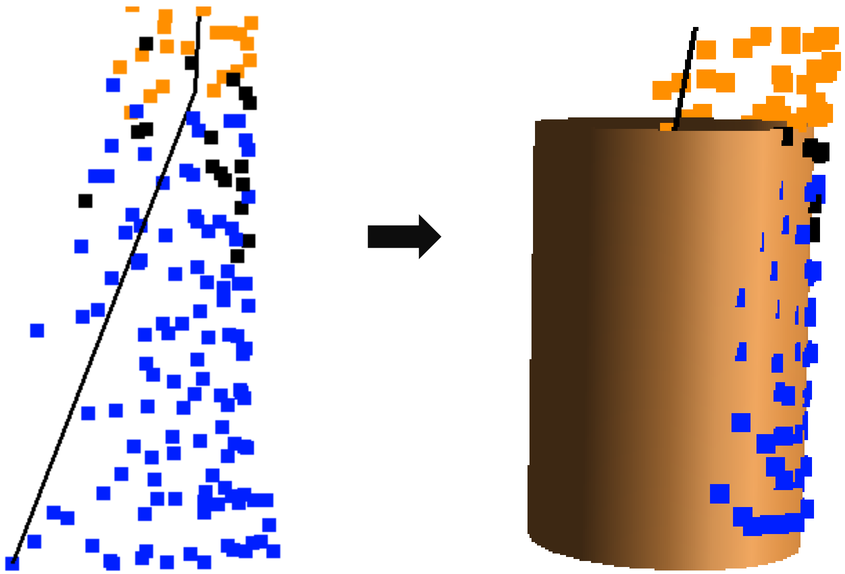
- Fit cylinders to approximate the branch geometry
  - 2^{nd} iteration: Weighted non-linear least squares



- **Weight:**
  - $w_i = 1 - \frac{D_i}{D_{max}}$
- **Objective function:**
  - $\min \sum D_i w_i$

3.3 Branch fitting

- Fit cylinders to approximate the branch geometry
  - Fit a cylinder first for the main trunk



3.3 Branch fitting

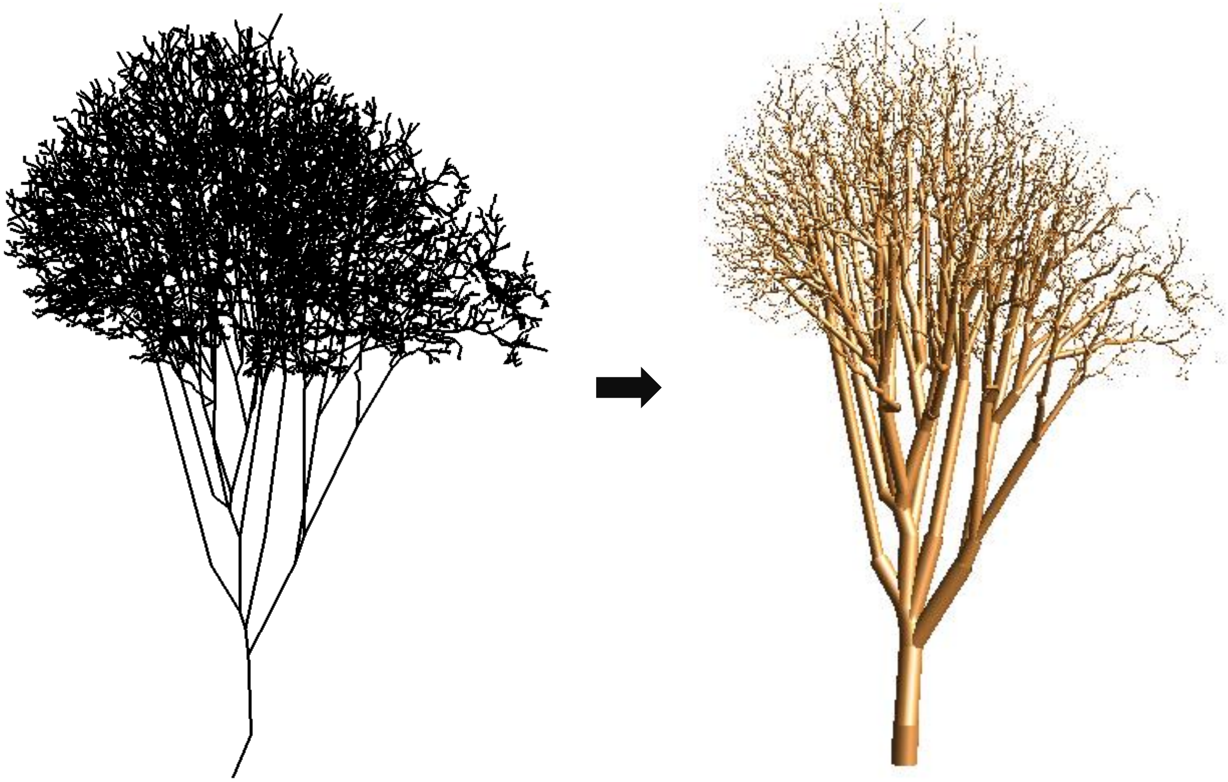
- Fit cylinders to approximate the branch geometry
  - Small branches don't have enough points to fit
  - We derive the radius by:

$$r_{e_i} = r_t \frac{w_{e_i}}{\sum_j w_{e_j}}$$

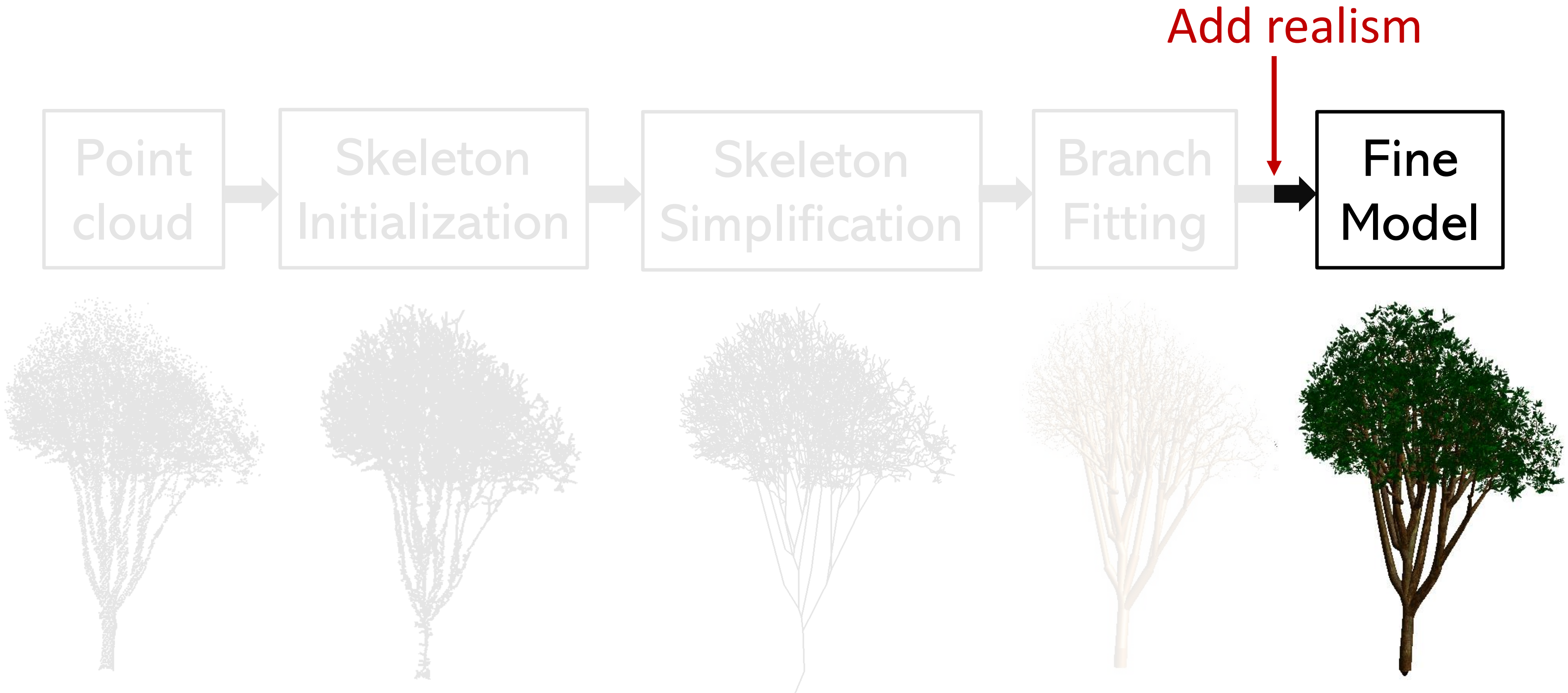
r is the radius, e_i is the i^{th} branch, t is the trunk, w is the weight of the corresponding branch edge.

3.3 Branch fitting

- Fit cylinders to approximate the branch geometry



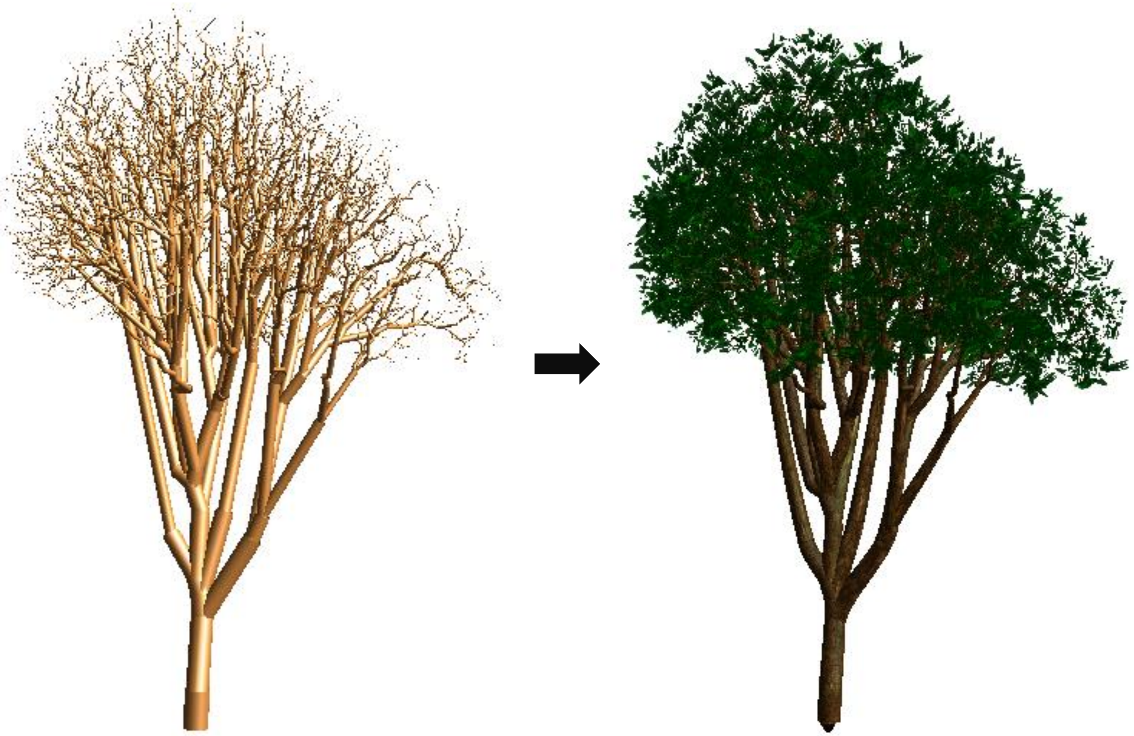
3. Methodology Overview



TU Delft

3.4 Adding realism

- Add leaves and texture to the tree model
  - Reconstructing leaves almost impossible
  - Randomly grow leaves on top of branches



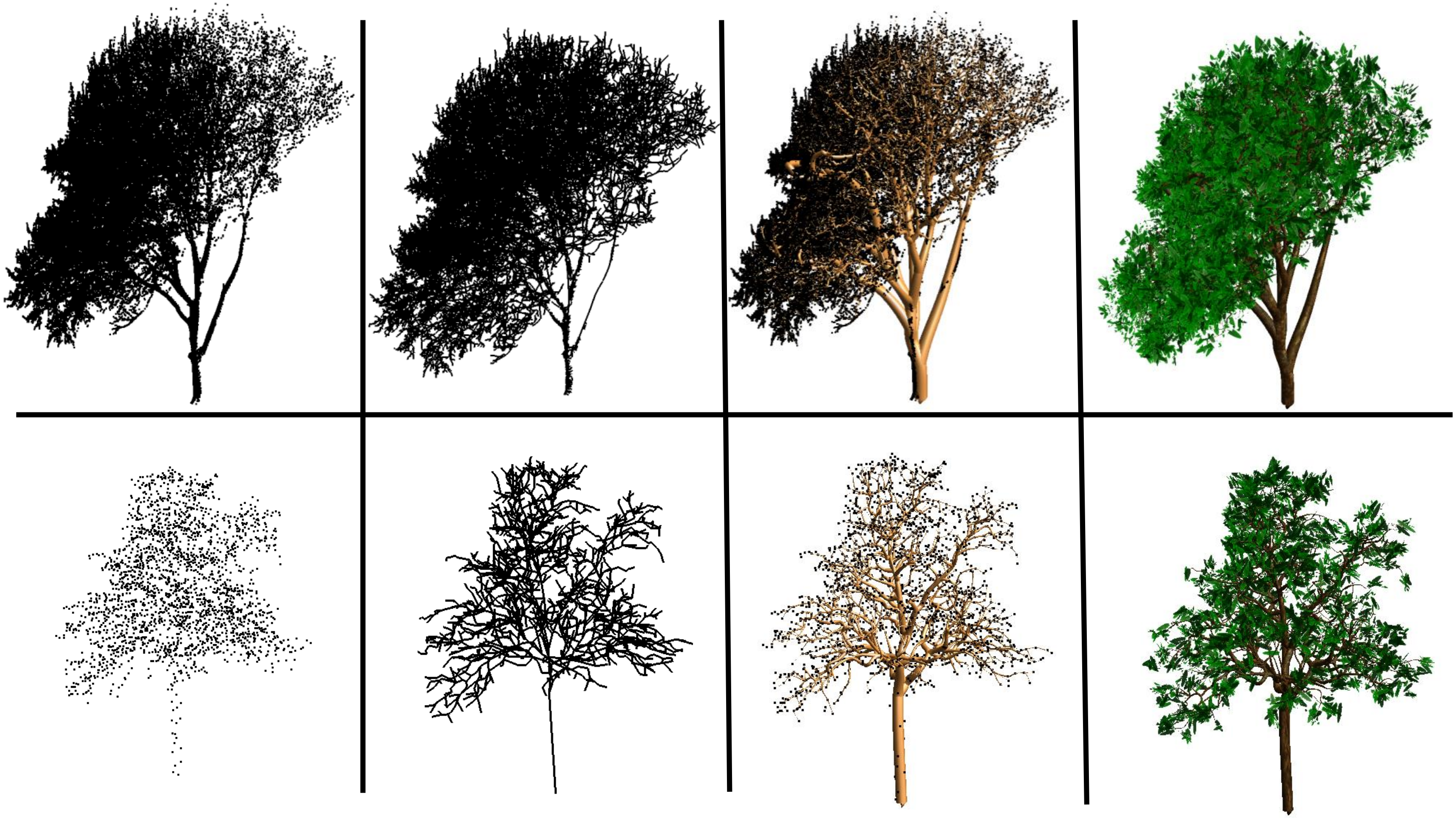
TU Delft

4. Results & Discussion

4.1 Results: different tree types

Input points skeleton branches fine model

4.1 Results: different data sources



Input points skeleton branches fine model

TUDelft

4.1 Results: model vs real tree

TUDelft

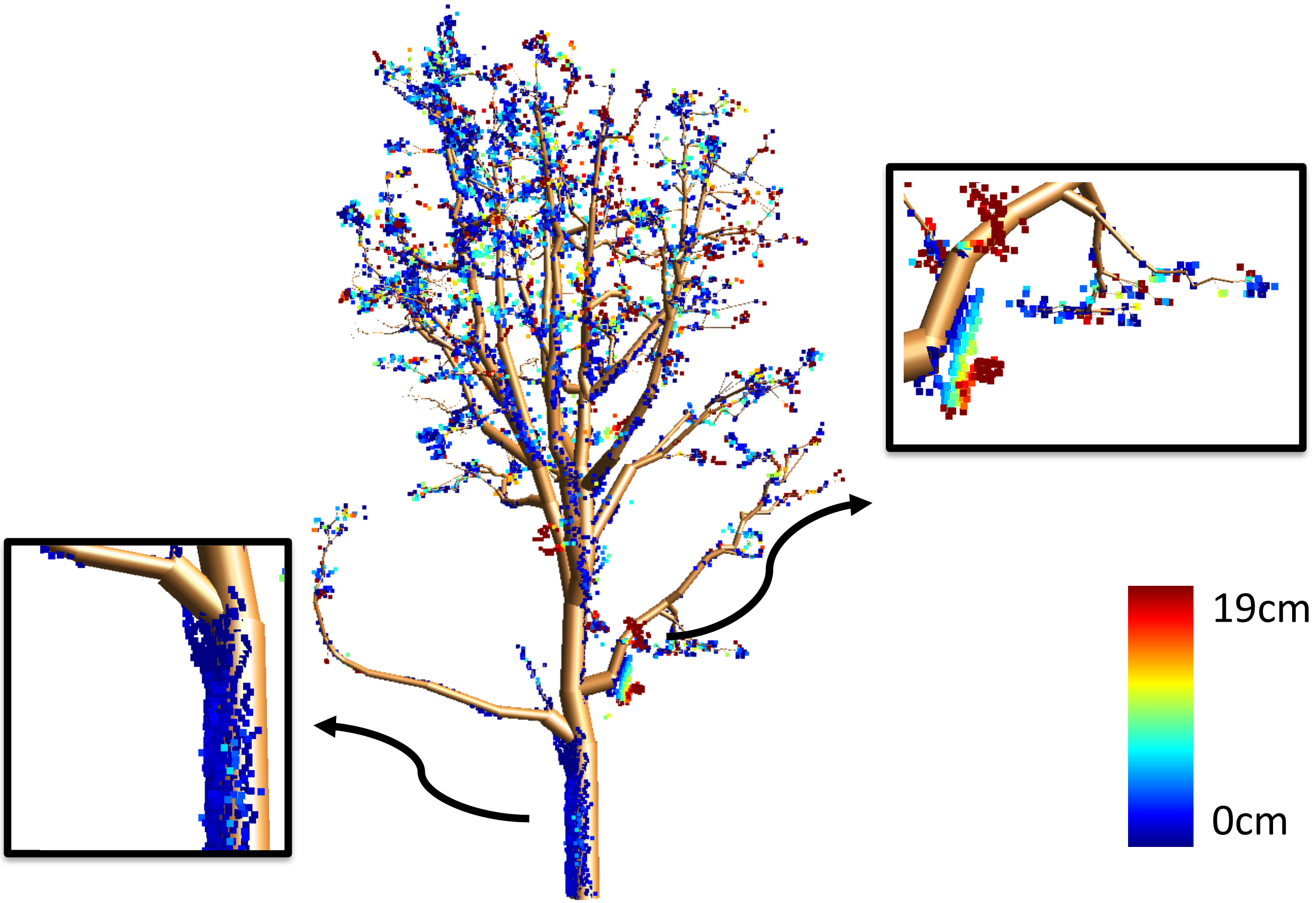
4.2 Evaluation: geometrical accuracy

- Measure the distance from points to the model

	Tree No.1	Tree No.2	Tree No.3
			
Height	5.61m	10.05m	10.61m
Mean distance	2.76cm	10.04cm	6.25cm
Standard deviation	2cm	8cm	6cm

4.2 Evaluation: geometrical accuracy

- Error visualization

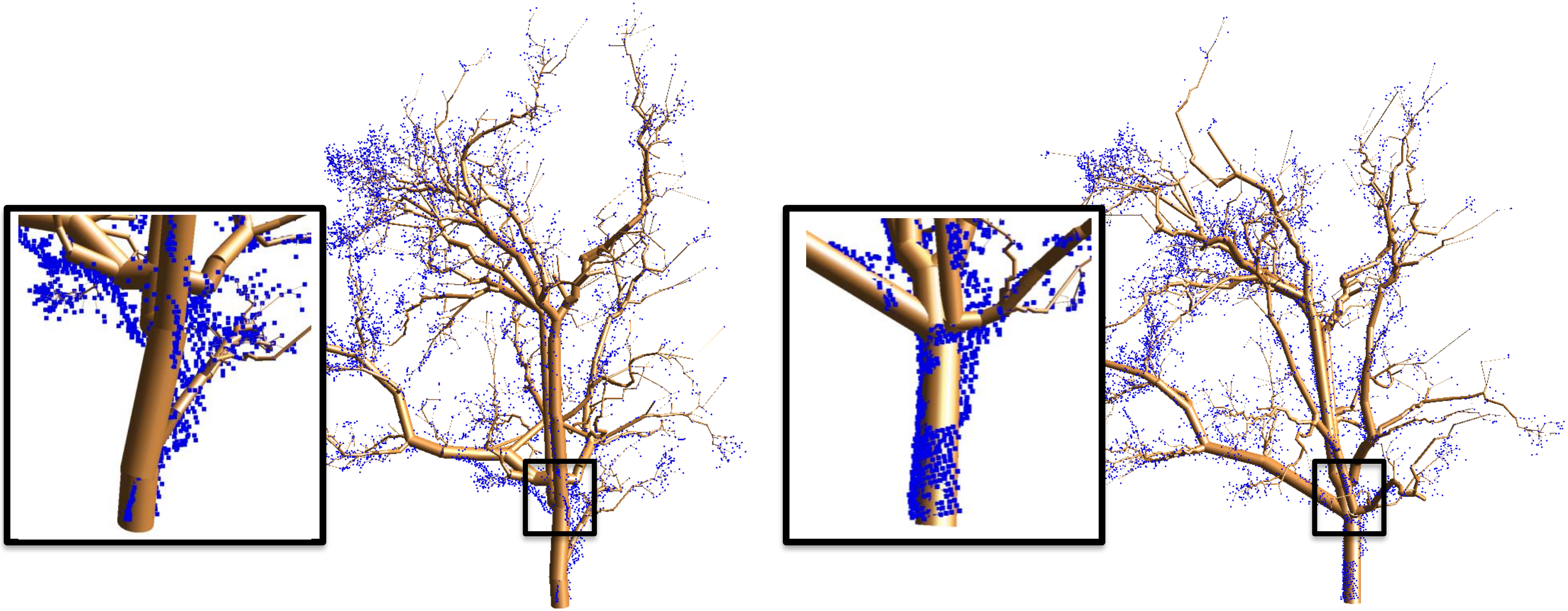


4.3 Evaluation: computation efficiency

	Tree No.1 	Tree No.2 	Tree No.3 	Tree No.4 
Point count	2488	11855	28993	137407
Triangulation	0.152s	0.753s	1.652s	9.006s
Skeleton initialization	0.043s	0.195s	0.583s	3.35s
Simplification	0.013s	0.037s	0.096s	0.475s
Branch fitting	0.014s	0.072s	0.099s	0.521s
Rendering	1.215s	0.501s	4.252s	12.965s

4.4 Comparison

- Improve topological fidelity



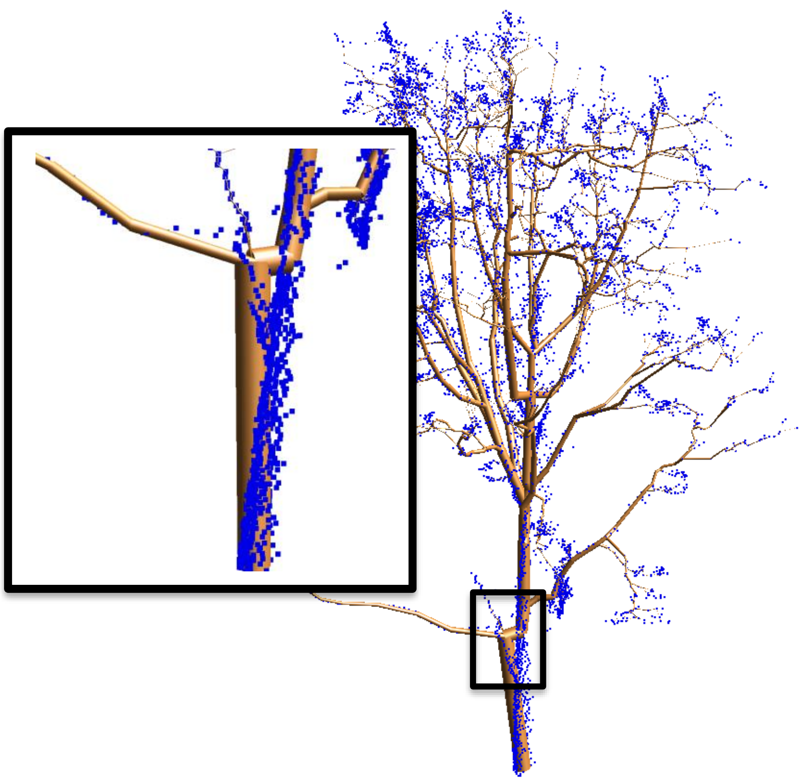
Livny's method

Our method

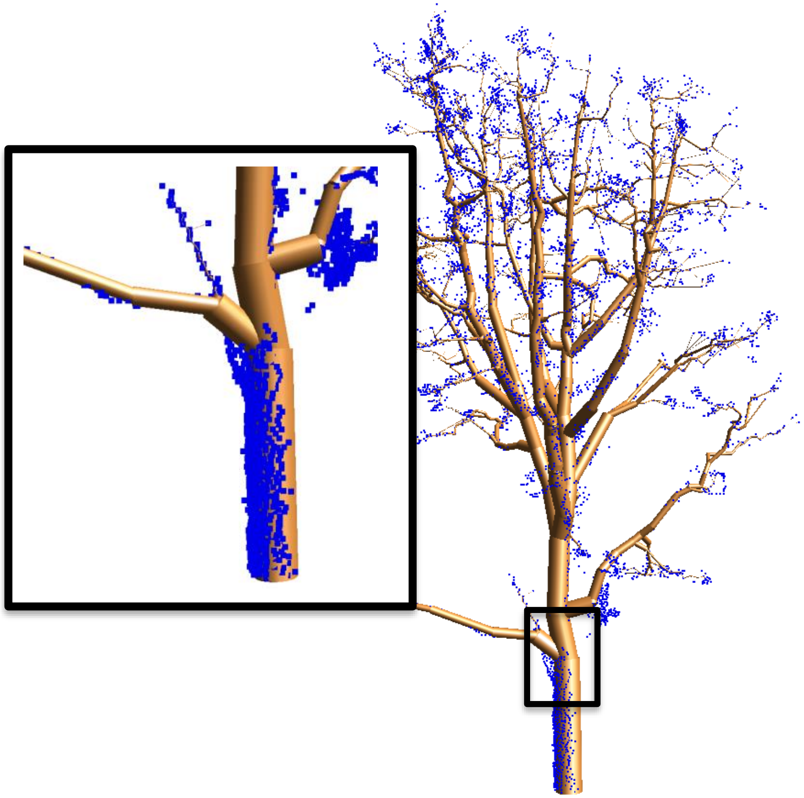
TU Delft

4.4 Comparison

- Improve geometrical accuracy



Livny’s method

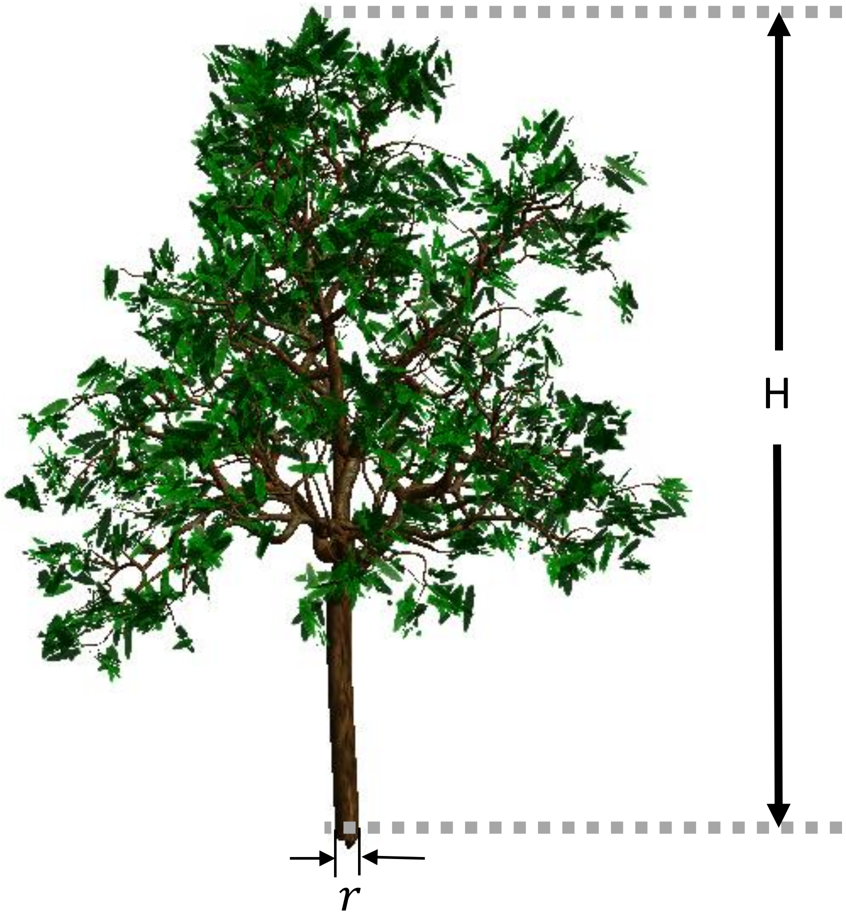


Our method

TU Delft

4.5 Applications

- Compute tree height & trunk thickness



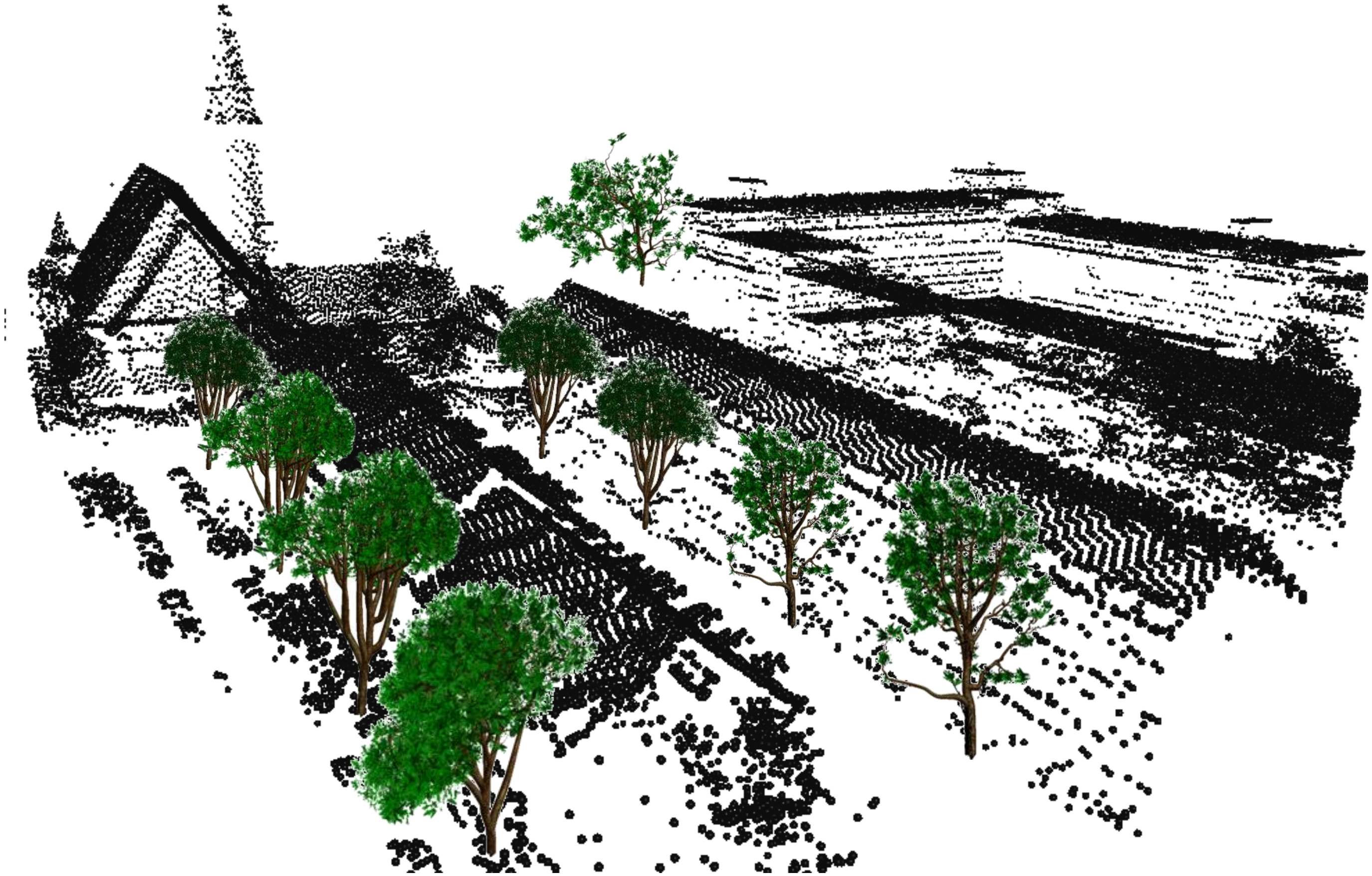
4.5 Applications

- Estimate wood volume, biomass..

TU Delft

4.5 Applications

- Enhance realism in urban scenes



4.6 Limitations

- Data-driven mainly
- Not involving the natural growing rules of tree branches

5. Conclusions & Future Work

5.1 Conclusions

- Fully-automatic
- Widely applicable to various trees
- Able to achieve high modelling quality from static scanning data and mobile scanning data
- Able to generate plausible results from airborne scanning data

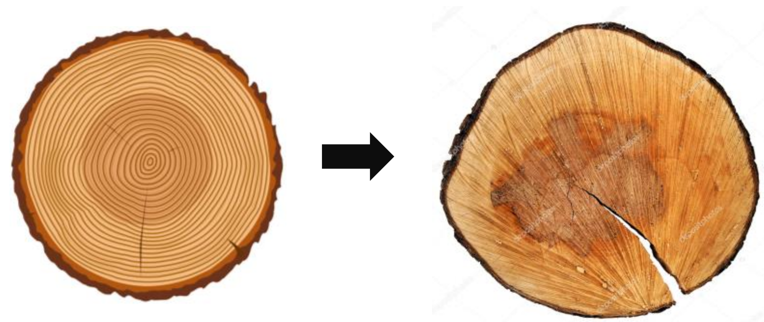
5.2 Future Work

- Individual tree

☞ A set of trees

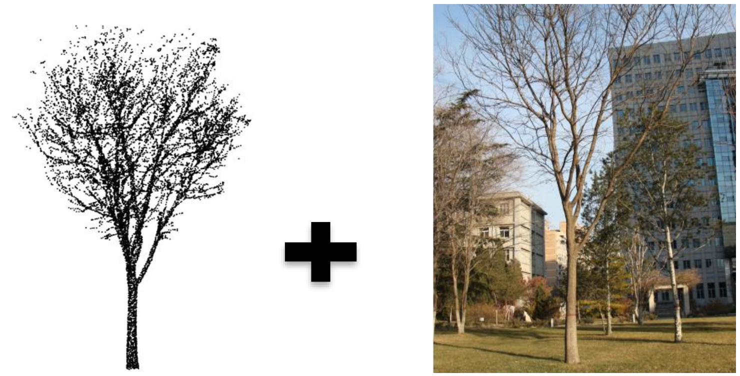
- Cylinder fitting

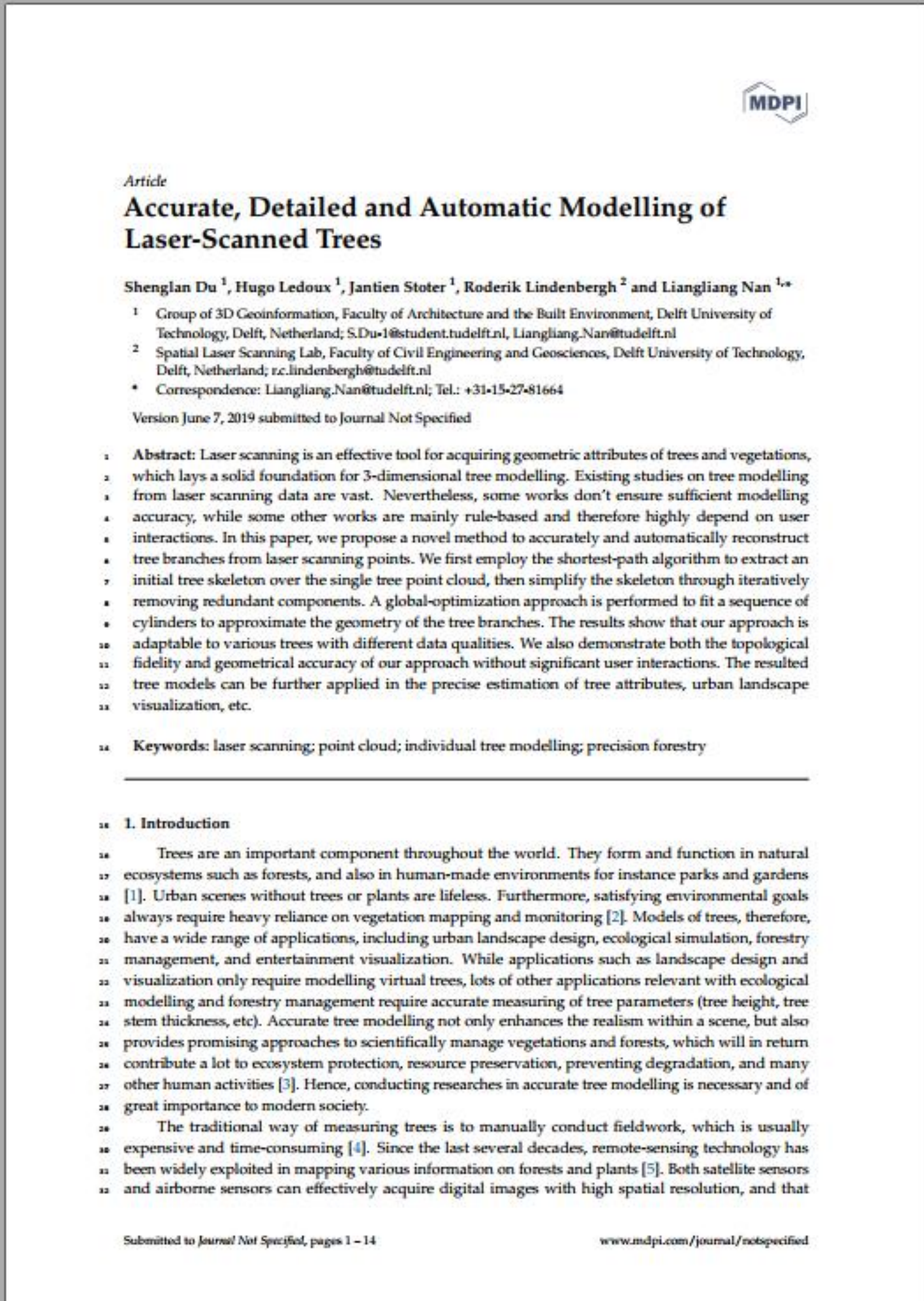
☞ Free form surface fitting



- Point cloud only

☞ Points with Images




MDPI

Article

Accurate, Detailed and Automatic Modelling of Laser-Scanned Trees

Shenglan Du [1], Hugo Ledoux [1], Jantien Stoter [1], Roderik Lindenbergh [2] and Liangliang Nan [1,*]

[1] Group of 3D Geoinformation, Faculty of Architecture and the Built Environment, Delft University of Technology, Delft, Netherland; S.Du-1@student.tudelft.nl, Liangliang.Nan@tudelft.nl
[2] Spatial Laser Scanning Lab, Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft, Netherland; r.c.lindenbergh@tudelft.nl
* Correspondence: Liangliang.Nan@tudelft.nl; Tel.: +31-15-27-81664

Version June 7, 2019 submitted to Journal Not Specified

Abstract: Laser scanning is an effective tool for acquiring geometric attributes of trees and vegetations, which lays a solid foundation for 3-dimensional tree modelling. Existing studies on tree modelling from laser scanning data are vast. Nevertheless, some works don't ensure sufficient modelling accuracy, while some other works are mainly rule-based and therefore highly depend on user interactions. In this paper, we propose a novel method to accurately and automatically reconstruct tree branches from laser scanning points. We first employ the shortest-path algorithm to extract an initial tree skeleton over the single tree point cloud, then simplify the skeleton through iteratively removing redundant components. A global-optimization approach is performed to fit a sequence of cylinders to approximate the geometry of the tree branches. The results show that our approach is adaptable to various trees with different data qualities. We also demonstrate both the topological fidelity and geometrical accuracy of our approach without significant user interactions. The resulted tree models can be further applied in the precise estimation of tree attributes, urban landscape visualization, etc.

Keywords: laser scanning; point cloud; individual tree modelling; precision forestry

1. Introduction

Trees are an important component throughout the world. They form and function in natural ecosystems such as forests, and also in human-made environments for instance parks and gardens [1]. Urban scenes without trees or plants are lifeless. Furthermore, satisfying environmental goals always require heavy reliance on vegetation mapping and monitoring [2]. Models of trees, therefore, have a wide range of applications, including urban landscape design, ecological simulation, forestry management, and entertainment visualization. While applications such as landscape design and visualization only require modelling virtual trees, lots of other applications relevant with ecological modelling and forestry management require accurate measuring of tree parameters (tree height, tree stem thickness, etc). Accurate tree modelling not only enhances the realism within a scene, but also provides promising approaches to scientifically manage vegetations and forests, which will in return contribute a lot to ecosystem protection, resource preservation, preventing degradation, and many other human activities [3]. Hence, conducting researches in accurate tree modelling is necessary and of great importance to modern society.

The traditional way of measuring trees is to manually conduct fieldwork, which is usually expensive and time-consuming [4]. Since the last several decades, remote-sensing technology has been widely exploited in mapping various information on forests and plants [5]. Both satellite sensors and airborne sensors can effectively acquire digital images with high spatial resolution, and that