TAMPERE UNIVERSITY OF TECHNOLOGY

MARKKU ÅKERBLOM

Quantitative tree modeling from laser scanning data

Master of Science Thesis

Examiners:
Professor Mikko Kaasalainen,
D.Sc. (Tech.) Pasi Raumonen

Examiners and topic approved in the Science and Environmental Engineering Faculty Council meeting on 08 February 2012

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Teknis-luonnontieteellinen koulutusohjelma
MARKKU ÅKERBLOM: Quantitative tree modeling from laser scanning data
Diplomityö, 88 sivua
Toukokuu 2012
Pääaine: Matematiikka
Tarkastajat: Professori Mikko Kaasalainen, TkT Pasi Raumonen
Avainsanat: sovellettu matematiikka, inversio-ongelma, laserkeilaus, laserskannaus, pistepilvi

Maanpäällinen laserskannaus tarjoaa mahdollisuuden tehdä tarkkoja metsämittauksia esimerkiksi yksittäisistä puista. Mittaaminen on kuitenkin vasta alkua, sillä puusta saatavat mittaustulokset eivät itsessään ole useinkaan mielenkiintoisia vaan tuloksista jonkin algoritmin avulla lasketut ominaisuudet ovat todellinen mielenkiinnon kohde. Tässä opinnäytetyössä esitellään uusi menetelmä puumallin rekonstuktuointiin yksittäisestä puusta saatujen lasermittausten perusteella. Menetelmä perustuu topologisiin ominaisuuksiin sekä suhteellisiin mittoihin, mikä tekee siitä mittakaavasta riippumattoman ja nopean. Algoritmi tuottaa lopputuloksenaan kattavan ja kvantitatiivisen kolmiulotteisen sylinterimallin, joka kuvastaa mitatun puun pintaa ja tilavuutta. Sylinterimalli on kattava ja kvantitatiivinen siinä mielessä, että se sisältää jokaisen oksan koon ja paikan sekä tiedon oksien välilistä suhteista.

Tässä työssä esitellään algoritmi yleisellä tasolla sekä matemaattiset käsitteet, joiden varaan se rakentuu. Mukaan on sisällytetty myös useita esimerkkejä algoritmin käytöstä.

Abstract

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Science and Engineering
MARKKU ÅKERBLOM: Quantitative tree modeling from laser scanning data
Master of Science Thesis, 88 pages
May 2012
Major: Mathematics
Examiners: Professor Mikko Kaasalainen, D.Sc. (Tech.) Pasi Raumonen
Keywords: applied mathematics, inverse problem, laser scanning, point cloud

Terrestrial laser scanning provides accurate forest measurements, for example, from single trees. However, the analysis of such measurements requires complicated algorithms to find useful information about the scanned trees. In this thesis a new method for reconstructing a tree model from single tree laser scanning measurements is presented. The algorithm is based on topological properties and relative measures which makes it scale independent and fast. The output of the algorithm is a reconstruction of the tree surface and volume as a three-dimensional cylinder model. The resulting model is complete and quantitative in the sense that it contains accurate information about the sizes and locations of each of the branches and also information about the relations between branches.

This thesis describes the outline of the tree analysis algorithm and the mathematical concepts it is based on. Furthermore, several examples of the usage of the algorithm are provided.

Preface

This thesis has been written in the Department of Mathematics at the Tampere University of Technology. The research was financially supported by the Academy of Finland project *Modelling and applications of stochastic and regular surfaces in inverse problems*.

I am grateful to Professor Seppo Pohjolainen, the Head of the Department, for giving me my first academic job at the department a couple of years ago. My time at the department has already taught me a lot.

Thank you to the examiner of this thesis, Professor Mikko Kaasalainen, for giving me the opportunity to work in this project, and to write this thesis. I am also thankful for the lack of strict deadlines and meetings. These little things made the thesis writing a great deal less stressful.

D.Sc. Pasi Raumonen was the second examiner and my advisor during the writing process. I am grateful for all the help you have provided me and for the many interesting discussion we have had on and off topic. Hopefully our cooperation continues for many more years.

A special thanks to the people at `#triviaali_pökäle` for making the time at the university truly unforgettable and full of surprises.

I would like to thank my best friend Carlos for turning the thesis writing into a friendly race to the finish line. I did lose. But there really is no shame in losing to a better man. And I did win the previous race and I am planning to win the next one also.

Haluan kiittää vanhempiani, jotka varmasti vähättelevät osaansa tässäkin. Totuus kuitenkin on, että en olisi kirjoittanut tätä työtä ilman teitä ja teidän tukeanne. Toivon, että osaatte olla ylpeitä kovan työnne tuloksista.

Haluan huomioida myös siskoni, joka urhoollisesti tarjoutui tekemään vähemmän työtä, jotta minäkin saisin osani. Fyysinen raataminen raikkaassa maatilaympäristössä oli juuri, mitä tarvitsin akateemisen pohdiskelun vastapainoksi.

And lastly my love, Fanni. You were the one who got me to stop writing every once in a while. Thank you for standing by me and for dragging me away from the computer to remind me of what is truly important. ♡

Tampere April 23, 2012

Markku Åkerblom

Contents

Abbreviations and notation

Notation

a	Scalar number a.
$\mathbf{a}$	A vector.
M	A matrix.
$\mathbb{N}$	Set of natural numbers.
$\mathbb{Z}$	Set of integers.
$\mathbb{R}$	Set of real numbers.
τ	Topology.
$\mathcal{N}$	Normal space.
$\mathcal{T}$	Tangent space.
$\{a_1, a_2, \dots\}$	A set of elements.
$a \in A$	a is an element of set A.
$A \subset B$	Set A is a subset of set B.
$A \cup B$	Union of sets.
$A \cap B$	Intersection of sets.
$\bigcup_i A_i$	Union of multiple sets.
$\bigcap_i A_i$	Intersection of multiple sets.
$A \setminus B$	Difference of sets.
$\emptyset$	Empty set.
$\operatorname{ext}(A)$	Extended set of set A. (See page 33.)
$\operatorname{ext}^n(A)$	Extended set of degree n of set A.
M^{T}	Transpose of matrix M.
$\|\mathbf{a}\|$	Euclidean norm of vector $\mathbf{a}$.
$\|a\|$	Absolute value of the scalar a.
(M, d)	Metric space. Function d defines a distance in set M.
(X, τ)	Topological space. The collection τ is a topology of set X.
$\mathbb{E}(M)$	Expected value of matrix M.
$\operatorname{cov}(M)$	Covariance matrix of matrix M.
$\lceil a \rceil$	Ceil of real number a.
$\lfloor a \rfloor$	Floor of real number a.
$\sphericalangle(\mathbf{a}, \mathbf{b})$	Angle between the vectors $\mathbf{a}$ and $\mathbf{b}$.
$\mathbf{a} \perp \mathbf{b}$	The vectors $\mathbf{a}$ and $\mathbf{b}$ are perpendicular.
$\mathbf{a} \parallel \mathbf{b}$	The vectors $\mathbf{a}$ and $\mathbf{b}$ are parallel.
$\mathbf{a} \cdot \mathbf{b}$	Dot product of the vectors.
$\mathbf{a} \times \mathbf{b}$	Cross product of the vectors.
$\forall$	Universal quantification.
$\exists$	Existential quantification.
$p \wedge q$	Both of the statements p and q are true.
$p \vee q$	Either p or q or both of them are true.
f	A function.
$f : A \to B$	A mapping f from set A to set B.

$f(x)$	A function of the variable x.
$f'(x)$	First derivative of the function f.
$\sum_{i=1}^{n} a_i$	Sum of the elements $a_1, a_2, \ldots, a_n$.
$=$	Equal to.
$\neq$	Not equal to.
$\approx$	Approximately equal to.
$\propto$	Proportional to.
$a \leftarrow b$	The value of the variable a is updated to the value b.
$\Rightarrow$	Implication.
$\Leftrightarrow$	Equivalence.

Abbreviations

LIDAR	*Light Detection And Ranging*. Optical sensing technology to measure distances.
TLS	Terrestrial laser scanning.
LS	Least-squares.
PCA	Principal component analysis.
PC	Principal component.

1 Introduction

Forests are and have been for a long time the biggest natural resource in Finland. Today trees are utilized by various industries, and the growing economical and ecological pressure have created an urgent demand for forest and tree measurements. It has become crucial to receive more and more accurate measurements from trees and complete forests in order to make more detailed plans for forest preserving and consumption [26].

From the economical point of view, forests and forest industry are optimization problems. How to grow and use forests and trees to maximize the profit? How to minimize the excess material when consuming trees? Where to build tree processing facilities to optimize the processing chain? All these questions can be answered to a certain extend by understanding how trees and forests are constructed and where they are located.

Lately, also the ecological values have become a factor in forest development. People have become more aware of the carbon dioxide footprint of chemical decomposition that happens when logging excess is left into the forest. This way of dealing with logging excess was the industry trend until a few years ago, but since then new ways to utilize the excess have been developed.

One alternative is to burn the excess material in a bioenergy plant to produce energy. This way logging excess material can be seen as a renewable energy source. But one question remains to be answered; *which is more environmental friendly, to burn the excess material or to let it decompose naturally in the forest?* To answer this question one would need to know the *branch size distribution* of the tree in question, because the rate of decomposition is relative to the physical size of the material. Only by knowing the distribution can one make valid estimations about the decomposition time and the amount of the reaction products like carbon dioxide.

The amount of energy produced by a bioenergy plant is proportional to the biomass of the material burned in the reactor. To make accurate estimates about the energy production one needs to know the amount of bioenergy stored in the excess logging material. This can be found by weighting the material after the tree has been felled. Naturally it would be more efficient if there was no need to cut down the tree. This would be possible if there existed a way to measure the amount of biomass in a living tree.

1.1 Laser scanning

Even though the demand for accuracy is high, forest measurements are still at best fairly inaccurate. For example the only measurement done for a tree that is to be felled and sent to a sawmill, is the diameter measurement at the height of the chest (approximately 1.3 meters) of the person doing the measurement. In some cases only this single value is used to approximate the height of the tree. To improve the accuracy it is possible to use an ultrasound transponder. [8]

The measurement methods above only give information about the trunk of the

tree. They give no information about the location, size, or type of the branches and leaves or needles. One method that does gather such information is *laser scanning*.

There are two main types of laser scanning in the context of forest measurement. In both types of measurement the basic idea is the same. The scanner emits laser impulses that then reflect from the object that is being measured and the scanner computes the distance from the scanner to the object based on the time it took for the impulse to return. Making several such measurements one acquires a discrete sample of the surface of the object seen from the direction of the scanner.

Airborne laser scanning is carried out from a plane. The object in this case is the forest below the plane. The laser pulses reflect either from the trees, ground or somewhere in between. Each measurement is used to compute the height at which it was reflected when the altitude of the plane is known. The laser scanning results are then combined with satellite positioning data and possibly some satellite images to make an estimate of the number of trees and tree types in the measured area [15].

Airborne laser scanning is a great way to measure large areas of forest fast. However, the method has a high error level since every tree usually gives only one or two measurements. These measurements are then used to determine the height and type of the tree.

The other type of laser scanning is *terrestrial* laser scanning (TLS). This type of scanning is done on the ground, usually one tree at a time. It is also possible to scan multiple trees at a time when the scanner is moving. The scanner takes hundreds of thousands of accurate measurements from the surface of the tree. Furthermore, to maximize the coverage of the measurements, each tree is scanned from multiple, usually three, directions. The measurements from different directions are later combined to form a single point cloud presenting the surface of the tree [4, 2, 20]. Terrestrial laser scanning gives very accurate measurements, but the scanning can be fairly slow although nowadays the scanners are capable of capturing millions of points in under five minutes [3]. The measurement time is of course dependent on the equipment, sample density, and the size of the tree.

1.2 Measurement analysis

Even though laser scanning promises high accuracy, alone it is not enough. The measurements require a lot of post-processing and analysis to be useful. Many analysis methods for specific tasks are available (e.g. [1, 25, 22]), but a complete method for finding the exact location and size of all the branches and the topology of the tree is yet to be presented.

In [17] a method for the reconstruction of the stem and the largest branches is presented. The method is based on a voxel space approach and mathematical morphology [6]. In the voxel space approach, the space containing the point cloud is divided into cubical voxels similarly to what is shown in section 4.1. Then, for each voxel, it is determined whether the voxel is inside or outside the tree surface.

The usage of the voxel approach can cause inaccuracy in the reconstruction, and the selection of the size of the voxels is no simple task. By choosing too small voxels one increases the probability of gaps in the reconstruction. On the other hand, too large voxels decrease the level of the detail of the reconstruction.

1.3 Algorithm for a quantitative tree model

This thesis presents a method for reconstructing a comprehensive model of a single tree from terrestrial laser scanning data. The method was first introduced in [19]. Since then the method has been improved[18] and validated. In this thesis, I present methods, that our research team has developed to improve the quality of the reconstruction by filling gaps in the produced model as a post-processing step (section 5.5). I derive error estimates for certain quantities used in the reconstruction process. Finally, I validate the presented method by using generated tree models in section 6.1. The original tree model is compared with the reconstructed model, and limits for the maximum tolerated measurement error, and the minimum required sample density, are estimated by changing their respective values one at a time.

The development of tree modeling method has been developed in interdisciplinary collaboration with the *Finnish Geodetic Institute* and the *Finnish Environment Institute* (SYKE). The former has provided the laser scanning data used in this thesis.

The outline of the tree modeling method is the following:

The set of measurements are assumed to be a comprehensive presentation of the surface of the tree. The output of the method is a quantitative cylinder model which consists of numerous cylinders that present the trunk and branches of the reconstructed tree in a local scale. The transformation from the point cloud to the cylinder model is accomplished in a way that captures the topological relations between the branches. This means that starting from any cylinder in the model, one can find the respective parent, extension and child cylinders and advance over the whole model in a natural way. Quantitative tree characteristics such as the branch size distribution and the total or partial volumes of a tree can be quickly approximated from a completed cylinder model. Other information, such as the bifurcation angles and frequencies are also easily accessible.

More precisely the method for analysing the TLS point cloud is based on creating a cover of the point cloud. The unknown tree surface is covered with small partially overlapping sets conforming to the details of the surface. These sets are used to approximate the local properties, such as the size, shape and orientation of the surface. Because the sets are intersecting, an extension of any given combination of the sets by their neighboring sets is easily determined. Such an extension provides a vital tool for the method since it allows the change in scale, i.e., from small details to larger ones.

The cover sets are classified using their geometric properties, such as their dimensionality and direction, to find the trunk of the tree and its base. Next, the measurements from the surrounding ground, the undergrowth and possible other trees are filtered out using the neighbor relation of the cover sets. The remaining cover sets are divided into *components* based on their connectedness, and possible components not part of the tree are removed using heuristics. Ideally,

each set of measurements contains only one component but gaps in the data often lead to multiple components.

Each component is divided into segments corresponding to parts of the tree that can be called branches: connected parts with no bifurcations. The segment formation process widely uses the neighbor relation of the cover sets, and the relational structure information is recorded during the process.

Since the branches can be curved and have a non-constant radius, the segment is divided into even smaller parts, called regions, which are then approximated as cylinders. A cylinder is fitted to the points forming a region as a least-squares minimization problem. Geometric properties of the corresponding measurement subset are used to find initial values for the fitting process. A priori information, e.g., assuming the branch radius to diminish with the distance from the trunk, is also used to find the best approximation. When cylinders have been fitted to all the segments, the cylinder model can be further refined by locating and filling gaps between cylinders.

Geometric properties of the tree, such as the total volume or, for example, the volume of the trunk and the branch size distribution, are easily computable from the cylinder model. In addition, the model contains the exact location of each cylinder and the relations between cylinders. Therefore result visualization is fast and information on, e.g., bifurcation frequency and angle is available.

The method is automatic in the sense that it only requires a few parameters, such as the radius of the cover sets, and the measurements as inputs. User interaction is not required during the analysis. Since the method is based on topological concepts, such as connectivity, it is scale-independent. Thus the same realization of the method can be used to analyse very differently sized tree-like point clouds, and the accuracy of the method is mainly restricted by the accuracy of the measurements.

Because only the cover sets and their centers are needed for most of the steps, only tens or hundreds of thousands of points are mostly used instead of all the millions of measured points. Thus the memory requirements and the computational time are greatly reduced. The method has been implemented in MATLAB® and on average hardware the analysis takes somewhere from a few minutes up to ten minutes.

2

In this chapter, the mathematical concepts on which the tree model algorithm is based on are defined. The measurements are divided into sets which are then manipulated in different ways, and therefore the chapter starts from the basic set theory in section 2.1. The chapter also shows that the set of measurements is a metric space and a topological space. The former ensures that distances can be measured and the latter that neighbors and connectedness are defined. A formal definition for a surface is given in section 2.4, and the mathematical surface reconstruction problem is discussed in section 2.5.2. Finally, a method for approximating the surface properties and measurement set geometric properties is presented in section 2.5.3.

2.1 Set theory

Let A and B be sets. The sets are said to be equal if

$$\forall x \in A \mid x \in B \quad \wedge \quad \forall y \in B \mid y \in A, \tag{2.1}$$

which means that the two sets have the same elements. If two sets are not equal they are said to be *non-equal*.

Given two sets A and B the *union* of the sets is a set

$$A \cup B = \{x \in A \quad \vee \quad x \in B\}. \tag{2.2}$$

The *intersection* of the sets is a set

$$A \cap B = \{x \in A \quad \wedge \quad x \in B\}. \tag{2.3}$$

The *difference of the sets* is a set

$$A \setminus B = \{x \in A \quad \wedge \quad x \notin B\}. \tag{2.4}$$

A set A is a *subset* of a set B if

$$\forall x \in A \mid x \in B. \tag{2.5}$$

If A is a subset of B, it is denoted as $A \subset B$. In this case B is said to be a *superset* of the set A.

A set of subsets $P = \{A_1, A_2, \ldots, A_n\}$ of a set A is called a *cover* of set A if

$$A_1 \cup A_2 \cup \cdots \cup A_n = A. \tag{2.6}$$

That is to say that if the union of the subsets is the original set A then the set of subsets is a cover of the set A.

In the special case where the sets of a cover are pairwise disjoint the cover is called a partition. The cover $P = \{A_1, A_2, \ldots, A_n\}$ of a set A is a *partition* of set A if

$$\forall i \neq j \quad A_i \cap A_j = \emptyset. \tag{2.7}$$

2.2 Topology

A *topology* on a set X is a collection τ of subsets. The collection τ must include at least the empty set and the whole set X. Furthermore, for every two subsets in the topology, their intersection and union must be included in the topology also. Mathematically the following conditions must hold true for all the subsets A and B in the topology τ:

$$\emptyset, X \in \tau \tag{2.8a}$$
$$A, B \in \tau \quad \Rightarrow \quad A \cup B \in \tau \tag{2.8b}$$
$$A, B \in \tau \quad \Rightarrow \quad A \cap B \in \tau \tag{2.8c}$$

The pair (X, τ) is called a *topological space*. The sets in τ are called *open sets* [28]. By defining a topology on a set, certain properties, such as *connectedness*, can be defined for the set.

2.2.1 Connectedness

A topological space (X, τ) is said to be *connected* if it consists of a single piece. This is equivalent with the condition that the set X cannot be presented as an union of two disjoint elements of the collection τ [24].

The simplest mathematical formulation for connectedness is the following.

$$\forall A \in \tau, A \neq X \quad \exists B \in \tau, B \neq A \quad : \quad A \cap B \neq \emptyset \tag{2.9}$$

If a topological space is not connected then it is said to be *disconnected*. Furthermore, *connectedness* is a property of a topological space that states whether the space is connected or disconnected.

2.3 Metric spaces

A function $d : M \times M \to \mathbb{R}$ is a *metric* if it satisfies the following conditions for all elements $x, y, z \in M$.

$$d(x, y) \geq 0 \tag{2.10a}$$
$$d(x, y) = 0 \Leftrightarrow x = y \tag{2.10b}$$
$$d(x, y) = d(y, x) \tag{2.10c}$$
$$d(x, z) \leq d(x, y) + d(y, z) \tag{2.10d}$$

The pair (M, d) is called a *metric space* and the value $d(x, y)$ is called the *distance* between the elements x and y. [23]

For example, the real coordinate space $\mathbb{R}^n$ with the usual distance function

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}, \tag{2.11}$$

is a metric space. When the real coordinate space is viewed as a *vector space*[13, p. 217] and combined with the above distance, it is called the *Euclidean space*. Later on the text $\mathbb{R}^n$ is used to note the Euclidean space if not stated otherwise.

2.3.1 neighborhoods

Let (M, d) be a metric space, $\mathbf{a} \in X$ and $r > 0$. The subset of M consisting of points $\mathbf{p} \in M$ for which

$$d(\mathbf{a}, \mathbf{p}) < r, \tag{2.12}$$

is called an *open ball about* $\mathbf{a}$ *of radius* r [16]. In this thesis an open ball is called an r-ball and it is denoted by

$$B(\mathbf{a}, r) = \{\, \mathbf{p} \in M \mid d(\mathbf{a}, \mathbf{p}) < r \,\}. \tag{2.13}$$

Let (M, d) be a metric space and $\mathbf{a} \in M$. A subset A of M is called a *neighborhood* of $\mathbf{a}$ if

$$\exists r > 0 \mid B(a, r) \subset A. \tag{2.14}$$

By this definition $B(\mathbf{a}, r)$ is a neighborhood of $\mathbf{a}$ for all $r > 0$ [16].

The r-balls defined in equation (2.13) form a topology on the set M since the collection of r-balls with the set M satisfy the conditions (2.8). This specific topology can be called the *metric topology* and it can be denoted τ_d to emphasize the connection to the metric d [28].

2.4 Manifolds and surfaces

A graph of a function $f : A \to B$ is a set of the ordered pairs $(\mathbf{a}, \mathbf{b}) \in A \times B$ for which

$$f(\mathbf{a}) = \mathbf{b} \tag{2.15}$$

holds. If a function maps from the set $\mathbb{R}^m$ to the set $\mathbb{R}^n$ then the points in the graph are elements of the Cartesian product $\mathbb{R}^m \times \mathbb{R}^n = \mathbb{R}^{m+n}$.

The subset M of the Euclidean space $\mathbb{R}^n$ is a k-dimensional *manifold* if it is locally a graph of some smooth function that expresses $n-k$ variables as a function of the other k variables [9]. This means that for every point $\mathbf{p} \in M$ there exists a subset $A_\mathbf{p}$ of M and a smooth function $f_\mathbf{p} : B \subset \mathbb{R}^k \to \mathbb{R}^{n-k}$ so that the points in $A_\mathbf{p}$ are a graph of $f_\mathbf{p}$ [21]. A 2-dimensional manifold is called a *surface* [11]. In the space $\mathbb{R}^3$ a surface is locally a graph of some function $f : \mathbb{R}^2 \to \mathbb{R}$ where, for example, $z = f(x, y)$.

Let the set M be a k-dimensional manifold in $\mathbb{R}^n$ and $\mathbf{p}_0$ a point on the manifold. Let $A_{\mathbf{p}_0}$ be a subset of M so that $\mathbf{p}_0 \in A_{\mathbf{p}_0}$ and a function $f_{\mathbf{p}_0}$ whose graph the manifold is in $A_{\mathbf{p}_0}$. The graph of the function $g : \mathbb{R}^k \to \mathbb{R}^{n-k}$

$$g(\mathbf{p}) = f_{\mathbf{p}_0}(\mathbf{p}_0) + (\mathbf{p} - \mathbf{p}_0) f'_{\mathbf{p}_0}(\mathbf{p}_0)^{\mathrm{T}} \tag{2.16}$$

is a part of a k-dimensional hyperplane and tangent to the manifold at the point $\mathbf{p}_0$. The *tangent space* $\mathcal{T}_{\mathbf{p}_0}(M)$ of the manifold M in the point $\mathbf{p}_0$ is the vector space whose elements have a starting point at $\mathbf{p}_0$ and an end point on the graph of the function $g(\mathbf{p})$ [21]. A two-dimensional tangent space is called a *tangent plane*.

The *normal space* $\mathcal{N}_{\mathbf{p}_0}(M)$ of the manifold M in the point $\mathbf{p}_0$ is the vector space whose elements start from the point $\mathbf{p}_0$ and that are perpendicular to all the

elements in the tangent space $\mathcal{T}_{\mathbf{p}_0}(M)$. The elements of the normal space are called *normal vectors*. [21] In the case of surfaces embedded in $\mathbb{R}^3$ there is only one direction for each point in the surface that is perpendicular to the corresponding tangent plane.

Let S be a surface in $\mathbb{R}^n$ and $\mathbf{a}, \mathbf{b}$ points on the surface S. Then the distance in the surface S between the points $\mathbf{a}$ and $\mathbf{b}$ has to be measured along the surface. Let d_S be the metric that defines this distance for all the points in the surface S. It can be shown that

$$d_S(\mathbf{a}, \mathbf{b}) \geq d(\mathbf{a}, \mathbf{b}) \quad \forall \mathbf{a}, \mathbf{b} \in S. \tag{2.17}$$

Figure 2.1 illustrates this connection. The inequality (2.17) states that the distance measured on the surface is always larger or equal to the Euclidean metric d. However, in a local scale the difference between the distances $|d_S - d|$ is small enough so d gives a good estimate of the distance on the surface.

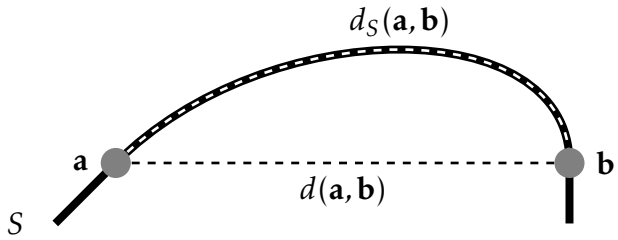


Figure 2.1: Surface metric approximation. The solid black line describes the surface S. The white dashed line gives the distance d_S along the surface and the black dashed line gives the Euclidean distance d between the points $\mathbf{a}$ and $\mathbf{b}$ of the surface. The Euclidean distance is always shorter or the same as the distance along the surface.

2.5 From point cloud to surface

This section describes how the measurements are viewed mathematically and how the tree surface can be reconstructed. In section 2.5.2 the nature of inverse problems is also discussed. In section 2.5.3 a method called *principal component analysis* is presented. The method allows the local approximation of the surface to be constructed.

2.5.1 Point cloud

Let $\mathbf{x}_i$ be a point in $\mathbb{R}^n$ for all indices $i = 1 \ldots k$. The set $P = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$ is a *point cloud*. This simply means that a point cloud is a finite collection of points. In this thesis, most of the points clouds are subsets of the three-dimensional Euclidean space $\mathbb{R}^3$.

Because a point cloud P is a subset of the Euclidean space $\mathbb{R}^n$, it is also a metric space with the metric d_P defined as the restriction of the Euclidean metric (2.11)

The r-ball (2.13) is also well defined in a point cloud, and therefore a point cloud is also a topological space with the metric topology τ_{d_P}.

Point clouds can be presented as matrices, which makes certain operations simpler. If a point cloud P has m elements $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m \in \mathbb{R}^n$, then the point cloud can be presented as an $m \times n$ matrix X with the row vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$. The order of the row vectors does not need to match the indexing of the point cloud, and in most cases the order is irrelevant. The matrix notation simplifies operations that are applied to a point cloud.

2.5.2 Surface reconstruction

In this thesis, collections of measurements are considered to be point clouds. Furthermore, the measured points are seen as samples of the surface of an object, namely a tree. The goal is to reconstruct that surface from the measurements. This reconstruction problem is an *inverse problem*.

An inverse problem is *ill-posed* by definition, that is an inverse problem must violate at least one of the Hadamard conditions [12, p. 9]. These conditions are the *existence*, the *uniqueness* and the *stability* of the solution. If a problem satisfies the Hadamard conditions it is *well-posed* and it is called a direct problem.

For example, given a known system and input, the direct problem can be to find the output of the system. The inverse problem is the inverse of this: finding the input that would cause the known output [7, p. 2]. The surface reconstruction problem presented next falls into this definition.

The inverse problem in this thesis is the surface reconstruction given a point cloud P of surface samples. The goal in this case is to find a parametrized surface S that optimally fits the measurements in some sense. By doing so one finds the optimal parameters for the surface. One possibility is to minimize the distance between the measurements and the surface in a *least-squares* sense. The described reconstruction problem violates the first two Hadamard conditions: existence and uniqueness.

In a least-squares optimization problem the objective function to be minimized is the sum of all the n squared distances between the samples and the surface [29]:

$$f(S) = \sum_{i=1}^{n} w_i (d(S, \mathbf{p}_i))^2, \tag{2.18}$$

where w_i is the possible weighting coefficient and $d(S, \mathbf{p})$ the shortest distance between the surface and the point $\mathbf{p}$. A few surface fitting methods are discussed in more detail in section 5.1.

2.5.3 Principal component analysis

Often *principal component analysis* is used to reduce the dimensionality of a data set [10], but, as in this thesis, it can also be used to analyse the variance of a data set. The directions in which the data vary the most can be found through the analysis. This means that the shape and direction of the data can be estimated in some space.

Let us define the concepts of point cloud covariance and principal component analysis.

Point cloud covariance Let the elements of a vector $\mathbf{x} \in \mathbb{R}^m$ samples of a variable. The *expected value*, or the *mean*, of a the vector $\mathbf{x}$ is

$$\mathbb{E}(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i, \tag{2.19}$$

where $\mathbf{x}_i$ is the i^{TH} element of the vector $\mathbf{x}$. Furthermore, if the columns of the matrix $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{m \times n}$ are considered as samples of n random variables, $\mathbb{E}(X) \in \mathbb{R}^n$ is a row vector with elements $\mathbb{E}(\mathbf{x}_1), \mathbb{E}(\mathbf{x}_2), \ldots, \mathbb{E}(\mathbf{x}_n)$ [5, p. 81].

The *sample covariance matrix* of the matrix X is the matrix $\operatorname{cov}(X)$ whose elements have the following form:

$$\operatorname{cov}(X)_{ij} = \frac{1}{m} \sum_{k=1}^{m} (x_{ki} - \mathbb{E}(x_i))(x_{kj} - \mathbb{E}\left(x_j\right)), \tag{2.20}$$

where $\operatorname{cov}(X)_{ij}$ is the element on the i^{TH} row and j^{TH} column and x_{ij} is an element of the matrix X. The element measures *covariance* between the variables defined by the columns $\mathbf{x}_i$ and $\mathbf{x}_j$ of the matrix X; that is, the strength of the their linear relationship [27]. The dimension of the sample covariance matrix is the same as the sample matrix X.

Principal components Let a data set have N variables that can be correlated. Principal component analysis produces *principal components* that are linear combinations of the original variables. Principal components are uncorrelated and usually ordered such that the first principal component explains most of the variance in the original variables, the second PC the second most variance in the uncorrelated space, and so on. The number of principal components is always lower or equal to N.

Let $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ be a vector of N random variables with a known sample covariance matrix Σ. The k^{TH} largest principal component z_k is a linear combination of the variables $x_1, x_2, \ldots, x_N$.

$$z_k = \mathbf{a}_k^{\mathrm{T}} \mathbf{x}, \tag{2.21}$$

where $\mathbf{a}_k$ is the eigenvector corresponding to the k^{TH} largest eigenvalue of Σ. When analysing the geometry of a data set, the principal components in themselves are not interesting. The eigenvalues and the eigenvectors of the matrix Σ are sufficient.

Because the covariance matrix is symmetric its eigenvectors are orthogonal:

$$\mathbf{a}_1 \perp \mathbf{a}_2 \perp \cdots \perp \mathbf{a}_N. \tag{2.22}$$

This ensures that the principal components are uncorrelated. Geometrically, $\mathbf{a}_1$ can be interpreted as the direction in which the variance in the variables is largest, $\mathbf{a}_2$ as the second largest, and so on.

The *total variance* [13] of the data is

$$\lambda_{\text{total}} = \sum_{i=1}^{N} \lambda_i, \tag{2.23}$$

where λ_i are the eigenvalues of the covariance matrix. The ratio $\frac{\lambda_k}{\lambda_{\text{total}}}$ describes how much of the variance the k^{TH} principal component explains. The ratio $\frac{\lambda_k}{\lambda_p}$, $k > p$, of the eigenvalues corresponding to the k^{TH} and p^{TH} principal components, describes how elongated the set would be when projected on a plane defined by the k^{TH} and p^{TH} eigenvectors. Because of this geometrical interpretation, the eigenvalues and eigenvectors of the covariance matrix can be used to characterize point clouds.

Let the point cloud $P \subset \mathbb{R}^n$ be a sample of a manifold $M \subset \mathbb{R}^k$. The principal components of the set $B(\mathbf{p}, r)$ for some point $\mathbf{p} \in P$ and any $r > 0$ give an approximation of the shape of the manifold in that neighborhood, as long as $B(\mathbf{p}, r)$ corresponds to a small, local part of M: The span of the k largest eigenvectors approximates the tangent space of the manifold in that point, and the $n-k$ smallest principal components span the approximated normal space. In the case $n = 3$ and $k = 2$, the two largest eigenvectors span the approximated tangent plane, and the smallest one is an approximation of the surface normal in the point $\mathbf{p}$.

An example of principal component analysis can be seen in figure 2.2. The figure shows two examples of two-dimensional point clouds and the eigenvectors corresponding to their principal components. The eigenvectors have been normalized and scaled with the corresponding eigenvalues. Because the point cloud in the first set has the same amount of variance in all directions, the vectors are similar in length, and the variation explanation ratios of both components are close to 50 percent. The ratio λ_1/λ_2 is close to unity, so the point cloud is not elongated. In the second example, the data are elongated, and vary the most in roughly the direction of the vector $(1, 1)$. This causes the length difference of the scaled eigenvectors. The ratio λ_1/λ_2 is now larger than one, which means that the point cloud is elongated.

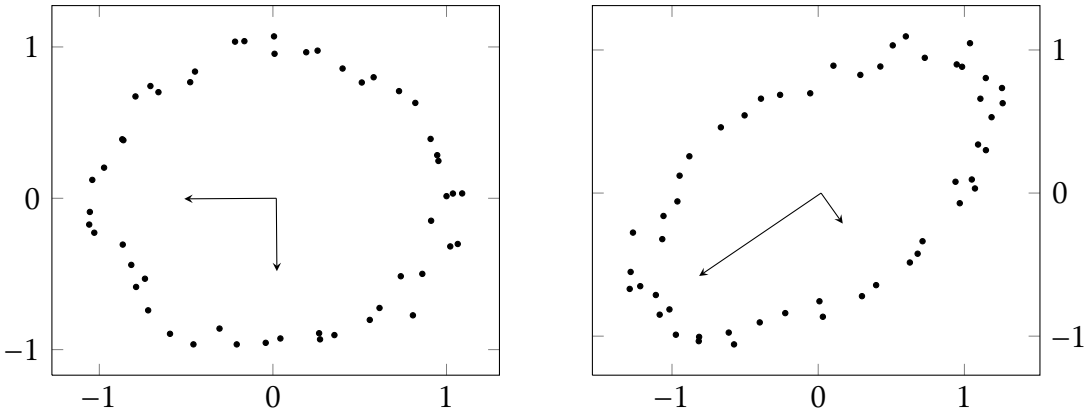
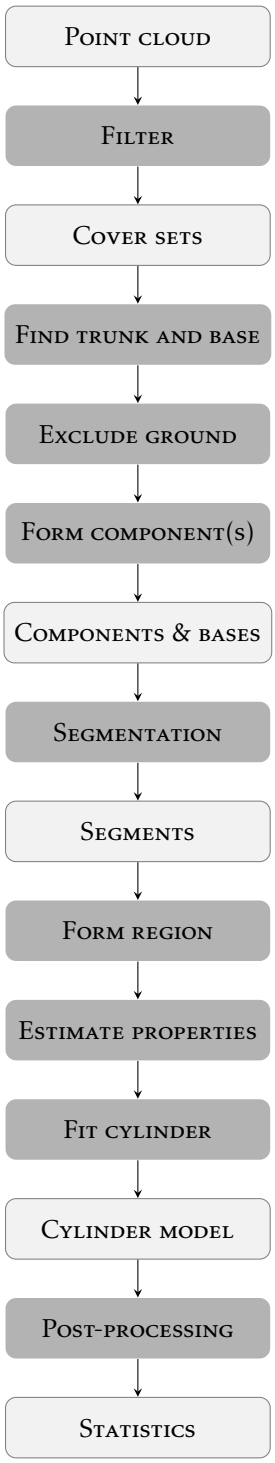


Figure 2.2: An example of a principle component analysis. Both data sets have 50 points. The vectors are the eigenvectors corresponding to the principal components scaled with the eigenvalues. Since the first set does not vary more in any direction than any other, the vectors have similar lengths. The second set has the greatest variance in the direction of the longer vector.

3 Tree model algorithm

This chapter introduces a tree model algorithm that has been developed to approximate the surface and volume of a single tree from a set of point cloud data [19, 18]. Since the algorithm is very complex, it is not presented completely in this document, but an overview of the algorithm's work flow and basic principles on which it is build on are presented step-by-step in the sections of this chapter. The result of the algorithm is a quantitative cylinder model that contains the location and size of each branch as well as the relations between branches.

The flowchart on this page shows a visualization of the steps of the algorithm. The white boxes stand for new data, or information that is produced by the previous sub-processes and that is utilized by the subsequent steps. It should be noted that all the information, not just the previously computed, is usually used by the upcoming sub-processes. The sub-processes are denoted nodes with a darker colour. These processes are taken in the order shown, and each utilizes the outputs of the previous step so they cannot be completed parallel to each other.



Since the algorithm is strongly based on certain mathematical and computational methods, an additional table 3.1 is used to show which steps depend on which methods. The methods are also mentioned in the text of this section and explained in further detail in chapter 4. These methods should be viewed as a set of tools rather than stages of the algorithm.

3.1 Input data

The algorithm operates on a point cloud of three-dimensional measurements, which are usually recorded with a LIDAR-scanner. There are typically hundreds of thousands to millions[1] of points from a single tree. Each point is considered to be a sample of the surface of the tree. These samples have errors, or a level of inaccuracy, due to many reasons such as the level of detail of the equipment and the fact that parts of the tree can be moving due to for example wind conditions. The data set can also contain measurements of the ground and undergrowth.

[1] The measurement sets used in the examples of this thesis contained one to four million points each.

Table 3.1: Connections between the algorithm steps and the mathematical and computational methods.

	Partitioning	Cylinder fitting	Set extension	Direction estimation	Characterization
Forming cover sets	X				X
Locating trunk			X	X	X
Excluding ground			X		
Forming components			X		
Forming segments		X	X	X	
Forming regions		X	X	X	
Approximating surface		X			
Filling gaps	X				

A tree is usually scanned from more than one direction to ensure that all the parts are visible in the scan. However, because of the complex geometry of the tree and the limitations of the scanning equipment[2], there are always imperfections in the data, which means that some parts of the tree get more measurements than the others and some get very little or none at all.

Before the measurements can be analysed using the algorithm presented here the measurements from different scanning directions have to be combined into a single Cartesian coordinate system. This thesis omits the preprocessing required for the measurements and assumes that the input data is in the correct format.

3.1.1 Filtering

The measurement data can contain points that should not be used in the modeling process. Such points are for example *phantom* measurements that are reflected from mere air between the branches and isolated measurements that are not connected to the rest of the data set. The unnecessary measurements are removed by filtering the point cloud.

Filtering can be done with the help of an r-cube partition or cover sets. In both cases the basic idea is the same. Form small sets using a random point as center points and remove sets that are not part of a large enough component.

3.2 Creating the cover sets

The algorithm begins by creating a cover for the point cloud. The elements of the cover are sets that are defined by a center point, that is an element of the point cloud, and a radius r, which can be constant for all the cover sets or it may vary. A cover set is defined as a spherical environment of the center point with the

[2]The measurements are made on ground level. Furthermore, some branches can be blocked from the scanner by other branches.

radius r. It should be noted that the cover sets do not need to be disjoint. On the contrary they should be intersecting if possible. Details of the cover set creation and their properties are discussed in section 4.2. The cover set creation process utilizes an r-cube partition described in section 4.1.

Once the cover sets have been defined, their center points are used as an approximation of the whole point cloud. This ensures faster computational times with only a small drop in accuracy. The cover sets are also assigned an identifying number, or an index.

At the end of the cover creation step, each cover set has a distinctive number, and by using the index numbers of the cover sets the following two-way connections can be defined:

- given a point, one knows which cover set(s) it is in,
- given a cover set, one can identify the points in it.

Given the previous point–cover set connections one can define the *neighboring sets* for each cover set. In short, given a cover set A, the cover set B is a neighboring set of the set A if

$$A \cap B \neq \emptyset \tag{3.1}$$

holds, which means that the cover sets A and B have common elements. It is easy to see that the relation is symmetric. The neighboring set relation is the basis for the set extension (see section 4.2.1 for details). By taking the union of a cover set and all of its neighboring sets, the extended cover set of the first degree is formed. This process can be continued by taking the union of the first degree extended cover set and all the neighboring sets of its member cover sets, and thus creating the second degree extended cover set. The extended set of a set A is noted as $\mathrm{ext}\,(A)$ and the n^{TH} degree extension as $\mathrm{ext}^n(A)$. If the cover sets conform with the surface, the set extension moves along the surface.

Once the cover sets and their first and second degree extensions have been formed, *characterization* can begin (see details in section 4.3). In this step, multiple geometric characteristics are computed for each cover set. Some of the characteristic computations use just the original cover sets, but others can use the extended cover sets to achieve better accuracy.

3.3 Characterizing the cover sets

The geometric characterization process is based on approximating the structure tensor with the principal component analysis described in section 2.5.3. The structure tensor is a scale independent way of describing the geometry of the cover set.

During the characterization process, estimates for the direction, size and dimensionality are computed for each cover set. The characteristics are used in the later steps of the algorithm to *classify* the cover sets into categories. It should be noted that after this step the analysis is executed on the cover sets instead of individual points, unless specified otherwise. The classification process and principles are described in detail is section 4.4.

There are three main types of cover sets, each of which are usually present in an arbitrary set of tree measurement data; sets can be part of the ground, trunk or branches. The first of these is irrelevant for the analysis and it should be excluded as early in the process as possible. The algorithm does this by first locating the trunk and by using the trunk to identify the ground points.

3.4 Finding the trunk

The trunk cover sets are fairly planar since the radius of the trunk is expected to be larger than the radius of the branches. The direction of the sets should also be close to parallel to the expected trunk direction, which is usually the direction of the positive z-axis. This simple *initial filtering* does not find all the trunk cover sets and it might get some false positives. Therefore it is important to improve the trunk classification. The trunk set is next extended by including the neighboring sets of its member sets. This is done so that the trunk set will have a clearer form and not many gaps.

The trunk classification continues by inspecting which of the cover sets in the initial trunk set are connected to each other. The trunk set is assumed to be the union of the largest[3] sets of connected cover sets initially classified as trunk. Additionally, it is possible to enlarge the trunk set with smaller collections of connected sets initially classified as trunk, given that they are relatively close to the axis defined by the largest set. It should also be noted that the trunk set is formed before the components (see section 3.6), and therefore the trunk set may be disconnected.

When the trunk set has taken its final form, the next step is to find the *base* of the trunk, and the tree. The base is defined to be a thin layer of points in the lowest part of the trunk. The estimated trunk direction is used to find the base set.

3.5 Excluding the ground

Since the base of the tree is known it is fair to assume that everything lower than the base is part of the ground. Therefore the *initial ground set* is defined to be the part of the first degree extension of the base that is not part of the trunk. Mathematically this means that

$$G_{\text{init}} = \text{ext}\,(B) \setminus T, \tag{3.2}$$

where G_{init} is the initial ground set, B is the base set and T is the trunk set.

After this initial classification, if the initial ground set G_{init} is non-empty, the ground set is extended with its neighboring cover sets that are not part of the trunk. This is continued as long as cover sets are still added to the ground set. The consecutive ground set extensions G_k and G_{k+1} satisfy the following conditions

[3] Largest in the sense of the number of cover sets rather than individual points.

for all indices $k = 1, 2, \ldots$

$$G_1 = G_{\text{init}} \tag{3.3a}$$
$$G_{k+1} = \text{ext}\,(G_k) \setminus T \tag{3.3b}$$
$$G_{k+1} \neq G_k. \tag{3.3c}$$

It should be noted that the ground set is not going to include just the measurements presenting parts of the actual ground, but also the measurements reflected from the undergrowth and the possible other trees and foreign objects. This kind of an approach on the ground formation gives the desired result as long as the ground is not connected to the measured tree along any other path than the trunk.

All the cover sets in the ground set can be excluded from further analysis. However, additional measurements that are not from the object tree can still remain. These points are just not connected to the to the ground set G.

3.6 Forming the components

From this stage onwards, the algorithm essentially divides the point cloud into smaller and smaller subsets. There is a specific way of doing so, and the details are presented in this and the following sections; components 3.6, segments 3.7, regions 3.8. To sum the division process up, the respective relations are as follow:

$$\text{region} \subset \text{segment} \subset \text{component} \subset \text{point cloud}. \tag{3.4}$$

A component is a maximum set of connected cover sets. The goal of this step is to form as many components as needed so that each cover set is in some component. In the case of an ideal set of measurements, there exists only a single[4] component, which would mean that all the cover sets are connected to each other. A non-ideal case usually requires much more than one component.

The formation of the components starts from the base of the trunk set just like in the case of the ground set. Only now the extension direction is upwards from the base. It is important to prevent the component from expanding to cover sets that are part of the ground set. The extension continues as long as the cardinality of the set keeps growing.

Once the first component has been formed, if there still exist cover sets that are not part of the ground set or the first component, additional components must be formed. The forming is done by set extension, but now the starting point can be any cover set that has not been assigned to a component.

Mathematically, the formation of components can be expressed as in equation (3.3). Let C_k^i be the k^{TH} extension of the i^{TH} component. Such sets must satisfy the following conditions for all components $i = 1, 2, \ldots$ and all stages $k = 1, 2, \ldots$:

$$C_1^i = C_{\text{init}}^i \tag{3.5a}$$
$$C_{k+1}^i = \text{ext}\left(C_k^i\right) \setminus G \tag{3.5b}$$
$$C_{k+1}^i \neq C_k^i, \tag{3.5c}$$

[4]Connectedness is dependent on the radius of the cover sets and therefore so is the number of components. For example, by choosing a very large radius, all the cover sets are bound to be connected. This naturally is not the desired case since the cover sets would not preserve the local properties of the point cloud.

where G is the ground set. In the case of the first component, the set C^1_{init} is the base set of the trunk, and in the case of the other components C^i_{init} is an arbitrary cover set that has not been assigned into any component.

By definition a component can be formed even by a single cover set. This means that the component forming continues until all the cover sets have been assigned into a component.

Once the components have been formed they get filtered. Components with a small number of cover sets in them are excluded from further analysis. Also, there is still the possibility of having cover sets and components that are not part of the measured tree. Such cover sets can, for example, be part of another tree in the background. Such components are filtered by using their size, location and distance to the trunk. The cover sets in the filtered components are added to the ground set to exclude them in the remaining steps.

For the remaining components, a base has to be defined. The base is a part which can be seen as the starting point of the component. Chronologically, the base would be the *oldest* part of the piece of the tree the component presents. Intuitively, the base would be the part of the tree that has the largest radius. However, choosing the base from the cover sets is not so straightforward except for the first component for which the base is the base of the tree defined with the help of the trunk.

For the additional components, the base is chosen by inspecting the direction of the components and their distance to the trunk of the tree. There are four possible outcomes for the selection of the initial base set depending on the location of the component relative to the trunk.

- If the component is fairly parallel to and close[5] to the trunk its center point closest to the trunk is *higher* from the ground than the furthest center point, then the initial base set is the component's *highest cover set*.
- If as above, but substitute *low* for *high*.
- If the component is fairly parallel to and far from the trunk, then the component is assumed to be a branch growing upwards or downwards, but since the two can not be distinguished, the initial base set is chosen to be the component's *highest cover set*.
- Otherwise the initial base set is the cover set that is *closest* to the trunk set.

The initial base set of a component is a single cover set. The final base set is the extended set of that cover set. The base is used as the initial subset during the next step that is called segmenting.

This is not the only possible heuristic for the base forming. The presented set of rules assumes that all the components are direct extensions of the trunk, though it is also possible that a component is an extension of another component rather than the trunk. In such cases, it is irrelevant which part of the given component is closest to the trunk since the correct base would be the part closest to the parent component.

[5]Distances are measured perpendicular to the estimated trunk direction. Distance between the trunk and a component is defined to be the minimum of the distances between the mean of the trunk and the center points of the cover sets in the component.

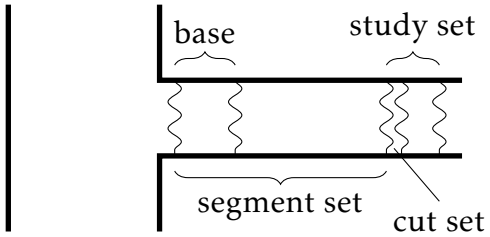


Figure 3.1: Different parts of the component during the segmentation process. The cut set is part of the study set.

3.7 Segmenting the components

Next each component is divided into *segments*. Ideally a segment presents a branch of the tree.[6] A branch has neither bifurcations nor very sharp corners. A component can have anywhere from one to thousands of segments. The segmentation process is based on the local topology of the tree. More precisely the neighboring relation is used to form and move a study set along the component. The connectedness of the set is inspected at each step, and if it becomes disconnected, starting points of new segments might have been found. This makes the process fairly independent from the size of the part of the tree presented by a component.

The segmentation process is iterative and it begins from the base of the component to be processed. During the process the *segment set* is extended one *layer*[7] at a time. Each segment formation starts from its base. In the case of the first segment in a component, the base of the segment is the base of the component. With other segments, the base is determined iteratively during the process.

On each iteration, the component is divided into two parts: the segment and its complement. The two are separated by a *cut set* which is a layer of neighboring sets of the segment. For efficiency, the topological structure of the rest of the component is approximated by a *study set* formed by the next couple[8] of layers. The cut set is part of the study set. The different sets used during the segmentation process are visualized in figure 3.1.

During each iteration the connectedness of the study set is determined. If the the study set is connected, then the cut set is included into the segment and the process advances to the next iteration. If the study set is divided into two or more disconnected parts, then further analysis is required. Each of the study region components is analysed and the following scenarios are possible.

New branch

The component of the study set presents a (usually smaller) child branch. The corresponding part of the cut set is assigned as the base for the new child branch which is later segmented similarly. The part of the cut set presenting a child branch is excluded from the set extension process.

[6] During this step the trunk is also seen as a branch.

[7] A layer is a set of the neighboring sets of the current segment set.

[8] The number of layers in the study set is dependent on the radius of the cover sets and on the radius of the component.

Extension
The component of the study set presents an extension of the current branch. In this case, it is important not to end the current segment but to continue it accordingly.

Piece of segment
A component of the study set presents small parts of the current segment. In this scenario, the small parts can be caused by short extensions of the current branch or by gaps in the measurements. If possible, the set extension should continue to such a direction.

The analysis required in the case of bifurcations is explained in detail in section 4.7. A segment has been formed when it can no longer be extended in any direction. When a segment is completed, the process continues from the possible child branches. The segmentation of a component continues recursively until each of the cover sets in the component have been assigned to a segment.

3.8 Dividing segments into cylinder-shaped regions

After the components have been divided into segments, the process continues by approximating the surface and volume of each segment. Since a segment is assumed to present a part that would be called a branch in a real tree, the volume limited by the segment is locally estimated to be cylinder-shaped. Because a branch can be curved and its radius can vary, it must be divided into smaller pieces which have as little curvature as possible and a constant radius. These pieces are called *regions*. A region is a small collection of cover sets and it is the unit which is approximated as a cylinder. Once a region is formed the change in detail from cover sets to individual measurements should be made. Otherwise same measurements could be placed in two different regions and the fitted cylinders could overlap.

As with the component and segment forming, a region is formed through a set extension process starting from the base of a region. The base of the first region is naturally the base of the segment. The region forming process has two different stages. During the first stage initial regions are formed. In the latter stage consecutive regions are joined to receive better approximations of the region properties, such as its direction. The region-forming stages are described next.

3.8.1 Initial stage

Depending on the shape of the segment, there are two different ways the set extension can happen.

If a segment is very elongated[9], the axis direction of the segment can be estimated with a high accuracy. The region is then formed by using two planes perpendicular to the axis direction. The first plane is placed under[10] the base.

[9]Elongated is determined through principal component analysis. *Very elongated* means that the segment is likely to be fairly straight and therefore it can be divided by using planes perpendicular to the axis direction.

[10]Here *under* is considered in the direction of the estimated axis.

On each step of the extension process, the second plane is moved upwards. The region is the set of points between the two limiting planes. The size and shape of the region is inspected on each step for stopping conditions, and the number of steps is also limited.

If a segment is not elongated, the extension process is based on the neighbor relations as in the case of the segments and the components. The region is extended with the neighboring cover sets of the current region. Extension into cover sets that are not part of the same segment, or that have been assigned to another region, is prevented. The maximum number of iterations is again limited, and the following additional stopping conditions are valid during both alternative ways of region forming:

- The region cannot be extended any further. The whole segment has been processed.
- The region is elongated enough. If the region is fairly elongated, the axis direction of the region can usually be approximated with a sufficient accuracy.

When a stopping condition is fulfilled, a region is successfully formed. If there are points left in the non-elongated segment, the neighboring cover sets of the finished region are set as the base for the next region. In the case of an elongated segment, the upper limiting plane is set as the lower limiting plane.

At the end of the initial stage, a cylinder is fitted to the data to find an estimate of the radius of the region. These radii are later used to estimate the radius of the segment, which is furthermore used as an error detector in the cylinder fitting process described in section 5.3.

3.8.2 Region refining

If a segment has more than one region at the end of the initial region-forming stage, then the regions are refined to achieve better estimates of their geometric properties. Otherwise the initial region is accepted as the final region with no changes.

Let R_i and R_{i+1} be the matrix presentations of the sets of cover sets forming two consecutive initial regions in a segment. Also, let $\mathbf{m}_i$ and $\mathbf{m}_{i+1}$ be the mean points of the regions, respectively. Now the axis direction estimate for the final region $\tilde{R}_i$ is

$$\tilde{\mathbf{a}}_i = \mathbf{m}_{i+1} - \mathbf{m}_i. \tag{3.6}$$

The points belonging to the final region $\tilde{R}_i$ depend on whether the initial region R_i is the first, last or one of the middle regions in the segment. In each case the accepted points are the points in the matrix $R_{i,i+1}$ which is the matrix presentation for the combination of points in the matrices R_i and R_{i+1} that satisfy the respective condition in the following set of inequalities:

$$\mathbf{r}_{i,i+1} \cdot \tilde{\mathbf{a}}_i \leq \mathbf{m}_{i+1} \cdot \tilde{\mathbf{a}}_i \qquad \text{first region} \tag{3.7a}$$

$$\mathbf{m}_i \cdot \tilde{\mathbf{a}}_i \leq \mathbf{r}_{i,i+1} \cdot \tilde{\mathbf{a}}_i \qquad \text{last region} \tag{3.7b}$$

$$\mathbf{m}_i \cdot \tilde{\mathbf{a}}_i \leq \mathbf{r}_{i,i+1} \cdot \tilde{\mathbf{a}}_i \leq \mathbf{m}_{i+1} \cdot \tilde{\mathbf{a}}_i \qquad \text{otherwise} \tag{3.7c}$$

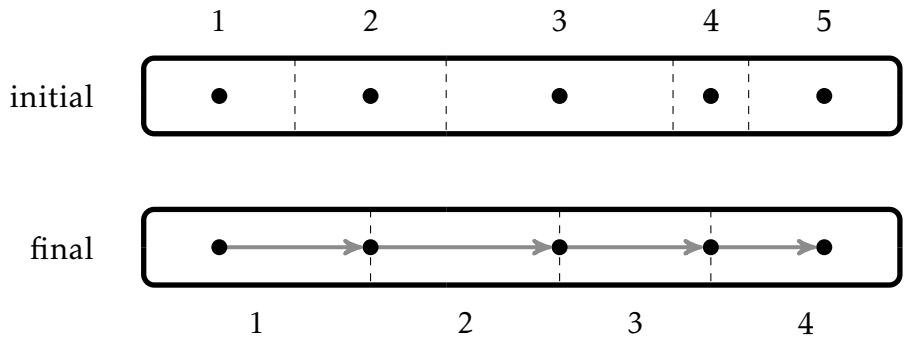


Figure 3.2: Stages of the region forming process. The regions from the initial stage are combined as shown in the figure during the final stage. The dashed line shows the limits of each region in the segment. The circles are the mean points of each initial region. The vectors indicate how the axis direction estimates are computed during the final stage.

where $\mathbf{r}_{i,i+1}$ is a row vector of $R_{i,i+1}$ giving the location of a point and $(\cdot)$ is the dot product operator. In other words when forming the first one of the final regions, points below the mean of the next initial region are included. Similarly, in the case of the last final region, the points above the previous mean are included. Figure 3.2 illustrates the point selection process.

3.9 Fitting a cylinder to a region

A region should be straight, have no bifurcations, and have a constant radius. If these conditions hold, and the noise in the measurements is small compared to the radius, then the surface of the region can be approximated as a cylinder. The cylinder presentation is chosen in a least-squares sense. The fitting methods are discussed in detail in section 5.1. Two different approaches are presented for the given problem, but due to the complexity of the fitting problem both methods require good initial values.

The following table lists how the initial cylinder properties can be found through a geometric analysis of the point cloud forming the region.

Axis direction
: If a segment has more than one region, the axis direction estimate is given by equation (3.6). Otherwise the direction can be found through principal component analysis.

Axis point
: A point through which the axis goes can be approximated with the mean of the points, since it is likely to be in the middle of the point cloud.

Radius
: Once the axis point and the axis direction are known, the radius can be estimated as the average distance between the points and the cylinder axis.

Once the initial values are computed, the cylinder fitting process can begin. The process utilizes both the cylinder fitting method described in section 5.1.1

and the circle fitting in section 5.1.2. The fitting process includes a lot of error detection and correction which are discussed in sections 5.3 and 5.4.

In short, a cylinder is fitted to the measurements[11] included in the cover sets in the region with the initial values described above. The length of the cylinder is computed by projecting all the points on to the axis of the resulting cylinder and by taking the difference of the minimum and maximum of the projections. Certain error estimates are then computed for the axis direction, axis point, and the radius.

If the error is too large, points with the largest error are removed, initial values are updated and a new fitting is done. This is repeated for a limited number of times until a suitable solution is found or all the attempts are used.

The cylinder fitting is attempted for each region in every segment always starting from the region including the base of the segment. The segment processing order is the same as the order in which they where formed. This ensures that the relational information of the branches can be passed to the cylinder presentation of the tree. Once all the regions, segments, and components have been processed, the cylinder model is complete. Examples of cylinder models found through the described modeling process are presented in figure 3.3.

3.10 Analysing the cylinder model

The cylinder model contains the values for the radius, length, starting point, and axis direction for each of the cylinders. Each cylinder also has a type, which for the moment is simply a division to trunk and branch cylinders. In addition, the following relation information is available for each cylinder.

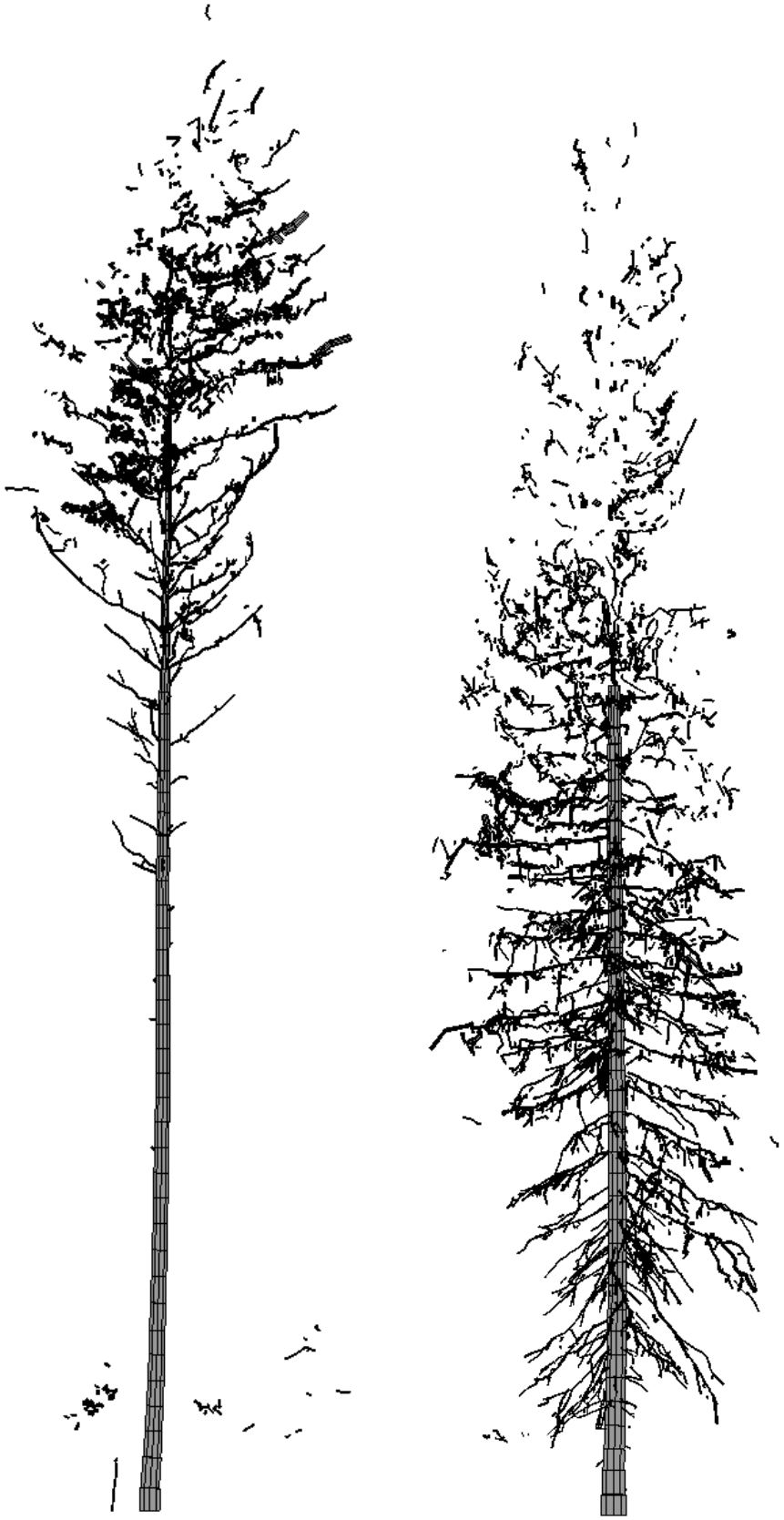
Property	Number of	Description
Parent	$0 \vee 1$	Reference to the parent cylinder from which the given cylinder is an extension of or a bifurcated child branch of.
Extension	$0 \vee 1$	Reference to a cylinder that is part of the same branch and starts right after the given cylinder in the axis direction.
Children	$0 \ldots N$	References to branches that are bifurcated from the part of the branch that the given cylinder presents.

Computing volumes The total volume or many interesting partial volumes of the tree can be computed with

$$V = \sum_{i=1}^{N} \pi h_i r_i^2 \cdot p(i), \tag{3.8}$$

where N is the number of the cylinders, h_i is the length and r_i the radius of the i^{TH} cylinder and $p(\cdot)$ a function that has the value 1 for the cylinders with the

[11] From this point on, all the measurements are used, not just the center points of the cover sets.



(a) Pine **(b)** Spruce

Figure 3.3: Example cylinder models produced by the tree analysis process. The cylinder model were derived from laser scanning data provided by the Finnish Geodetic Institute.

wanted qualities and 0 for the others. For example for the total volume of the tree we have

$$p(i) = 1, \quad \forall i \in \{1, \ldots, N\}, \tag{3.9}$$

for the trunk volume

$$p(i) = \begin{cases} 1 & \text{if cylinder } i \text{ in trunk} \\ 0 & \text{otherwise,} \end{cases} \tag{3.10}$$

and for the volume of the branches and parts of the trunk with a larger radius than a given limit $r_{\min}$

$$p(i) = \begin{cases} 1 & \text{if } r_i > r_{\min} \\ 0 & \text{otherwise} \end{cases}. \tag{3.11}$$

Branch size distribution The distribution of the branch length or the branch volume as a function of the branch radius can be computed easily from the cylinder model. The cylinder radii are categorized into intervals. Denoting the limits for the K intervals $[a_1, a_2, \ldots, a_{K+1}]$, the total length $l(k)$ of the branches with a radius in the interval $k \in \{1, \ldots, K\}$ is

$$l_k = \sum_{j \in J_k} h_j, \qquad J_k = \{\, i \mid a_k \leq r_i < a_{k+1} \,\}. \tag{3.12}$$

Similarly, the total volume in the given interval is

$$v_k = \sum_{j \in J_k} \pi h_j r_j^2, \qquad J_k = \{\, i \mid a_k \leq r_i < a_{k+1} \,\}. \tag{3.13}$$

By computing $l_1, l_2, \ldots, l_K$ or $v_1, v_2, \ldots, v_K$, the *branch length* or *volume distribution* is found, respectively. Examples of the two distributions are plotted in figure 3.4.

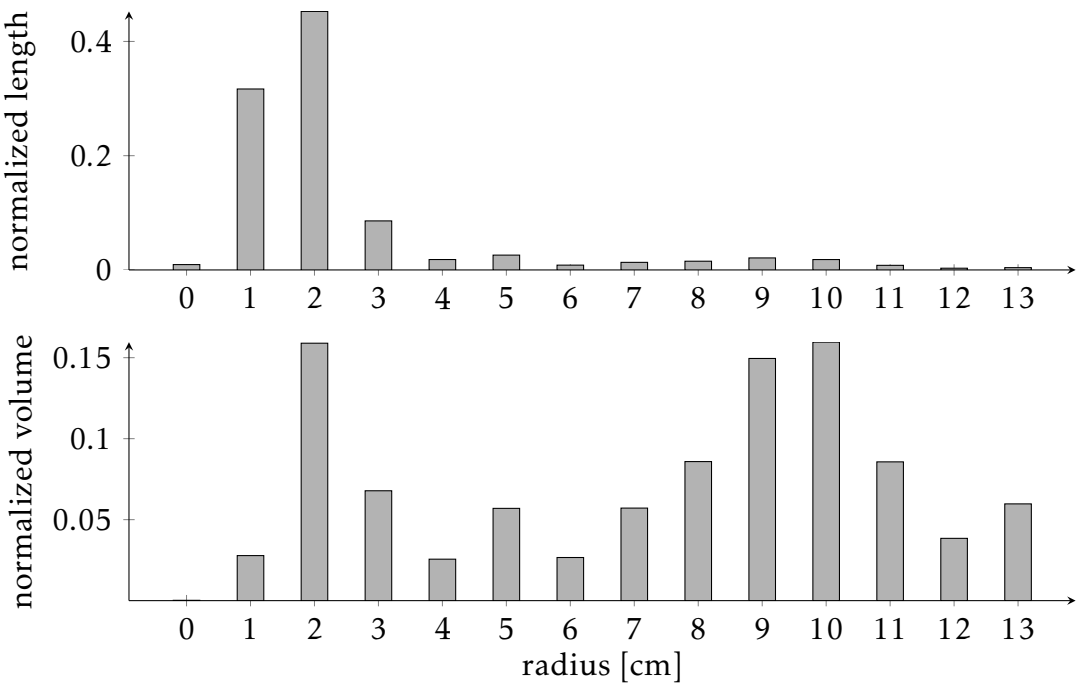


Figure 3.4: Sample size distributions of a pine tree. The upper figure shows how the length of the branches is distributed as a function of the radius. It is clear that this distribution is dominated by the small branches. The lower figure shows the distribution of the volume of the branches. Contrary to the length, the volume is distributed more equally. Cylinders with radii less than 1 cm have been binned to zero radius.

4 METHODS

In this chapter certain key methods that are used in the algorithm are presented. The methods are in no specific order, and most of them are used in multiple steps of the algorithm. The steps related to each method are mentioned in the text, but a complete list can be found in table 3.1. Some of the methods, e.g. the partitioning in section 4.1, are more involved with the implementation of the algorithm, while others form the fundamental basis of the mathematical algorithm itself.

In section 4.1 a fast method, called partitioning, for finding the closest point of a given point is presented. This method is used in the early steps of the algorithm to form the cover sets, discussed in section 4.2, which are used throughout the rest of the algorithm. Partitioning can also be used in the post-processing stage after the cylinder fitting process to optimize the computational time.

Cover sets are small point clouds that can be *characterized* and *classified* according to their geometric properties. The methods required to accomplish these tasks are presented in sections 4.3 and 4.4, respectively. Additionally, methods for characterizing the direction of a point cloud are discussed in section 4.6.

The cover sets also provide a way for changing scale; i.e. moving from the local details to more comprehensive, global qualities. This is done by *set extension* which is discussed in section 4.2.1.

In section 4.7, the details of the bifurcation detection during the segmentation process of the algorithm are presented. The section explains how the study region of the segment is analysed to detect a possible bifurcation. Additionally, a method for compensating for the inaccuracy of the bifurcation detection is presented. This method is called *base extension* and it is presented in section 4.7.1.

4.1 Partitioning

When analysing a point cloud P, a problem that often arises is the definition of *closeness*. If the set P is a subset of a metric space (M, d) one can use the induced metric d_P to measure distances and choose a parameter $r \in \mathbb{R}$ which defines the upper bound for the distance at which the points are still close to the point $\mathbf{p}$. This defines a so called *open r-ball* which is the set

$$B_r(\mathbf{p}) = \{\mathbf{x} \in P \mid d_P(\mathbf{p}, \mathbf{x}) < r\}. \tag{4.1}$$

The simplest way to find the points that are close to a given point is to compute the distance between the given point and all the other points in the set. The closest points satisfy the condition 4.1 and the other points do not. However, this *brute force* -method[1] will require an enormous number of computations when the cardinality of the point cloud is large. Also, it is likely that most of the computations are futile since not nearly all of the points are expected to satisfy the given condition (4.1).

[1] More on the brute force -method in section 4.1.2.

Example 1. *If the cardinality* $N = 10^6$ *for the point cloud* P*, then* $10^6 - 1$ *computations are required in order to find the nearest points for a single point. Furthermore, approximately* $N^2 = 10^{12}$ *computations are required to find the nearest points for all the points.*

Now if for each point there are about 10^2 *points that satisfy the closeness condition, then the number of* futile *computations is* $10^6 - 10^2 \approx 10^6$ *for a single point. That corresponds to* 99.99%*.*

An alternative way of finding the closest points of an element in a set is presented next. The algorithm is based on constructing a convenient *partitioning* of the given set.

4.1.1 r-cube partition of a set

Let $P \subset \mathbb{R}^n$ be a point cloud and the parameter $r \in \mathbb{R}$. The set

$$C_P = \{\mathbf{x} \in \mathbb{R}^n \mid \forall i = 1 \dots n \quad \min_{\mathbf{y} \in P} y_i \leq x_i \leq \max_{\mathbf{y} \in P} y_i + r\}, \tag{4.2}$$

where x_i is the i^{TH} component of the point $\mathbf{x}$, is a *hyperrectangle*[2] and $P \subset C_P$.

The constructed hyperrectangle can be divided into *hypercubes*[3] with an edge length r. The number of required cubes N_i in a dimension i is

$$N_i = \left\lfloor \frac{\max_{\mathbf{y} \in P} y_i - \min_{\mathbf{y} \in P} y_i + r}{r} \right\rfloor \tag{4.3}$$

$$= \left\lfloor \frac{\max_{\mathbf{y} \in P} y_i - \min_{\mathbf{y} \in P} y_i}{r} \right\rfloor + 1 \tag{4.4}$$

and the total number of cubes is $N = \prod_{i=1}^{n} N_i$. The cubes are indexed in each dimension i with indices going from 1 to N_i. The index number grows when moving to the positive axis direction. Each cube can be identified with an index vector $\mathbf{w}$ for which

$$\forall i = 1 \dots n \quad w_i \in \mathbb{N} \,:\, 1 \leq w_i \leq N_i \tag{4.5}$$

holds.

Definition 1. *The* r-cube *of a point* $\mathbf{p}$ *in the set* $P \subset \mathbb{R}^n$ *is the hypercube*

$$C_{\mathbf{w_p}} = \{\mathbf{x} \in P \mid \forall i = 1, \dots, n, \quad r\Big((\mathbf{w_p})_i - 1\Big) \leq x_i - \min_{\mathbf{y} \in P} y_i < r(\mathbf{w_p})_i\}, \tag{4.6a}$$

$$(\mathbf{w_p})_i = 1 + \left\lfloor \frac{p_i - \min_{\mathbf{y} \in P} y_i}{r} \right\rfloor \quad i = 1 \dots n. \tag{4.6b}$$

Every point in a point cloud P has a uniquely defined r-cube. Multiple elements of P can have the same r-cube.

[2] A hyperrectangle is a generalization of a rectangle in higher dimensions.

[3] Hypercube is a generalization of a cube in larger dimensions. Each edge of a hypercube has the same length.

Definition 2. *The index vector* $\mathbf{w}$ *corresponding to the r-cube of a point* $\mathbf{p}$ *is called the r-*cube index *of the point p in set P.*

Definition 3. *For a set* $P = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ *and the parameter* $r > 0$*, the set*

$$P_P(r) = \{C_{\mathbf{w}_{\mathbf{x}_1}}, C_{\mathbf{w}_{\mathbf{x}_2}}, \ldots, C_{\mathbf{w}_{\mathbf{x}_m}}\} \tag{4.7}$$

*is a partition of the set P. This specific partition is called the r-*cube partition *of the set P.*

Theorem 1. *Let* $\mathbf{x}$ *and* $\mathbf{y}$ *be elements of a point cloud* $P \subset \mathbb{R}^n$ *and the parameter* $r \in \mathbb{R}, r > 0$*. If* $d(\mathbf{x}, \mathbf{y}) < r$ *then*

$$\left|(\mathbf{w}_\mathbf{x})_i - (\mathbf{w}_\mathbf{y})_i\right| \leq 1 \quad \textit{holds} \quad \forall i = 1 \ldots n, \tag{4.8}$$

where the vectors $\mathbf{w}_\mathbf{x}$ *and* $\mathbf{w}_\mathbf{y}$ *are the r-cube indices of the points* $\mathbf{x}$ *and* $\mathbf{y}$*, respectively.*

Proof. According to definition 1 for all indices $i = 1 \ldots n$

$$r\left((\mathbf{w}_\mathbf{x})_i - 1\right) \leq x_i - \min_{\mathbf{z} \in P} z_i < r(\mathbf{w}_\mathbf{x})_i,$$
$$r\Big((\mathbf{w}_\mathbf{y})_i - 1\Big) \leq y_i - \min_{\mathbf{z} \in P} z_i < r(\mathbf{w}_\mathbf{y})_i.$$

The inequalities that are particularly interesting are

$$x_i - \min_{\mathbf{z} \in P} z_i \geq r\left((\mathbf{w}_\mathbf{x})_i - 1\right),$$
$$y_i - \min_{\mathbf{z} \in P} z_i < r(\mathbf{w}_\mathbf{y})_i.$$

By multiplying the bottom inequality by (-1) and summing the result with the top inequality, we obtain

$$x_i - \min_{\mathbf{z} \in P} z_i - \left(y_i - \min_{\mathbf{z} \in P} z_i\right) \geq r\left((\mathbf{w}_\mathbf{x})_i - 1\right) - r(\mathbf{w}_\mathbf{y})_i.$$

This can be presented in the form

$$\frac{x_i - y_i}{r} + 1 \geq (\mathbf{w}_\mathbf{x})_i - (\mathbf{w}_\mathbf{y})_i. \tag{4.9}$$

Since $d(\mathbf{x}, \mathbf{y}) < r$ it follows that

$$\left|x_i - y_i\right| < r \quad \forall i = \{1, 2, \ldots, n\}. \tag{4.10}$$

To simplify the proof, it is assumed that $x_i \geq y_i$. This is a justified assumption since the analysis is done in each dimension separately, and one can always swap the points to make the assumption hold. Due to this assumption, equation (4.10) takes the form

$$x_i - y_i < r \quad \forall i = \{1, 2, \ldots, n\}. \tag{4.11}$$

By substituting this to the inequality (4.9) we have

$$2 = \frac{r}{r} + 1 > \frac{x_i - y_i}{r} + 1 \geq (\mathbf{w}_\mathbf{x})_i - (\mathbf{w}_\mathbf{y})_i. \tag{4.12}$$

Because $x_i \geq y_i$ then $(\mathbf{w_x})_i \geq (\mathbf{w_y})_i$ and $\left|(\mathbf{w_x})_i - (\mathbf{w_y})_i\right| = (\mathbf{w_x})_i - (\mathbf{w_y})_i$. The inequality (4.12) can be written as

$$\left|(\mathbf{w_x})_i - (\mathbf{w_y})_i\right| < 2. \tag{4.13}$$

Since the cube indices are natural numbers, the inequality (4.13) limits their difference to either 0 or 1 when $d(\mathbf{x}, \mathbf{y}) < r$.

It is trivial to find examples where $\left|(\mathbf{w_x})_i - (\mathbf{w_y})_i\right|$ is either zero or one and therefore these examples are omitted here. □

Corollary 2. *Let $P \subset \mathbb{R}^n$ be a point cloud and the point $p \in P$ and the parameter $r > 0$. There are at most 3^n candidate r-cubes that can contain points whose distance to the point p is less than the number r.*

Proof. Theorem 1 says that if the distance between two points is less than $r > 0$ the the corresponding r-cube indices may differ at most by one. This means that there are three possible values,–1, 0 and 1, for the difference of the cube indices in each dimension. Since there are n dimensions, the number of candidate cubes is 3^n. □

Figure 4.1 illustrates the location of the candidate cubes in a 2-dimensional case.

4.1.2 Computational time

Let the cardinality of the set P be M. One way to find the nearest elements for a given element in the set P is to use brute force. This means that all the elements in the set are tested against a closeness condition. In this approach, it would require $M - 1$ computations to find all the nearest points for a single given point. Furthermore, if the same is repeated for all the elements, the total computational time t_{brute} for this brute force approach is roughly

$$t_{\text{brute}} \propto \frac{M}{2}(M - 1). \tag{4.14}$$

The computational time of performing the same task with the r-cube partition –method depends on the cardinality N of the r-cube partition, the shape of the point cloud P and the distribution of the points in the point cloud. If the points in the set P are assumed to be distributed evenly in all the cubes, then the number of points in each cube is $\frac{M}{N}$. According to theorem 2 for each point in the set P, there are 3^n candidate r-cubes whose elements have to be tested. Each cube requires $3^n \cdot \frac{M}{N}$ computations and there are N cubes. The computational time of constructing the r-cube partition is proportional to the cardinality M. With the even distribution assumptions, the total computational time $t_{r\text{-cube}}$ of the method is

$$t_{r\text{-cube}} \propto M + \frac{3^n}{N}M^2 \tag{4.15}$$

$$= M(M\frac{3^n}{N} + 1). \tag{4.16}$$

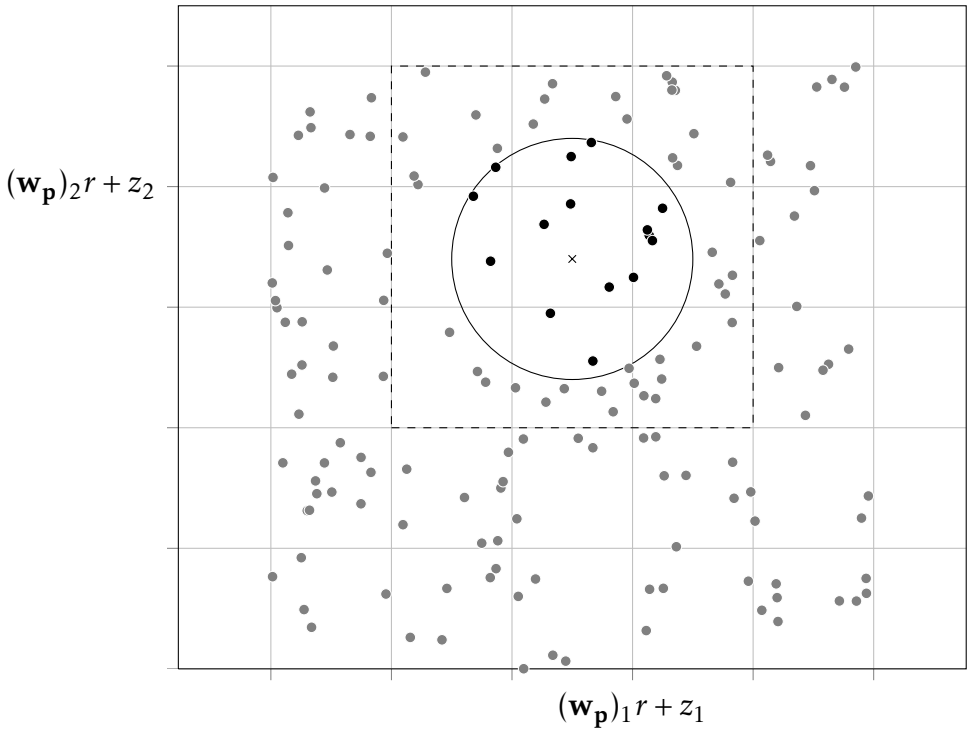


Figure 4.1: An example of r-cube partitioning of a point cloud P in a two-dimensional case. A point p is denoted by the x-mark. The large circle has a radius of r. The gray squares are the r-cubes of the point cloud. The r-cubes that could contain points whose distance from the point p is less than r are surrounded with a dashed line. The distance between point p and the black points is less than r and the distance to the gray points is larger or equal to r. The number $z_1 = \min_{y \in P} y_1$ and $z_2 = \min_{y \in P} y_2$.

When the two computational times from equations (4.14) and (4.15) are compared, we see that

$$t_{r\text{-cube}} < t_{\text{brute}} \tag{4.17}$$

when

$$N > \frac{2}{1 - \frac{3}{M}} \cdot 3^n \approx 2 \cdot 3^n, \quad (M \gg 1). \tag{4.18}$$

The result means that the r-cube partition –method is faster if the total number of cubes is larger than twice the number of candidate cubes for a single point (see theorem 2).

Example 2. *Let there be* 10^6 *evenly distributed points in a point cloud* $P \subset \mathbb{R}^3$ *and the point cloud is partitioned into* 10^4 *cubes. Then the computational time for the brute force approach is* $t_{brute} \propto 10^{12}$ *and the respective time for the r-cube partition approach is* $t_{r\text{-}cube} \propto 10^9$*. The latter time is about 0.1 percent of the former.*

4.1.3 Applications of the r-cube partition

As shown previously in this section, the r-cube partition can be used to optimize the computational time required to find the closest points of a given point. Such optimizations are done in the cover set creation and the cylinder fitting post-processing stages of the tree modeling algorithm.

The r-cube partition has the great advantage that if the point cloud does not change the partition has to be computed only once. Even if points are added into or removed from the point cloud, the partition info can be updated without computing everything from the start.

In the tree modeling algorithm, the r-cube partition of the initial input point cloud can be used also for filtering out small, isolated subsets of the point cloud as mentioned in section 3.1.1. If a given r-cube has a cardinality smaller than a given value and all of the neighboring r-cubes are empty sets, then the points in the given cube can be excluded from further analysis since they are not connected to other parts of the point cloud. Filtering is described in more detail in section 4.5.

4.2 Cover sets

Let P be a point cloud. A cover set of the set P is a subset of the point cloud. The union of all the cover sets is the complete point cloud. The collection of the cover sets is called a cover of the point cloud. There are of course numerous ways of forming a cover of a given set, but the one described here uses spherical neighborhoods of randomly selected points in the point cloud.

Let the number r be the radius used for the spherical environments and the number $d \leq r$ a parameter for minimum center point distance. The algorithm for creating a cover for a point cloud P can be described by the following steps. In the beginning, all points are available for center point selection and none of them are yet part of the cover.

1. Choose a random point $\mathbf{p}_i \in P$ that is available for center point selection.
2. Form the cover set C_i using the r-ball environment of the selected point:

$$C_i = B(\mathbf{p}_i, r), \tag{4.19}$$

3. Mark the points in the set C_i as classified.
4. Mark the points in $B(\mathbf{p}_i, d)$ as not available for center point selection.
5. Repeat until no available center points remain in the set P.

The r-cube partition (see section 4.1 for details) can be used in the implementation of the algorithm to form the sets $B(\mathbf{p}_i, r)$ and $B(\mathbf{p}_i, d)$ fast.

The magnitude of the cover set radius r should be chosen according to the structure of the scanned object, which means that prior information is required. If the magnitude of the radius is too large, the finer details of the object cannot be reconstructed since their geometric properties are simply not captured by the cover sets. On the other hand, if the radius is too small, the point cloud is likely to become divided into many disconnected components which again reduces the reconstruction level. For trees, a rule of thumb is that the cover set radius should be of the same order of magnitude as the radius of the smallest branches in the tree. The effects of the cover set radius on the reconstruction level are studied in section 6.1.3.

The parameter d is often chosen to be equal or slightly smaller than the parameter r. Since d controls the allowed amount of intersection, the former selection prevents a point that has been chosen as a center point from being in multiple cover sets. The latter would allow a center point to be in other cover sets as well, which would make the cover more dense. Decreasing the value of the parameter d means increases in the number of cover sets and the computational time of the cover generation.

As mentioned above, two cover sets can intersect, and nearby sets should intersect. If the cover sets A and B intersect, B is called a *neighbor set* of the set A and vice versa. The neighbor relation provides the basis for the *connectedness* of the cover sets which in turn can be used to determine the connectedness of the subsets of the point cloud P. The neighbor relation is also the basis for *set extension*.

4.2.1 Set extension

The cover sets contain information about the details of the surface on a very local scale. However, for the algorithm to be useful, a change in scale is often required. The intersecting cover sets offer a way to extend the selection of points from a single cover set to a larger part of the point cloud presenting the surface. The extension moves along the surface as long as the cover sets conform to the surface. The basis for the extension is presented next.

Let $C = \{C_i\}$ be a cover of a point cloud P. If the neighbor relations are known for each cover set, then an arbitrary cover set, or a collection of cover sets, C_{init} can be extended n times easily with the following algorithm.

1. The initial set $E_0 = C_{\text{init}}$.

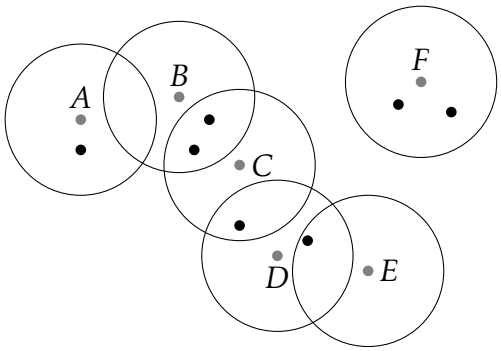


Figure 4.2: Examples of connected and disconnected subsets of a cover set. The cover sets are labelled according to the corresponding center point shown in gray. Additional points are shown as black dots. The circular environments used in forming the cover sets have been drawn as well. The set $B \cup C \cup D \cup E$ is connected. The set $E \cup F$ is disconnected. Even though the balls of the cover sets A and B intersect on the embedding plane, the set $A \cup B$ is disconnected since the cover sets do not have any common points.

2. For each consecutive step

$$E_{i+1} = \text{ext}\,(E_i) = E_i \cup \bigcup_{C_j \cap E_i \neq \emptyset} C_j. \tag{4.20}$$

3. Repeat step 2 for n times or until $E_{i+1} = E_i$ for some $i < n$.

The set generated by extending a set S for n times is called the n^{TH} order extension of the set S and noted $\text{ext}^n\,(S)$.

Connectedness of a set was defined in section 2.2. Set extension offers an alternative definition for the connectedness of a point cloud.

Definition 4. *The point cloud P with a cover $C = \{C_i\}$ is* connected *if there exists an index j for every cover set C_i so that*

$$\text{ext}^j\,(C_i) = P. \tag{4.21}$$

Otherwise if such an index does not exist for some cover set, then the point cloud is disconnected.

The definition states that a set is connected if any one of its cover sets can be extended to be equal to the whole set. Examples of connected and disconnected sets can be seen in figure 4.2.

4.3 Characterization

Once the cover sets are formed, certain geometric characteristics can be computed for each set. These characteristics are used later in the analysis process to classify and filter cover sets in or out. The computation of the characteristics is based on the *principal component analysis* described in section 2.5.3, and the characteristics are presented next. The computations are done for each cover set

separately, but in the following presentation they are simply called sets or point clouds, since the definitions are valid for any arbitrary point cloud.

In the definitions in this section let $P \subset \mathbb{R}^3$ be a point cloud and vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ be the normalized eigenvectors and the numbers $\lambda_1, \lambda_2, \lambda_3$ be the eigenvalues corresponding to the first, second and third largest principal components of the set P, respectively.

4.3.1 Relative characteristics

In this section, the characteristics of a point cloud that are independent from the scale of the set are presented. The first three definitions contain characterizations of the *dimensionality* of the set.

Definition 5 (Elongatedness)**.** *The* elongatedness $\mathrm{elo}(P)$ *of a set P is the ratio*

$$\mathrm{elo}(P) = \frac{\lambda_1 - \lambda_2}{\lambda_1}. \tag{4.22}$$

The greater the elongatedness value, the more elongated the set is. The elongatedness value is always between 0 and 1.

Example 3. *If all the points in set P are on the same line, the then* $\mathrm{elo}(P) = 1$ *since* $\lambda_2 = 0$*. If the set P is formed by the four corners of a square on some plane in* R^3*, then* $\mathrm{elo}(P) = 0$ *because the variance is the same in the direction of the first and second principal component, which means* $\lambda_1 = \lambda_2$*.*

Definition 6 (Planarness)**.** *The* planarness $\mathrm{plan}(P)$ *of a set P is the ratio*

$$\mathrm{plan}(P) = \frac{\lambda_2 - \lambda_3}{\lambda_1}. \tag{4.23}$$

If a set is planar, or *two-dimensional*, the variance in the direction of the third principal component is close to zero. This means that λ_3 is small and $\mathrm{plan}(P) \approx \frac{\lambda_2}{\lambda_1}$. Now since the set is planar $\lambda_2 \approx \lambda_1$ and $\mathrm{plan}(P) \approx 1$.

Example 4. *In the case of the previous example of the square, the set is very planar since* $\lambda_3 = 0$*,* $\lambda_2 = \lambda_1$ *and* $\mathrm{plan}(P) = 1$*. In another case where the set P is formed by the eight corners of a cube in* $\mathbb{R}^3$*,* $\mathrm{plan}(P) = 0$ *because the variance is the same in all the directions of the principal components, which means* $\lambda_2 = \lambda_3$*.*

Definition 7. *A set P is said to be* three-dimensional *if the ratio*

$$\mathrm{d}_3(P) = \frac{\lambda_3}{\lambda_1} \tag{4.24}$$

is close to one.

If the set is elongated or planar, $\lambda_3 = 0$ and $\mathrm{d}_3(P) = 0$.

Example 5. *In the previous example of the cube,* $\mathrm{d}_3(P)$ *is exactly* 1 *since* $\lambda_1 = \lambda_2 = \lambda_3$*.*

The sum of the previous *dimensionality* indicators is always one:

$$\mathrm{elo}(P) + \mathrm{plan}(P) + \mathrm{d}_3(P) = 1. \tag{4.25}$$

This follows directly from the definitions.

Definition 8 (Parallel-to-trunk value)**.** *The* parallel-to-trunk *value* $\mathrm{par}(P)$ *is*

$$\mathrm{par}(P) = 1 - |\mathbf{v}_3 \cdot \mathbf{u}|, \tag{4.26}$$

where $\mathbf{v}_3$ *direction of the normal of the set and* $\mathbf{u}$ *is the estimated trunk direction.*

Usually the trunk direction $\mathbf{u}$ is estimated to be the standard unit vector $\mathbf{e}_3 = [0, 0, 1]$.

In the case of an elongated set P, if the angle between the two vectors $\mathbf{v}_3$ and $\mathbf{u}$ is large, meaning the two vectors are nearly perpendicular, then the absolute value of their dot product is close to 0 and $\mathrm{par}(P)$ is close to 1. Similarly, if the vectors are close to parallel, the absolute value of their dot product is close to 1.

4.3.2 Absolute characteristics

The geometric characteristics presented above are *relative* since in the case of dimensionality all the values are ratios between absolute measures, and in the case of the parallel-to-trunk value, normalized direction vectors are used. In addition to the previous ones, there are ways to present *absolute* geometric properties of a point cloud. One such way is presented next.

As before, let $P \subset \mathbb{R}^3$ be a point cloud and vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ be the normalized eigenvectors of the covariance matrix of P. Also let M be a matrix that contains the elements of P as its row vectors.

Definition 9. *The* length of a set *is obtained by projecting all the points in a set P onto the eigenvector corresponding to the largest principal component and by taking the difference of the minimum and maximum of the computed values. The length of a set P is*

$$\max(M\mathbf{v}_1) - \min(M\mathbf{v}_1). \tag{4.27}$$

Similarly, it is possible to define the width and height of a set by projecting the points onto $\mathbf{v}_2$ and $\mathbf{v}_3$, respectively.

The absolute characteristics can be used alongside the relative ones to find cover sets with even more specific properties. For example, one can filter out points that are elongated, non-parallel to the trunk and more than 15 centimeters in length.

It should be noted that the use of absolute characteristics requires knowledge about the scale of the point cloud. This means that in the case of classification according to absolute characteristics, the same limiting conditions will not work, if for example, the unit system is changed.

Since the length and other physical dimensions of a set are computed with the help of the minimum and maximum values, the results are somewhat sensitive to measurement errors.

Example 6. *A one-dimensional cover set, with the radius* $r = 1$*, is formed to the point* $x = 1$*. The set has 100 elements uniformly distributed on the interval* $[0, 1]$ *and one element at* 1.9*. According to definition 9 the length of the set is* 1.9*, even though the single point at* $x = 1.9$ *is probably an outlier.*

Table 4.1: Characteristic criteria used in the initial classification of the cover sets. Note that if there is much undergrowth the planarness value of the ground can be low also, but this is just the initial classification.

Property	Ground	Trunk	Branch
elo(S)	low	low	high
plan(S)	high	high	low
par(S)	low	high	low

Problems described in example 6 can be helped by discarding a few of the extreme results after the projection. The downside is that the accuracy in cases with no outliers will also decrease. Despite the problems, the presented method will give sufficient estimates in most cases.

Absolute characteristics can be used to filter out parts that are so small that they cannot be accurately analysed. The precision of the laser scanner sets one such limitation.

4.4 Classification

Geometric characteristics of the cover sets are used to classify the cover sets into useful categories. Such categories are the *trunk* points, *branch* points and the *ground* points. Classification is done in order to locate the base of the tree, but in order to achieve this, the trunk must be found first. The trunk is easily identified in the point cloud since it is fundamentally different from the rest of the tree. The base of the tree is simply the lowest part of the trunk, and once the base of the tree is located, it is fairly easy to remove the measurements of the ground and undergrowth.

Ground points are the measurements reflected from the ground near the tree that was scanned. These points are not used in the final tree analysis process and they need to be filtered out. The filtering of ground points is described in detail in section 4.5.2.

The classification process starts with an initial classification, where the criteria shown in table 4.1 is used. Cover sets that meet the corresponding criteria are classified as parts of the ground, branches, or trunk.

The initial classification is likely to result in some false positives, and so the size of the connected components of the classified sets is checked, and only the largest connected sets form the initial set for the trunk.

4.5 Filtering

Most of the points in a point cloud produced by a laser scanner are samples of the surface being analysed. The point cloud can however contain other points that are not needed, or that can even disturb the analysis process. There are three main causes for unwanted points:

Ground
Measurements reflected from the ground surrounding the object tree are futile in the analysis of the tree. Typical measurements will contain at least a small part of ground and possible undergrowth.

Other trees
Other trees are likely to grow in the vicinity of the target tree of the scanning. Measurements can be reflected from such trees next to, or behind the target tree. Such points should be easy to remove unless the branches of the trees are in contact with the target tree.

Noise
When a laser beam hits a target whose magnitude is close to its own radius, it can scatter and cause multiple measurements that appear to have reflected from mere air. Such *phantom* measurements are not that common but they do happen. These points can be characterized as *noise*.

Since the input point cloud can contain measurements that cannot and should not be used in the analysis, the point cloud must be *filtered*. This means that all the measurements with a source described above must be identified and discarded from the actual analysis.

The filtering of the point cloud is divided to three different stages. The first stage is during the preprocessing of the point cloud data. At this time, points that have clearly been reflected from objects *far* away are discarded.

The other two stages are during the algorithm, and they are presented in sections 4.5.1 and 4.5.2. The first filtering removes small groups of isolated points, which can be either noise or irrelevant measurements. The second filtering identifies the ground and components that are part of some other tree, undergrowth or some other irrelevant object.

4.5.1 Initial filtering

At the beginning of the algorithm when forming the cover sets, an r-cube partition of the point cloud is created as described in section 4.1. In addition to assisting with the cover set formation, the partition can be used as a tool for filtering. It should be noted that this *initial* filtering does not consider the structure of the tree in any way, and the same method could be used in other applications also.

The number of elements in each r-cube is checked. If an r-cube has a number of elements that is lower than a given threshold and none of the 3^n-1 neighboring cubes have any points in them, then the points in the r-cube are considered to be isolated. Such points cannot be connected to any cover sets with the radius r because the distance to a point outside the 3^n cubes is always larger than r.

The r-cube based filtering is a good tool for removing the noise, i.e. phantom points, from the measurement data. Since the probability of a phantom measurement is fairly small, it means that such a measurement is likely to be the only, or one of the few, measurements in an r-cube. Furthermore, the neighboring cubes are also likely to be empty. Therefore such measurements are considered to be isolated and are discarded from the analysis.

Even if all the neighboring cubes of an r-cube are empty, it is possible to use the points inside the center cube and fit one or more cylinder to those points. This explains the limiting number of points in the initial filtering process. Such a cylinder would not be *connected*[4] to the rest of the tree but it would still contribute to the volume and branch size distribution of the tree.

During the initial filtering, duplicate points can also be removed. This means that if any two or more points $p_1, p_2, \ldots, p_k$ in any neighboring r-cubes of the point cloud satisfy the condition $d(p_i, p_j) < \varepsilon$ for any indices $i, j \in \{1 \ldots k\}, i \neq j$ and some small threshold ε, only one of the points should be preserved and others discarded. This can be done to make the computations faster; however, computing the distances is in itself time-consuming and therefore the duplicate point removal can also be skipped.

4.5.2 Excluding the ground

Another part of the tree analysis process where points are discarded from further analysis is after the geometric characteristics, described in section 4.3, have been computed. At this stage it is possible to remove the measurements that do not represent any part of the tree that is being analysed. Such measurements have been reflected either from the ground, undergrowth, or possible smaller trees near the tree in question.

The filtering at this stage is done based on the geometric characteristics of the cover sets and the general knowledge of tree structure. Since all of the conditions that are described here have exceptions, the conditions should be viewed as probabilities.

A cover set that is not part of the tree is likely to have a small parallel-to-trunk value since the trunk is in most cases perpendicular to the ground. Also, the planarness value of the cover set should be high since not a great curvature is expected for a piece of ground.

This qualification of a ground cover set can produce many false positives and therefore the filtering process must be continued before removing any points. To do this it is necessary to find an approximation of the ground level.

The *initial ground set* is the largest connected set of cover sets classified as ground according to the definition above. The ground level is defined to be the mean of the heights of the points in the initial ground set. Here the height is measured in the estimated direction of the trunk.

Once the ground level is computed it is easy to filter out points. The initial ground set can be extended to include all the points under the ground level. Furthermore, it is known that the branches of a tree do not usually touch the ground. Therefore it is safe to assume that there exists a vertical threshold h_{branches} under which there are only ground points and trunk points. All the points under this threshold that are not part of the trunk, and all the points these points are connected to, can also be classified as ground. The points in the ground set can be discarded from further analysis.

An estimate of this vertical threshold h_{branches} does not need to be very accurate and one can use values like $h_{\text{tree}}/20$, one twentieth of the height of the

[4]Here *not connected* means that there would be a gap greater or equal to r between this cylinder and all the other cylinders.

tree.

4.6 Set direction estimation

During many stages of the tree analysis algorithm, the shape of a given set of points is often interesting. In particular, the direction in which the set is most elongated[5] in is often required for the algorithm to work. In the case where the set is a cover set, the direction estimate should present the local direction of the surface of which the cover set is a sample. In this section various methods are presented for computing estimates for the direction of a set.

The basic tool for direction estimation is principal component analysis, which can be used to find the direction which explains most of the variance in the data. Usually the data are simply the points in the given set. PCA works well on sets that are fairly elongated, but it can be imprecise in the case of sets with a higher dimensionality. The details of the principal component analysis were presented in section 2.5.3. Two alternative ways for estimating set direction are presented next. Both methods depend on the cover sets and the neighbor relation.

Average direction vector Let B_0 be a cover set with neighbor sets $B_1, \ldots, B_m$. The vectors $\mathbf{b}_0$ and $\mathbf{b}_1, \ldots, \mathbf{b}_m$ are the direction estimates received through PCA for the sets, respectively. An estimate for the direction of the set B_0 can be found by finding the optimal vector $\mathbf{d}_{\text{opt}}$ which minimizes the sum of angles between itself and the individual direction vectors. An essentially similar goal is to find $\mathbf{d}_{\text{opt}}$ such that

$$\sum_{i=0}^{m} \left(\mathbf{b}_i \cdot \mathbf{d}_{\text{opt}} \right)^2 \tag{4.28}$$

is maximized. The optimal vector can be found by applying the principal component analysis to a matrix that has the individual direction vectors as rows. The optimal vector in this case is the direction of the largest principal component.

The accuracy of the average direction approximation can be increased by using weighted sums. Larger weights can be assigned to cover sets with the most points in them. This way the effect of, for example, undetected small parts of child branches can be minimized.

Perpendicular to normals Another way of approaching the problem is to inspect the normals[6] of the individual cover sets rather than the directions. Since an accurate direction estimate is no longer required for the cover sets, this method will work also for some planar sets.

Let the sets $B_0, B_1, \ldots, B_m$ be as above and $\mathbf{n}_0, \mathbf{n}_1, \ldots, \mathbf{n}_m$ their normals, respectively. If the normals of the cover sets are assumed to be close to perpendicular to the wanted set direction, then the optimal vector, conversely to the previous,

[5] Sometimes a set is not elongated, but the direction of the set can still be estimated by other characterizations.

[6] The normal of a point cloud was defined to be the direction of the smallest principal component.

minimizes the sum of the squares of the dot products between the itself and the normals. The problem is to find $\mathbf{d}_{\text{opt}}$ so that

$$\sum_{i=0}^{m} \left(\mathbf{n}_i \cdot \mathbf{d}_{\text{opt}} \right)^2 \tag{4.29}$$

is minimized. Again, the solution can be found by applying principal component analysis to the matrix containing the individual normal vectors as rows. Since the largest principal components define the plane in which the normals lie, the smallest principal component defines the direction perpendicular to the normals.

Again, more cover sets can be added to the inspected collection, but at some point the results will only describe characteristics of the point cloud at a scale larger to the desired one.

Using the normals to estimate set direction works especially well in the case of cover sets in the trunk of the tree. In such cases the normals are likely to be perpendicular to the trunk direction. Since the surface of the trunk is curved, the directions of the normals will vary enough so that the direction estimate of the original cover set can be found.

4.7 Bifurcation analysis

During the segmentation of the components it is vital to know where a segment ends and another begins. As described in section 3.7, the forming of a segment is based on extending the current segment set one layer at a time starting from the base of the segment. The first extension layer of the segment set is called the *cut set* and the union of the few first extension layers is called the *study set*. The division of the component into these sets was visualized in figure 3.1 on page 19. As presented in the overview, when the study region becomes disconnected, further analysis is required on each of the components. The steps required for the analysis are presented here.

For simplicity, a single connected component of the study set is henceforth called simply a component. This should not be confused with the components of the tree. The following analysis is done for each of the components separately in the order of increasing cardinality measured by the number of the cover sets in them.

The analysis begins with a simple comparison between the cardinality of the component of the study set and the whole study set. If their ratio is very small, the corresponding cut set should be merged into the segment. Also, if the relative component size is large, the component is probably an extension of the segment and should be merged into it.

If the cardinality comparison of the components is inconclusive, the component is extended by a few layers to see whether it is in reality connected to other parts of the study set. This can occur, for example, due to gaps in the measurement data. If the component is connected, it is merged to the part of the study set it is connected to, and the analysis continues in another component. If the component cannot be extended the desired number of times, or it is otherwise

small in size[7], it is assumed to be too small to form a segment on its own and it is merged to the current segment.

At this point, if the component has not been merged into the segment, it is assumed to be fairly large and a start of a new branch. It is also possible that the component itself is bifurcated at its base, and the non-connected parts must be separated before creating the base for the new branch. The detection and separation of the non-connected parts is done similarly to the corresponding actions at the segmentation level. A new base is added for each connected component.

If the component is not bifurcated, the analysis continues by the estimation of the direction of the component and by its comparison to the direction of the segment. If the component is not fairly parallel to the segment then a new base is added and the extension in that direction is terminated. If the component is parallel, cylinders are fitted to the previous few layers of the segment and to the component. If the radius of the cylinder fitted to the component is relatively small compared to the radius of the other cylinder, then a new base is added. Otherwise the extension continues in that direction.

4.7.1 Base expansion

During the segmentation process there is a certain margin of error before a bifurcation is detected. The study set extends a certain amount into the part of the component that is going to be classified as a child branch. This effect causes a small stub to be left on the current segment and the base of the new branch to be a little too far. The stub can be minimized by using a process called *base expansion* during the segmentation step.[8]

Base expansion begins by expanding the base of the child segment backwards into the parent segment. The number of expansion layers is relative to the size of the base; If the base is small then the base is expanded only one or two layers. It is likely that some of the candidate cover sets found through this method are not part of the child segment and therefore some checking is required.

The cover sets whose directions do not agree with the mean direction of the cover sets in the base are filtered out. Furthermore, a circle is fitted to the original base of the segment, and the cover sets whose center points are far from the circle in the direction perpendicular to the mean direction are discarded.

[7] Size is measured with the cardinality of the component, the number of cover sets connected directly to the segment, or the absolute characteristics presented in section 4.3.2

[8] See section 5.5.2 for an additional way of fixing the gap caused by a stub.

5 Cylinder fitting

This chapter focuses on the cylinder fitting process included in the tree analysis algorithm described in chapter 3. The two different fitting methods utilized by the process are presented in section 5.1. The basic principles of the methods are presented and their properties, such as robustness, are discussed.

The fitting process[1] is explained in detail in section 5.3. The structure of the process is presented in this section. The various types of errors are also described as is error correction. Quantities to measure the size of the error are also suggested.

5.1 Fitting methods

In this section, the two cylinder fitting methods used in the cylinder fitting process and in various other parts of the tree analysis algorithm are presented. Both methods can be used for the same tasks, but usually *cylinder fitting* is preferred to *circle fitting* since the former requires less exact information.

5.1.1 Cylinder fitting

Let S be the surface of a cylinder with a radius r and the normalized axis direction $\mathbf{a}$. The closest point of the cylinder to the origin is $b\mathbf{n}$, with $\|\mathbf{n}\| = 1$. The situation is presented in figure 5.1 Now the perpendicular distance from an arbitrary point to the cylinder axis is $d(S, \mathbf{p}) + r$. This distance can also be expressed as the component of $\mathbf{p} - (b + r)\mathbf{n}$ perpendicular to direction $\mathbf{a}$. The distance from the point $\mathbf{p}$ to the surface of the cylinder can therefore be formulated as

$$d(S, \mathbf{p}) = |(\mathbf{p} - (b + r)\mathbf{n}) \times \mathbf{a}| - r. \tag{5.1}$$

The least-squares minimization problem would then be to minimize

$$\sum_{\mathbf{p}_i \in P} (d(S, \mathbf{p_i}))^2, \tag{5.2}$$

when the set P forms the point cloud of measurements.

Fitting a cylinder to a given point cloud is a non-linear optimization problem as can be seen from equation (5.1). Such a problem can be solved using an iterative non-linear techniques such as the Levenberg-Marquardt method [14] or the Gauss-Newton method[2]. However, due to the nonlinearity, the methods require good initial values to converge to the correct solution. The required initial values were presented in section 3.9 and they are the radius r_0, axis point $\mathbf{z}_0$ and axis direction $\mathbf{a}_0$.

[1] The overview of the fitting process was presented in section 3.9.

[2] The Gauss-Newton method has been used in the MATLAB® implementation of the algorithm.

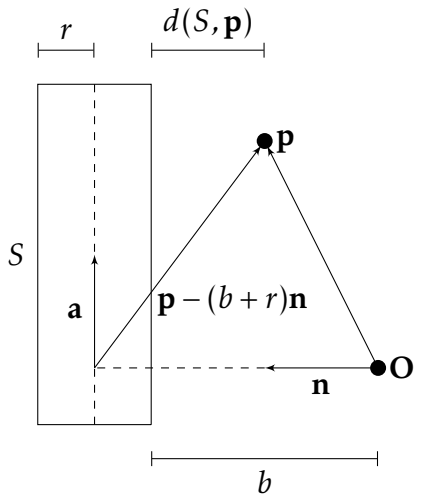


Figure 5.1: Distance between an arbitrary point $\mathbf{p}$ and the surface S of a cylinder. The distance $d(S, \mathbf{p})$ can be expressed as the difference between the component of the vector $\mathbf{p} - (b + r)\mathbf{n}$ perpendicular to the axis $\mathbf{a}$ and the radius r. It should be noted that the vector $\mathbf{p} - (b + r)\mathbf{n}$ is not necessarily on the plane defined by the vectors $\mathbf{n}$ and $\mathbf{a}$.

5.1.2 Circle fitting with a known axis direction

Given a set of measurements, if the axis direction of the cylinder is known, the cylinder fitting problem can be reduced to a 2-dimensional circle fitting problem. In this case the measurements are projected on a plane that is perpendicular to the axis direction $\mathbf{a}$ and a circle is fitted to the received 2-dimensional data points in a least-squares sense.

Let the matrix P contain the 3-dimensional points as row vectors and the direction vector $\mathbf{a}$ of the cylinder has the components a_1, a_2, a_3. To reduce the dimensionality of the point cloud, a plane perpendicular to the axis direction needs to be defined. This is done by generating two unit vectors $\mathbf{a}_1^\perp$ and $\mathbf{a}_2^\perp$ that are perpendicular to the axis direction:

$$\mathbf{a}_1^\perp = \begin{cases} [1,0,0] & a_1 = 0 \quad \wedge \quad a_2 = 0 \\ [a_2, -a_1, 0] & \text{otherwise} \end{cases} \tag{5.3}$$

$$\mathbf{a}_2^\perp = \mathbf{a} \times \mathbf{a}_1^\perp, \tag{5.4}$$

The 3-dimensional point cloud with the matrix presentation P is projected onto the perpendicular plane by the following matrix multiplication.

$$\tilde{P} = PA, \tag{5.5}$$

where $\tilde{P}$ is an $N \times 2$-matrix that contains the projected points, A is a $3x2$-matrix with the columns $\mathbf{a}_1^\perp$ and $\mathbf{a}_2^\perp$, and N is the number of points in the point cloud.

The optimization problem is to find the center point $\mathbf{c}$ and the radius r of the circle so that the following sum is minimized:

$$\sum_i (d(S, \tilde{\mathbf{p}}_i))^2, \tag{5.6}$$

where S is the circumference of the circle. The distance can be written as a function of the center point and the radius as

$$d(S, \tilde{\mathbf{p}}_i) = |\tilde{\mathbf{p}}_i - \mathbf{c}| - r. \tag{5.7}$$

The radius of the optimal circle is then the radius of the resulting cylinder. The point on the axis of that cylinder can be found the following way.

Theorem 3. *Let P be a three-dimensional point cloud and $\tilde{P}$ the resulting point cloud when P is projected onto a plane. Let the plane point $\mathbf{c}$ be the center point of the optimal circle fitted to the point cloud $\tilde{P}$. For any point $\tilde{\mathbf{p}} = \mathbf{p}A \in \tilde{P}$, if the difference between the points $\tilde{\mathbf{p}}$ and the center point $\mathbf{c}$ is the 2×1-vector*

$$\mathbf{b} = \tilde{\mathbf{p}} - \mathbf{c}, \tag{5.8}$$

then the axis of the optimal cylinder must go though the point

$$\mathbf{p}_{axis} = \mathbf{p} + A\mathbf{b}. \tag{5.9}$$

Proof. To prove the theorem one needs to show that the distance between an arbitrary point $\mathbf{p} \in P$ and the axis line $\mathbf{p}_{\text{axis}} + n\mathbf{a}$ is equal to the distance from the corresponding point $\tilde{\mathbf{p}}$ in the projected set to the center point of the circle.

The distance from the point $\mathbf{p}$ to the line L defined by the point $\mathbf{p}_{\text{axis}}$ and the direction $\mathbf{a}$ is

$$d(L, \mathbf{p}) = \frac{\|\mathbf{a} \times (\mathbf{p}_{\text{axis}} - \mathbf{p})\|}{\|\mathbf{a}\|}. \tag{5.10}$$

The norm of the direction vector $\mathbf{a}$ is one. Additionally, by using equation (5.9) we receive

$$\begin{aligned} d(L, \mathbf{p}) &= \|\mathbf{a} \times (\mathbf{p} + A\mathbf{b} - \mathbf{p})\| && (5.11)\\ &= \|\mathbf{a} \times A\mathbf{b}\| && (5.12)\\ &= \|[\mathbf{a} \times \mathbf{a}_1^{\perp} \quad \mathbf{a} \times \mathbf{a}_2^{\perp}]\mathbf{b}\| && (5.13)\\ &= \|[\mathbf{a}_2^{\perp} \quad \mathbf{a}_1^{\perp}]\| \cdot \|\mathbf{b}\| && (5.14) \end{aligned}$$

Because the vectors $\mathbf{a}$, $\mathbf{a}_1^{\perp}$ and $\mathbf{a}_2^{\perp}$ are orthonormal the norm of the above matrix is equal to one:

$$d(L, \mathbf{p}) = \|\mathbf{b}\|, \tag{5.15}$$

which is the same as the distance from the corresponding projected point to the center point of the circle as defined in equation (5.8). □

5.2 Cylinder length and starting point

Neither of the methods presented above give information about the length of the cylinder, and therefore an estimate of the length of the cylinder needs to be computed manually. Also, since the precise location of the cylinder is needed, the point on the axis needs to be updated to be the starting point of the cylinder.

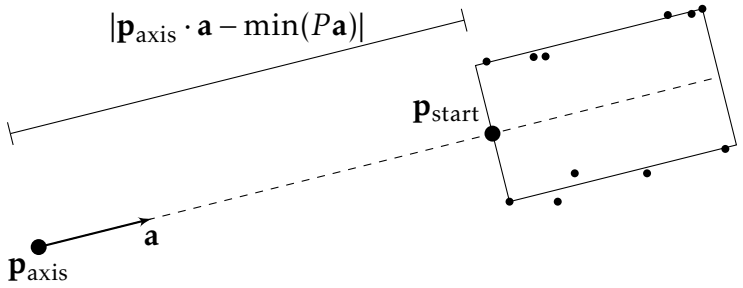


Figure 5.2: Computing the starting point. The starting point $\mathbf{p}_{\text{start}}$ can be found by computing the distance between the point on the axis $\mathbf{p}_{\text{axis}}$ and the minimum value of the projections of the points in set P onto the axis $\mathbf{a}$. The axis is scaled by the result and added to the point on the axis. The smaller points present the point cloud P.

5.2.1 Length estimate

Let r, $\mathbf{a}$, and $\mathbf{p}_{\text{axis}}$ be the radius, axis direction and the point on the axis given by a fitting method, respectively. An estimate of the length h of the cylinder can be found by projecting all the data points P on to the axis of the cylinder and by taking the difference of the maximum and minimum values of the projections.

$$h = \max(P\mathbf{a}) - \min(P\mathbf{a}). \tag{5.16}$$

This length estimate is not very robust. In theory a single outlier can distort the estimate by any amount without an upper bound. However, due to the region-forming process, the distance between the points has an upper bound, which means that the maximum distance between an outlier and the rest of the set is less than the radius of the cover sets.

In practice, the given length estimate has proven to be sufficient. However, a more detailed error analysis for the length estimate is conducted in section 5.2.3.

5.2.2 Computing the start point

If the above estimate for the length is used, the cylinder starting point can be defined to be the point on the axis corresponding to the minimum projection value in (5.16). This starting point $\mathbf{p}_{\text{start}}$ can be found by computing

$$\mathbf{p}_{\text{start}} = \mathbf{p}_{\text{axis}} - (\mathbf{p}_{\text{axis}} \cdot \mathbf{a} - \min(P\mathbf{a})) \cdot \mathbf{a}. \tag{5.17}$$

The process is illustrated in figure 5.2.

When all the quantities - radius r, length h, axis direction $\mathbf{a}$ and starting point $\mathbf{p}_{\text{start}}$ - are known, a cylinder is fully defined. The same parameters are also the values returned by the cylinder fitting methods.

5.2.3 Error estimation

The starting point does not cause error in the cylinder length but the axis direction can. If the axis direction $\mathbf{a}'$ is off by an angle α, the error estimate for the length can be computed in the following way. See figure 5.3.

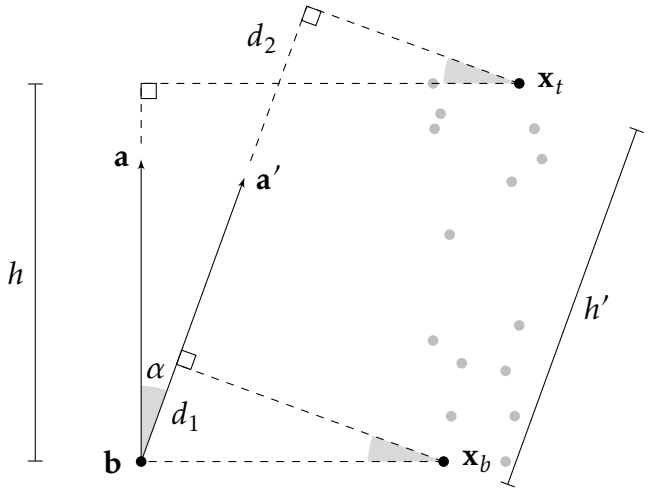


Figure 5.3: The effect of the axis direction on the length of a cylinder. When the axis direction $\mathbf{a}$ changes by the angle α to the direction $\mathbf{a}'$ the length changes from h to h'. The situation is viewed on a plane defined by the two direction vectors $\mathbf{a}$ and $\mathbf{a}'$. The point $\mathbf{x}_b$ is the point that gives the lowest projection value and the point $\mathbf{x}_t$ gives the highest. All the highlighted angles are equal to α.

Let the vector $\mathbf{a}$ be the correct direction and h the corresponding length. The length h' corresponding to the direction $\mathbf{a}'$ is dependent on the angle between the direction vectors and the shape of the point cloud. Also, certain extreme points need to be selected from the point cloud. The point $\mathbf{x}_b$ is the point that gives the lowest projection value when projected onto the vector $\mathbf{a}$, and the point $\mathbf{x}_t$ gives the highest such value.

We assume that the points that give the extreme projections are always the same independent of the distortion angle α. This is certainly not always the case. If the points in the point cloud are evenly distributed and there are enough of them, then there should be multiple points at the top of the point cloud (and similarly at the bottom) that give approximately the same extreme projection value. This means that the points $\mathbf{x}_b$ and $\mathbf{x}_t$ can be chosen in multiple ways. In order to keep the extreme points constant when the axis direction changes, the point $\mathbf{x}_b$ should be chosen as close to a reference point $\mathbf{b}$ as possible and the point $\mathbf{x}_t$ as far as possible, as is done in figure 5.3.

The distorted length h' can now be expressed with the help of h:

$$h' = \frac{h}{\cos\alpha} - d_1 + d_2, \tag{5.18}$$

where the variables d_1 and d_2 are

$$d_1 = d_{A,\mathbf{a}}(\mathbf{x}_b, \mathbf{b}) \sin\alpha, \tag{5.19}$$

$$d_2 = d_{A,\mathbf{a}}(\mathbf{x}_t, \mathbf{b}) \sin\alpha - h \sin\alpha \tan\alpha. \tag{5.20}$$

The distance $d_{A,\mathbf{a}}(\mathbf{x}, \mathbf{y})$ means that the points are first projected on the plane A, which in this case is spanned by the vectors $\mathbf{a}$ and $\mathbf{a}'$, and then on the plane in which the distance is measured in the direction perpendicular to $\mathbf{a}$.

By combining the previous equations h' takes the form

$$h' = h\frac{1}{\cos\alpha}(1-\sin^2\alpha) + (d_{A,\mathbf{a}}(\mathbf{x}_t,\mathbf{b}) - d_{A,\mathbf{a}}(\mathbf{x}_b,\mathbf{b}))\sin\alpha. \tag{5.21}$$

The difference between the two distances is equal to the distance between the points $\mathbf{x}_b$ and $\mathbf{x}_t$ in the same direction:

$$h' = h\frac{1}{\cos\alpha}(1-\sin^2\alpha) + d_{A,\mathbf{a}}(\mathbf{x}_b,\mathbf{x}_t)\sin\alpha \tag{5.22}$$

$$= h\cos\alpha + d_{A,\mathbf{a}}(\mathbf{x}_b,\mathbf{x}_t)\sin\alpha. \tag{5.23}$$

The distorted length is clearly dependent on the angle of distortion and the shape of the set on the plane formed by the two directions, i.e., the length h in the direction of the vector $\mathbf{a}$ and the coefficient $d_{A,\mathbf{a}}(\mathbf{x}_b,\mathbf{x}_t)$, which is a little more ambiguous. If the point cloud has a clear rectangular shape on the study plane, then the term $d_{A,\mathbf{a}}(\mathbf{x}_b,\mathbf{x}_t)$ can be seen as the *width* of the set perpendicular to the direction $\mathbf{a}$. In most cases the coefficient has no clear geometric meaning since it can even be zero for some sets.

$\Delta h = |h' - h|$ can now be formulated:

$$\Delta h = \left|h(\cos\alpha - 1) + d_{A,\mathbf{a}}(\mathbf{x}_b,\mathbf{x}_t)\sin\alpha\right|. \tag{5.24}$$

The magnitude of the error is dependent on the fine structure of the point cloud. Therefore exact error limits are impossible to define but multiple examples are given below.

Example 7. *Let the point cloud have a clear rectangular shape as discussed above. If the correct length of the cylinder is h and the width is ch for some constant c, the relative error is*

$$\frac{\Delta h}{h} = |\cos\alpha - 1 + c\sin\alpha|. \tag{5.25}$$

The values of the relative error are visualized for several values of the constant c as the function of the distortion angle α in figure 5.4. The results show that when the distortion angle is small ($\alpha < \frac{\pi}{4}$) the error stays relatively small in the case of elongated[3] *cylinders. However, in the case of wide cylinders, the error can be close to 50% even with small angles.*

The results show that the shape of the cylinder fitting region can affect the fitting accuracy greatly. Therefore the fitting regions should be elongated.[4] The length error can be kept small by altering the region forming process to require the formed regions to have, for example, a length to width ratio close to 3 : 1.

Example 8. *In the previous example the shape of the point cloud was assumed to be rectangular. If this is not the case then the term $d_{A,\mathbf{a}}(\mathbf{x}_b,\mathbf{x}_t)$ is not an estimation of the width of the point cloud. In fact the distance of the two extreme points perpendicular to the axis direction can be zero. This would mean that the second term in the length*

[3] Cylinders that are longer than they are wide.

[4] The elongated shape can also make the direction estimation more accurate.

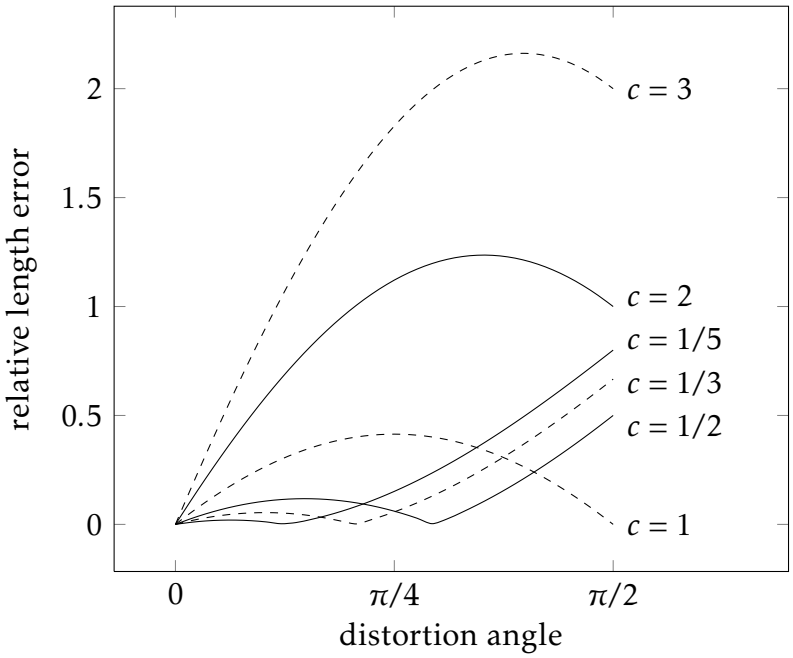


Figure 5.4: Relative error of the length of a cylinder as a function of the distortion angle. The constant c is the ratio between the length and the width of the fitting region.

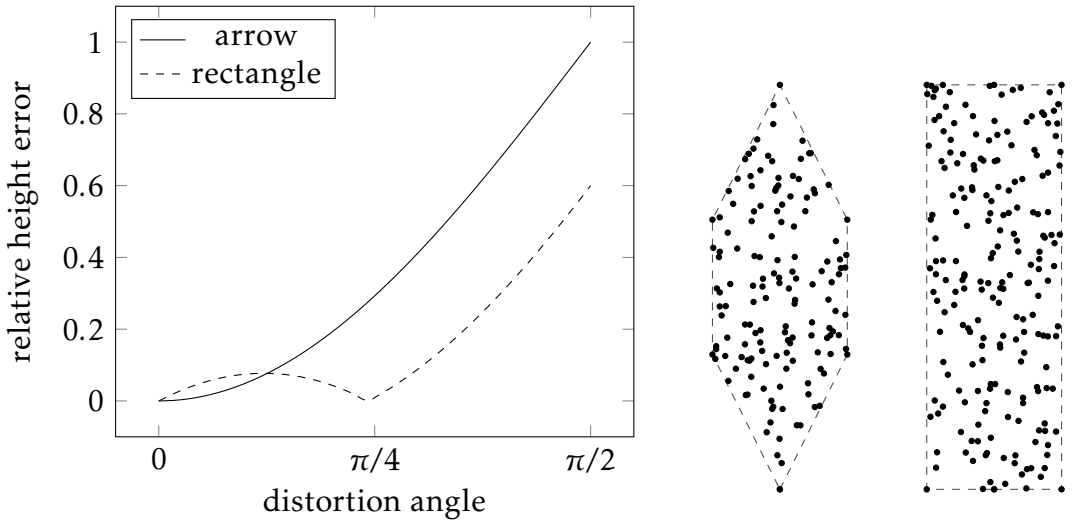
error formula (5.24) *vanishes and the error only depends on the length of the set and the distortion angle.*

Figure 5.5 shows how the second term of the error formula affects the error. Figure 5.5a shows two error plots, one of which corresponds to a rectangular point cloud with the length–width ratio 5 : 2*, and the other corresponding to a point cloud with arrow-shaped heads meaning that the* width *equals zero. The different point cloud shapes are visualized in figures 5.5b and 5.5c. It should be noted that if not all of the points in the point cloud lie on the same line, the error estimate is not accurate after a certain angle. The figure shows that in the case of small angles the error is smaller in the case of arrow-shaped point clouds. In the case of larger angles the analysis is harder since the extreme point will change away from the arrow tip.*

5.3 Cylinder fitting process

The fitting is a subroutine of the tree analysis algorithm that receives the point cloud and the initial values as inputs, and returns (if possible) the parameters for an optimal cylinder with the given inputs. In addition to the initial values for the radius, axis direction, and axis point, the process also receives information about the possible successful cylinders fitted on to the same segment. This information is used for segment level error checking. Similarly, information about the size of the whole tree is passed to the process for tree level error checking.

The outline of the cylinder fitting process is presented in figure 5.6 as a simple flowchart.



(a) Height error **(b)** Arrow **(c)** Rectangle

Figure 5.5: Comparison of the length error of point clouds with different shapes. It should be noted that the error estimates are accurate only as long as the extreme projections are given by the assumed points: tips of the arrow and the corners of the rectangle.

Finding the axis direction The fitting process begins by applying the method seen in section 5.1.1 for a given number of times. After each try, given that the solution converged, the axis direction of the proposed cylinder is checked for errors. First the direction of the cylinder is compared to the direction estimate of the segment. If the angle between them is more than ninety degrees, the direction of the cylinder is flipped. This means that the starting point is moved in the direction of the axis with the magnitude of the length of the cylinder, and the axis direction is reversed. So the new axis direction $\tilde{\mathbf{a}}$ and the new starting point $\tilde{\mathbf{p}}$ become

$$\tilde{\mathbf{a}} = -\mathbf{a} \tag{5.26}$$

$$\tilde{\mathbf{p}} = \mathbf{p} + h\mathbf{a}. \tag{5.27}$$

This ensures that all the cylinders in a segment are oriented the same way and away from the base of the segment.

Next, the axis direction of the proposed cylinder is compared to the estimated direction and the direction of the possible previous cylinder fitted to the same segment. If the angle between either of these pairs of vectors is too large, the fitting attempt is classified as a failure. In this case the distance between the surface of the proposed cylinder and each individual point is inspected. The points that are farthest from the surface are discarded. A new estimate for the axis direction is received by doing a principal component analysis for the remaining point cloud. The axis point is also updated to be the mean point of that data set. If attempts remain, a new fitting is tried with the updated point cloud.

If the axis direction does not vary too much from the expected direction or the direction of the previous cylinder, the axis direction is accepted. The process continues by inspecting the correctness of the axis point and the radius.

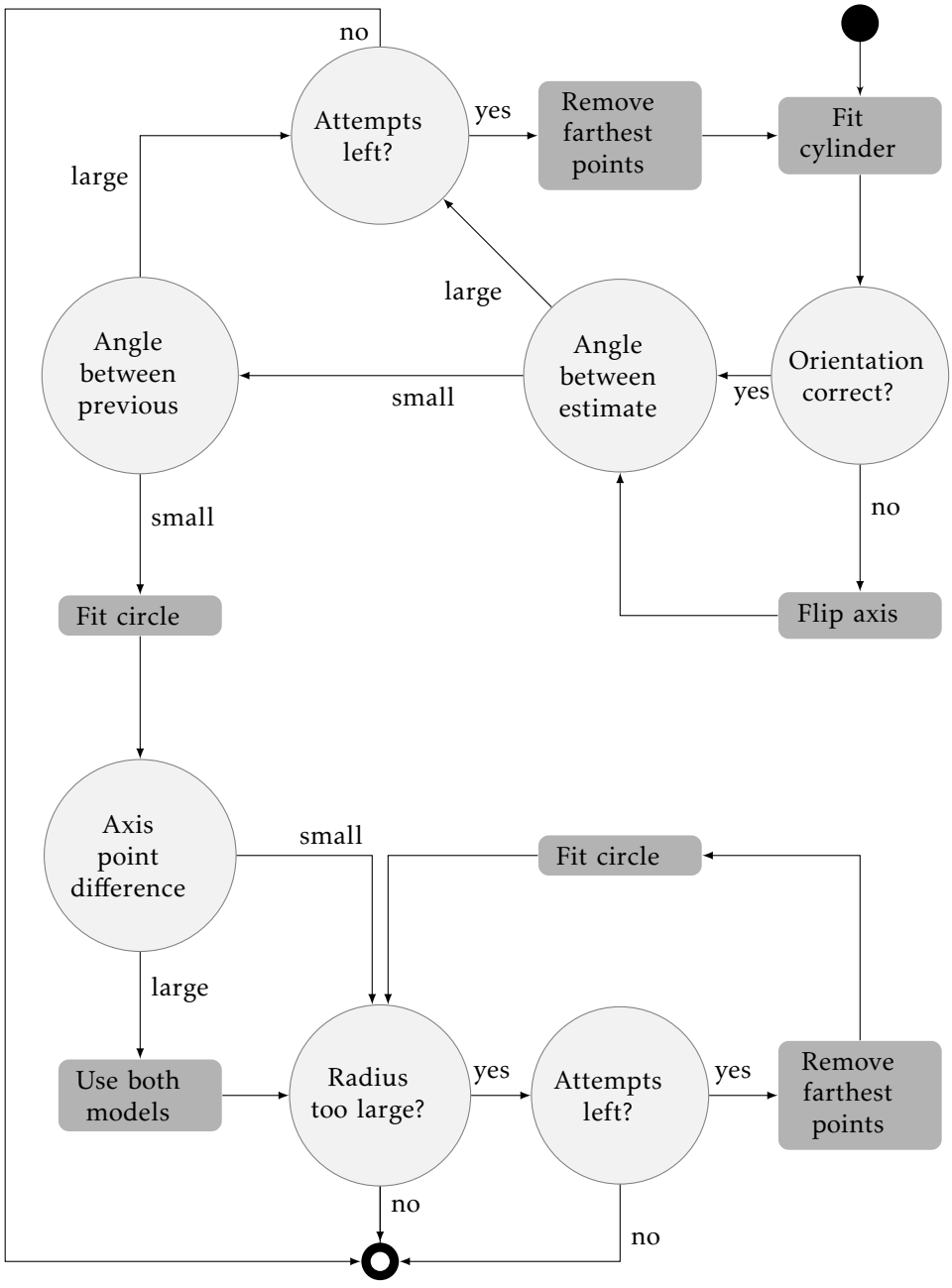


Figure 5.6: Cylinder fitting process. The steps of the cylinder fitting process shown as a flowchart. The rectangles are the stages and the circles are the decisions required by the process. The solid circle is the starting point for the algorithm and the hollow circle the ending point of the process.

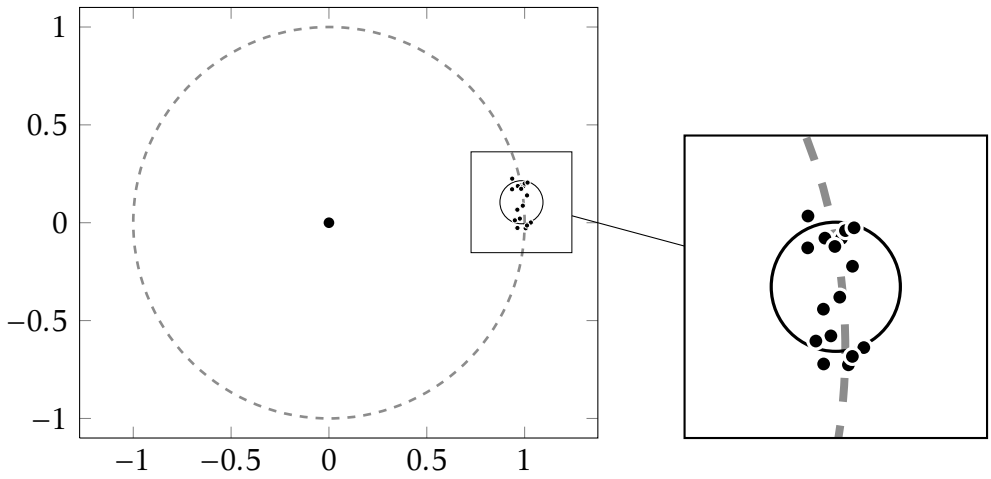


Figure 5.7: Two possible interpretations for the least-squares optimal solution viewed in the axis direction. The point cloud is either an incomplete sample of a larger cylinder (dashed), or a complete sample of a smaller cylinder (solid) but the measurements contain much noise. The difference between the axis points of the two possible cases is large.

Checking the axis point In most cases, if the axis direction is found correctly, the axis point is also close to correct. However, if, e.g., the point cloud is incomplete (see figure 5.7), there is a possibility for a large error in the computed axis point. In such a case, the cylinder fitting method might miss the fact that the measurements are in fact an incomplete sample of a surface with a large radius, rather than a complete sample of a surface with a small radius, or vice versa.

In both cases, the variance of the distance between the points and the cylinder surface can be equally small. The ratio between the distance variance and the radius of the proposed cylinder is very different in both cases. However not even this ratio can be used to choose the correct interpretation.

The process tries to detect the problem by using the circle fitting method for the point cloud and the axis of the accepted cylinder. If the axis point given by the circle fitting method varies greatly from the corresponding value of the cylinder fitting method, the described problem may exist. In this case, the cylinder suggested by the circle fitting method is accepted as a competitive result. The two alternatives are further inspected side by side.

Naturally, if the circle fitting method produces a result that is close to the cylinder fitting one, only one of them, the cylinder fitting alternative, is accepted for further analysis.

Testing the radius As in the case of the axis direction, during this step a limited number of fitting attempts are possibly made. However, the radii of the solutions accepted in the previous steps are first checked for errors. If no errors are detected, a solution is accepted as the final result of the fitting process. Otherwise a new fitting is attempted by using the circle fitting method. If the cylinder fitting method was successful earlier, the resulting axis direction is used here, otherwise the original estimate is recalled. The fitting result at each step is checked for the radius errors described below. If the result has errors, the points farthest from

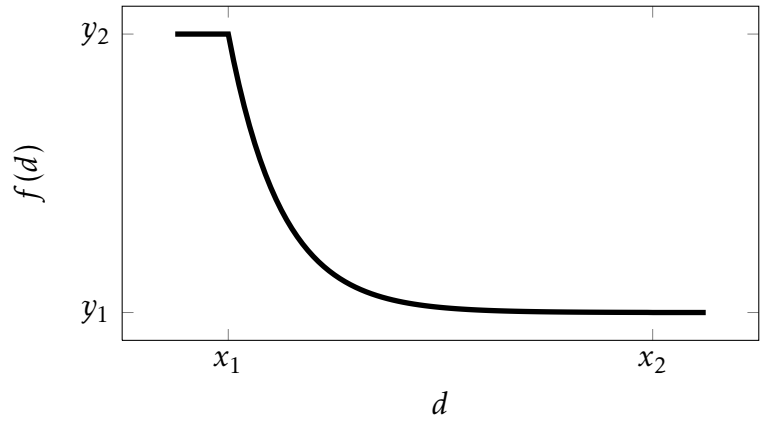


Figure 5.8: Plot of a radius scaling function $f(d)$ presented in equation (5.29). The values $x_1 = 0.1$, $x_2 = 0.9$, $y_1 = 0.2$, $y_2 = 1$ and $a = 10$ were used for the figure.

the circumference are removed before starting a new step.

At each step, the radius of the proposed fitting result is checked for errors. The following quantities are used if they are available.

Maximum tree radius $r_{\text{tree}}(h, d)$ is dependent on the location of the mean point of the region relative to the size of the tree, more precisely the length h measured from the ground level and the distance d from that point to the mean of the trunk measured perpendicular to the trunk direction. The maximum tree radius is computed relative to the largest radius r_{trunk} of the first segment, which is assumed to be the beginning of the trunk:

$$r_{\text{tree}}(h, d) = r_{\text{trunk}} f(d) g(h)\,, \tag{5.28}$$

where $f(\cdot), g(\cdot) : \mathbb{R}^+ \to (0, 1]$ are functions that produce proper coefficients for scaling the radius. Such functions are, for example,

$$f(d) = \begin{cases} y_2 & d \le x_1 \\ y_1 & d \ge x_2 \\ (y_2 - y_1) e^{-a\left(\frac{d}{r_{\text{crown}}} - x_1\right)} + y_1 & \text{otherwise} \end{cases}, \tag{5.29}$$

$$g(h) = z_2 - (z_2 - z_1)\frac{h}{h_{\text{tree}}}\,, \tag{5.30}$$

where r_{crown} is an estimate of the radius of the crown of the tree, h_{tree} is an estimate of the tree height and x_1, x_2, y_1, y_2, z_1, z_2 and a are suitably selected parameters. These functions are visualized in figure 5.8.

Maximum segment radius estimate r_{seg} is computed for each segment based on the radii of the cylinders fitted to the regions during the initial stage of the region forming (see section 3.8). If the variance of these radii is large, the minimum radius is used for the limit. Otherwise the maximum is used.

The third quantity that is used for the radius error detection is the maximum radius r_{prev} of the cylinders previously fitted to the segment.

The defined radius quantities are used to inspect the radius candidate. If any

of the following conditions are true the circle fitting is classified as a failure:

$$r > (1 + p_1) r_{\text{tree}}(h, d)\,, \tag{5.31}$$

$$r > (1 + p_2) r_{\text{seg}}\,, \tag{5.32}$$

$$r > (1 + p_3) r_{\text{prev}} \quad \wedge \quad \frac{r_{\text{prev}}}{r_{\text{seg}}} > p_4\,, \tag{5.33}$$

where $p_i, i \in \{1, 2, 3, 4\}$ are suitably selected percentages which define thresholds. The first two conditions are fairly self-explanatory. In the third the latter half ensures that the previous cylinder is not very small, which could happen, for example, when a cylinder was last fitted to a small extension. The first half simply states that the new radius cannot be too much larger than the previous radius.

Output of the process If both the axis direction and radius testing steps of the cylinder fitting process were successful, then the whole process was also successful. The solution that was the first (and last) to pass all the conditions (5.31) is chosen as the output. The cylinder parameters - radius, length, axis direction and axis point - of the successful fitting are returned. The success status is also returned.

5.4 Fitting error detection

During the cylinder fitting process described above, the fitting was usually done as a least-squares optimization problem. However, the fitting result can be *wrong* in many ways even though it is least-squares optimal. Such fitting errors are caused, e.g., due to prior problems in the segment or the region forming processes. Also gaps in the measurements can cause unwanted results during the fitting process. The types of errors that can be present during each fitting attempt are discussed next.

5.4.1 Types of errors

In this section, multiple events that could indicate an incorrect fitting result are presented. It should, however, be noted that none of the presented indicators work in every case. Sometimes the data simply do not contain enough information for a *correct* fitting to be found.

Some of the indicators listed are part of the cylinder fitting process described earlier, but all of them were considered during the development.

Difference between computed angle and the estimate If the angle between the estimated axis direction and the axis direction produced by the fitting method is fairly large, then at least one of them is wrong. The problem is, of course, which one. The logical first step would be to recompute the direction estimate using a different method[5]. If the estimates are different, a new estimate must be chosen according to some heuristic.

[5]Set direction estimation methods were discussed in section 4.6

On the other hand, if the second estimate is close to the first one, then the problem is probably in the result produced by the fitting process. Such problems are usually caused by problems in the segment and region forming stages. For example, if the region in question includes small pieces of the beginnings of child branches, the computed axis is likely to be tilted towards the extra extension(s) (see figure 5.9). In this case, the magnitude of the unwanted axis tilt is relative to the size of the extra extension as compared to the size of the rest of the region.

The cure for the axis tilt is to remove[6] the farthest points from the data set and try again. Now, if the extra extension is small in size, its tip is likely among the farthest points and it is therefore removed. The axis direction will turn towards the correct one each time the process is repeated.

In the case where the size of the extension is similar to the size of the rest of the region, the effects of data elimination are more unpredictable. It is hard to know which points are eliminated, and whether the elimination will make the direction estimate better. An example can be seen in figure 5.10. The best approach would be to divide such a region in two smaller parts and try fitting cylinders separately into them. This is, however, difficult because there are no clear indicators for such a case.

Difference between new and previous angles The cylinders fitted to a single segment are expected to have a similar direction since the segment should not contain sharp turns. Therefore the angle between two consecutive cylinder axis directions can be used as an error indicator. If the angle is large, the last computed axis direction can be assumed to be wrong.

The maximum allowed angle between the consecutive directions should not be too small since the directions in a segment can have quite a large variance depending, e.g., on the measurement density. Furthermore, the axis directions of the fitted cylinders can be altered during the post-processing stage which corrects the possible errors made during the fitting process.

Large radius The radius of the fitted cylinder can sometimes be an obvious indicator for errors in the fitting process. As presented in section 5.3, several upper bounds for the radius a cylinder can be computed. If the radius of the fitting result is larger than the limits, something has probably gone wrong.

Sometimes, too large a radius is caused by badly formed regions. This can happen, for example, if the region contains an extension, which can be the stub of a smaller branch. The least-squares solution is correct but the radius can still be greater than some computed limit. Another possibility is that the point cloud scale is misinterpreted to be part of an incomplete large circle rather than a complete small circle, as was shown in figure 5.7

Variance of point distance As discussed above, the goal of the fitting methods is to find a solution that minimizes the sum of the distances between the points and the cylinder surface. Once the cylinder is defined, the individual distances are also easily available, and the variance of those distances can be computed.

[6] An alternative would be to decrease the weight of the farthest points, if the fitting method uses weighted sums.

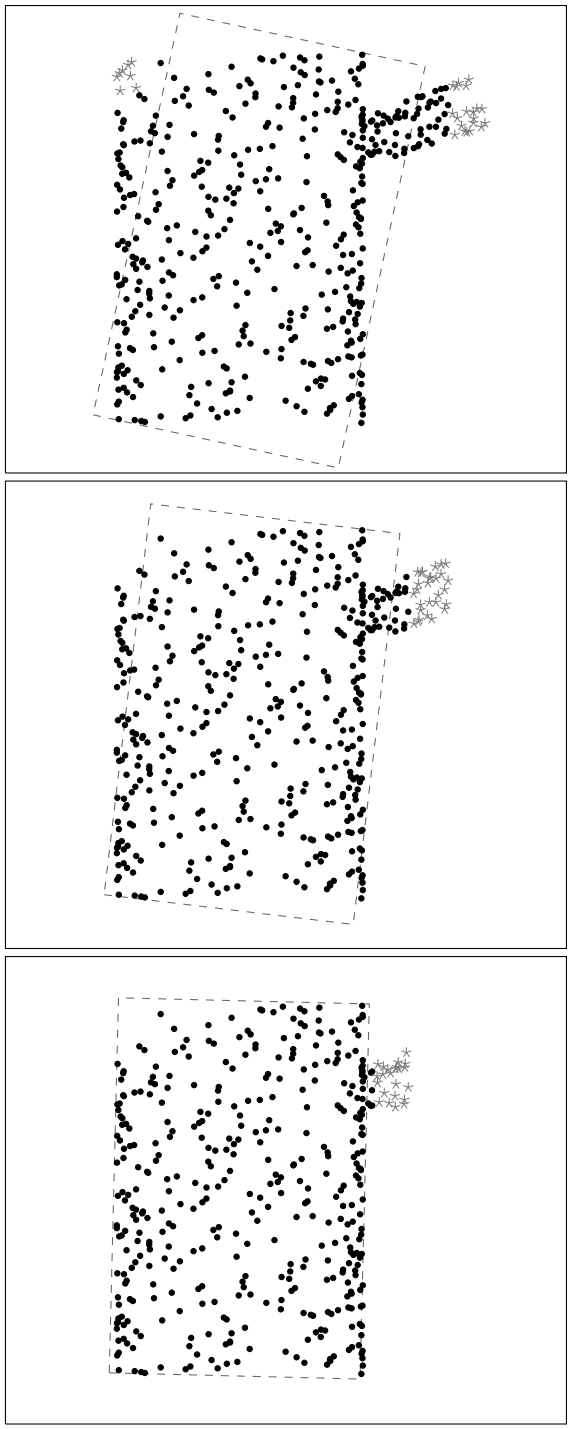


Figure 5.9: Example of how a small extension affects the fitting process. The rectangle shows the least-squares optimal cylinder fitted to the point cloud. The points marked by stars are the points farthest from the surface of the cylinder and are therefore removed before the next fitting attempt.

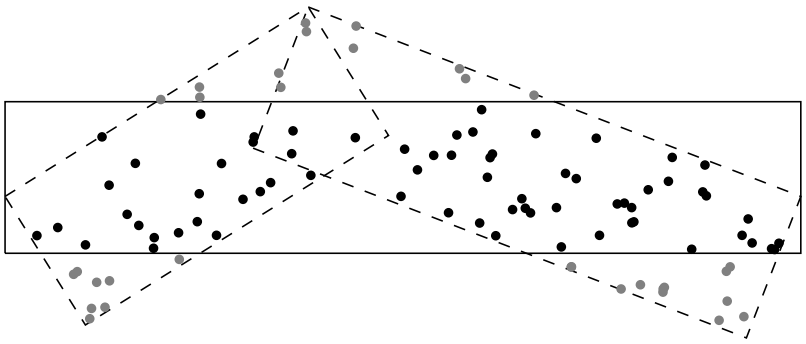


Figure 5.10: Example of the effects of bad region forming on the fitting process. The parts, with similar sizes, presented by the dashed rectangles have been assigned to a single region. The solid rectangle shows how the cylinder is likely to be fitted. Removing the farthest points, shown in gray, is not likely to improve the fitting result.

If the variance is small, the fit is probably good since all the points are relatively close to the surface. On the other hand, if the variance is large, it means that the point cloud included measurements that did not fit the proposed cylinder. There are two main scenarios why this can happen. The first is that the measurements are simply noisy. The cylinder is in the correct position but the noise in the measurements makes the variance large. This can make the distance variance an inaccurate error indicator.

The other possibility is that the computed axis direction is off. In such a case some of the measurements are likely to fit the surface well, but others can be very far away. This can happen, for example, in the case shown in figure 5.9 where the extension in the region caused the axis to tilt. In the figure it is clear that the variance of the distances becomes smaller as the axis moves towards the correct direction.

Points inside cylinder The *signed* distances between the points and the cylinder surface should be positive in most cases when the fitting has been successful. This is due to the fact that the measurements are unlikely to reflect from inside the tree surface. Of course, in the case of small branches, e.g., wind conditions can cause exceptions to this rule. Another possibility is that the branch is not locally cylindrical. These exceptions make it hard to define limits for the allowed number of points inside the cylinder.

If a large percentage of the points in the region lie inside the proposed cylinder, the fitting is probably incorrect. There should naturally be a tolerance interval since even in successful cases the points can be a little inside the surface. Again, in the case presented in figure 5.9, the number of points inside the cylinder decreases when the cylinder turns towards the correct direction.

5.5 Cylinder model post-processing

As the result of the cylinder fitting process described in section 5.1.1, completed for each individual region, a complete cylinder model has been formed. However, the algorithm does not stop there, since the cylinder model can be checked for errors and the cylinders can be modified in the post-processing stage.

It should be noted that the point cloud is no longer required since all the analysis is carried out with only the cylinder information, i.e., geometric properties and topological relations.

The post-processing stage is divided into two separate parts. First the cylinders in each segment are checked for errors and modified if needed. Then, in the second part, the algorithm finds gaps between segments and adds additional cylinders into them. As a result, the cylinder model becomes more *continuous*. The two parts of the post-processing stage are presented in sections 5.5.1 and 5.5.2 in the order they are used in the algorithm.

5.5.1 Filling gaps in a segment

Based on the segment forming process, a segment should present a part of the scanned tree that is similar to a branch. This means that the cylinders fitted to the segment should also form a whole that is *continuous*, in the sense that there are no large gaps between the individual cylinders, and *smooth*, in the sense that neither the radius nor the axis direction should change too much between consecutive cylinders.

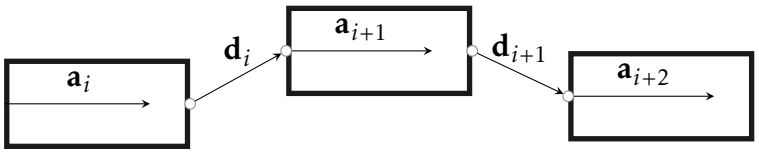
The previous requirements can be formulated into mathematical conditions that can be checked in each cylinder set corresponding to a segment. The correctness of cylinders can be checked in the following way.

Let C_i, C_{i+1} and C_{i+2} be three consecutive cylinders in a segment with axis directions $\mathbf{a}_i$, $\mathbf{a}_{i+1}$ and $\mathbf{a}_{i+2}$, respectively. Also, let the vector $\mathbf{d}_i$ be the difference from the end point of the cylinder C_i to the starting point of the cylinder C_{i+1}, and $\mathbf{d}_{i+1}$ a similar vector between the cylinders C_{i+1} and C_{i+2}.

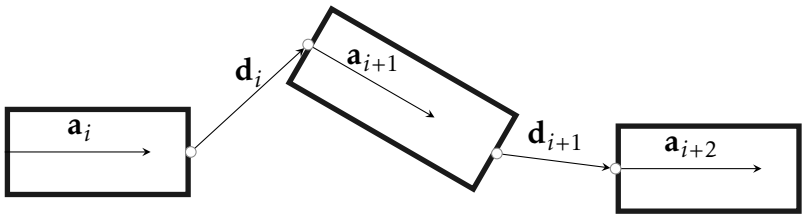
The lengths $\|\mathbf{d}_i\|$ and $\|\mathbf{d}_{i+1}\|$ can be used to indicate gaps in the segment. In the ideal case, the end point of a cylinder is the starting point of the next cylinder in the segment, and the distance between the points is zero. Additionally, the angles between the distance vectors and the direction vectors can be used to analyse the location of the cylinders more closely.

Next I discuss a couple of common examples of situations where gap filling is required. In the examples it is assumed that the axis directions $\mathbf{a}_i$ and $\mathbf{a}_{i+2}$ are close to each other and that the corresponding cylinders C_i and C_{i+2} are close to aligned.

Correct direction Consider a case where the radius of a cylinder in the middle of a segment has grown large due to a stub of a bifurcated branch. The distorted radius is easy to correct by averaging the radii of the previous and the next cylinders if they have radii with similar sizes. However, when the radius of the middle cylinder is altered, the axis point remains the same. The cylinder looks like it has been moved away from the correct alignment in the direction perpendicular to its axis direction (see figure 5.11a).



(a) Similar axis direction



(b) Different axis direction

Figure 5.11: Gap filling situations inside a segment. Two complicated cases where cylinder properties other than the length, need to be altered.

This type of situation where the axis direction is correct but the alignment is not is characterized by the following conditions:

$$\sphericalangle(\mathbf{a}_i, \mathbf{a}_{i+1}) \quad \wedge \quad \sphericalangle(\mathbf{a}_{i+1}, \mathbf{a}_{i+2}) \quad \text{are small,} \tag{5.34a}$$

$$\sphericalangle(\mathbf{a}_i, \mathbf{d}_i) \quad \wedge \quad \sphericalangle(\mathbf{a}_{i+2}, \mathbf{d}_{i+1}) \quad \text{are large.} \tag{5.34b}$$

The first condition ensures that the cylinders in fact have a similar axis direction. The second states that the end point of the first cylinder is not aligned with the start point of the second cylinder, and the same for the second and the third cylinder.

To fix the alignment and the gaps between the cylinders, the length of the cylinder C_i is increased by the magnitude of the distance $\mathbf{d}_i$ in the direction of $\mathbf{a}_i$. Next, the starting point of the cylinder C_{i+1} is moved to the new end point of the first cylinder C_i. The length of the middle cylinder is recomputed so that the cylinder reaches the start of the third cylinder.

Wrong direction Another possible situation is that the axis direction of the middle cylinder does not agree with the directions of the other two cylinders (see figure 5.11b). This can be caused when an extension distorts the direction during the fitting process and the error is not caught:

The situation is characterized as follows.

$$\sphericalangle(\mathbf{a}_i, \mathbf{a}_{i+1}) \quad \wedge \quad \sphericalangle(\mathbf{a}_{i+1}, \mathbf{a}_{i+2}) \quad \text{are large,} \tag{5.35a}$$

$$\sphericalangle(\mathbf{a}_i, \mathbf{d}_i) \quad \vee \quad \sphericalangle(\mathbf{a}_{i+2}, \mathbf{d}_{i+1}) \quad \text{is large.} \tag{5.35b}$$

The first condition states that the axis directions are not similar. The second condition says that either one or both of the extreme points of the middle cylinder C_{i+1} are off the alignment.

To fix this type of situation, the axis direction of the second cylinder must also be altered in addition to its starting point and length. The end point of the first

cylinder is updated as in the previous case. The updated middle cylinder will have an axis direction that has the same direction as the vector from the new end point of the cylinder C_i to the starting point of the cylinder C_{i+2}. The length of the cylinder will be the length of that vector, and the starting point will be the new end point of the cylinder C_i.

Correct alignment and direction In the most common case the alignment and the axis direction of the middle cylinder are correct. The problem is then to get rid of the gaps between the cylinders. This achieved as in the previous cases with the following steps:

- Increase the length of the cylinder C_i by $|\mathbf{a}_i \cdot \mathbf{d}_i|$.
- Increase the length of the cylinder C_{i+1} by $|\mathbf{a}_{i+1} \cdot \mathbf{d}_{i+1}|$.

5.5.2 Filling gaps between segments

The previous section explained how gaps inside segments could be filled. The procedure was relatively simple since in a segment the parent and extension relations are all set. In this section, gaps between cylinders from different segments are inspected. The process is not as straightforward since it is not clear what cylinder should be the parent of the first cylinder in a segment, or even if such a cylinder exists.

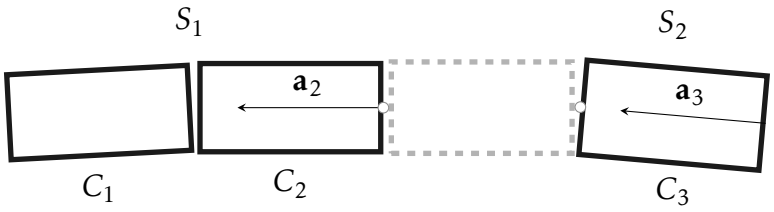
During the process when the gaps between the segments are filled, only cylinders that have no parent are considered. The assumption is that a cylinder might not have a parent because a cylinder is missing where the parent should be. This can be caused by gaps in the original measurements or by a failed fitting result during the cylinder fitting. The goal of the process is to fit an additional cylinder into the cylinder set in a way that makes the relational information more complete.

The inter-segment gap filling is divided into two stages according to the possible cylinder relations. The two relations are presented in figure 5.12 and explained below.

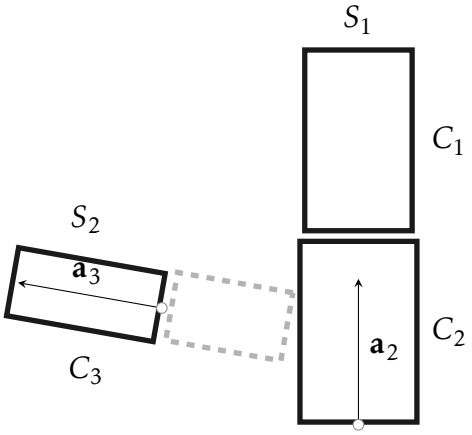
Parent – Extension
A cylinder is missing in the middle of a segment. This has caused the segment to be divided into two separate segments S_1 and S_2 in figure 5.12a. The new cylinder will be the extension of the last cylinder C_3 in the segment S_2 and the parent of the first cylinder C_2 in the segment S_1. The two segments are merged into one.

Parent – Child
The first cylinder of a segment is missing. In figure 5.12b the segment S_2 has no parent segment, and the cylinder C_3 has no parent cylinder. The new cylinder will be a child of the cylinder C_2 and the parent of the cylinder C_3. The segment relation between the two is also updated.

The two stages have to be completed in a specific order because the parent–extension -stage might add a cylinder that is essential in the parent–child -stage. This means that, if in the case presented in figure 5.12b, the cylinder C_2 were



(a) Parent – Extension



(b) Parent – Child

Figure 5.12: The two possible types of gaps between segments. In the upper case a segment has been split into two separate segments. In the lower case a segment has been isolated from its parent segment. In both cases the correcting cylinder is shown with a dashed line.

also missing, the new cylinder could not be found. However, if the cylinder C_2 is found earlier in the parent–extension -stage, the latter stage will work as shown in the figure.

The details of the stages are presented next. Both stages start by considering cylinders that have no parent cylinder set. Naturally, in an ideal case, the cylinder set would have only a single cylinder that has no parent, the root of the tree. However, if the parent of more than one cylinders is unset, it means that something has gone wrong. The process starts by finding candidate parents for the given cylinder.

Parent–extension -stage Let C_{ext} be a cylinder without a parent, with the axis direction $\mathbf{a}_{\text{ext}}$, starting point $\mathbf{p}_{\text{ext}}$ and a radius r_{ext}. A cylinder C_{can}, with the corresponding properties $\mathbf{a}_{\text{can}}$, $\mathbf{p}_{\text{can}}$, r_{can} and the end point $\mathbf{q}_{\text{can}}$, is a suitable candidate

if it has no extension and the following conditions are true:

$$\begin{aligned}
\sphericalangle(\mathbf{a}_{\text{ext}}, \mathbf{a}_{\text{can}}) &\quad \text{is small and positive,} && (5.36a)\\
|r_{\text{ext}} - r_{\text{can}}| &\quad \text{is small,} && (5.36b)\\
\|\mathbf{p}_{\text{ext}} - \mathbf{q}_{\text{can}}\| &\quad \text{is neither too large nor too small,} && (5.36c)\\
\sphericalangle(\mathbf{a}_{\text{ext}}, \mathbf{p}_{\text{ext}} - \mathbf{q}_{\text{can}}) &\quad \text{is small,} && (5.36d)\\
\mathbf{a}_{\text{ext}} \cdot \mathbf{p}_{\text{ext}} &> \mathbf{a}_{\text{ext}} \cdot \mathbf{q}_{\text{ext}} \,. && (5.36e)
\end{aligned}$$

The first condition (5.36a) ensures that both cylinders have similar directions. The positive restriction ensures that the directions are not close to opposites. The second condition (5.36b) filters out cylinders that are much smaller or larger than the given cylinder C_{ext}.

The third condition (5.36c) limits the distance between the parent and the extension. If the distance is larger than what could be filled with a *normal* length[7] cylinder, the candidate is probably not the correct parent. On the other hand, if the distance is small enough, no additional cylinder is needed. In such a situation, if all the other conditions are fulfilled, the parent–extension relation can be updated.

To find only the cylinders whose alignment is close to the given cylinder, the fourth condition 5.36d is required. Since the distance $\|\mathbf{p}_{\text{ext}} - \mathbf{q}_{\text{can}})\|$ cannot be large, the angle $\sphericalangle(\mathbf{a}_{\text{ext}}, \mathbf{p}_{\text{ext}} - \mathbf{q}_{\text{can}})$ will grow large if the alignment is not correct.

The inequality (5.36e) ensures that the candidate cylinder is on the correct side of the given cylinder. This condition requires the end point of the candidate cylinder to be behind the start point of the given cylinder in the axis direction $\mathbf{a}_{\text{ext}}$.

The set of cylinders that pass the conditions is noted T_{can}. If the set is empty, filling the gap for the given cylinder is impossible. If multiple cylinders pass the conditions, the correct parent must be chosen according to some heuristic. Usually choosing the closest candidate works well. The chosen candidate cylinder is called C_{par}.

Once only one candidate remains, the gap between the given cylinder and the candidate cylinder must be checked for additional cylinders. This means that even though the cylinder C_{par} is the closest cylinder that passed the conditions, it does not mean that there cannot be cylinders closer that do not satisfy the filtering conditions. An example can be seen in figure 5.13. The cylinder C_3 is blocking the gap, but it is not a valid candidate because of its axis direction.

To find the possible blocking cylinder another filtering is required. If any cylinder C_{blk} in the cylinder model satisfies the following conditions for the given

[7] The limiting length can be, e.g., the maximum of the lengths of the candidate cylinder and the given cylinder.

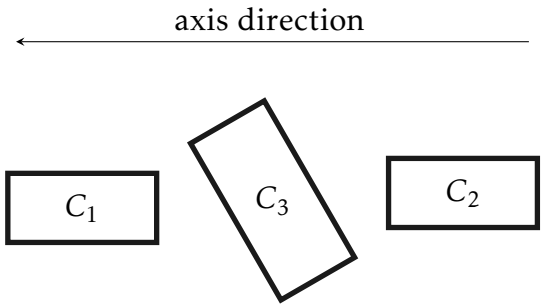


Figure 5.13: A non-candidate cylinder is blocking the gap. Each of the cylinder forms a segment on its own. The cylinder C_1 has no parent. The cylinder C_2 has no extension and it is a valid candidate for the parent of the cylinder C_1. However, a cylinder C_3 that is not a valid candidate is blocking the gap.

cylinder C_{ext}, the particular gap is blocked and can not be filled:

$$C_{\text{blk}} \notin T_{\text{can}} \,, \tag{5.37a}$$

$$\|\mathbf{p}_{\text{ext}} - \mathbf{p}_{\text{blk}}\| \quad \vee \quad \|\mathbf{p}_{\text{ext}} - \mathbf{q}_{\text{blk}}\| \quad \text{is not too large,} \tag{5.37b}$$

$$\begin{aligned} &\mathbf{a}_{\text{ext}} \cdot \mathbf{p}_{\text{ext}} < \mathbf{a}_{\text{ext}} \cdot \mathbf{p}_{\text{blk}} < \mathbf{a}_{\text{ext}} \cdot \mathbf{q}_{\text{par}} \quad \vee \\ &\mathbf{a}_{\text{ext}} \cdot \mathbf{p}_{\text{ext}} < \mathbf{a}_{\text{ext}} \cdot \mathbf{q}_{\text{blk}} < \mathbf{a}_{\text{ext}} \cdot \mathbf{q}_{\text{par}} \,, \end{aligned} \tag{5.37c}$$

$$\begin{aligned} &\mathbf{a}_{\text{blk}} \cdot \mathbf{p}_{\text{blk}} < \mathbf{a}_{\text{blk}} \cdot \mathbf{p}_{\text{ext}} < \mathbf{a}_{\text{blk}} \cdot \mathbf{q}_{\text{blk}} \quad \wedge \\ &\mathbf{a}_{\text{blk}} \cdot \mathbf{p}_{\text{blk}} > \mathbf{a}_{\text{blk}} \cdot \mathbf{q}_{\text{ext}} > \mathbf{a}_{\text{blk}} \cdot \mathbf{q}_{\text{blk}} \,. \end{aligned} \tag{5.37d}$$

First of all, the blocking cylinder can not be a parent candidate (equation (5.37a)), since the best candidate was chosen to be the closest one. Therefore none of the other candidates can be closer and block the gap. Equation (5.37b) states that at least one of the extreme points of the cylinder C_{blk} must be close to the cylinder C_{ext}.

The last two conditions are a bit more complex. Equation (5.37c) requires that, when measured in the axis direction $\mathbf{a}_{\text{ext}}$, at least one of the extreme points of the cylinder C_{blk} must be between the start point $\mathbf{p}_{\text{ext}}$ and the end point $\mathbf{q}_{\text{par}}$. The last condition (5.37d) ensures that the blocking cylinder is in the way. It states that when the start point $\mathbf{p}_{\text{ext}}$ is projected onto the axis of the blocking cylinder, the projected point is inside the blocking cylinder.

If any cylinder satisfies the conditions then the gap filling is a failure. If the gap is not blocked, it is filled by a new cylinder C_{gap} with the following properties:

$$\mathbf{a}_{\text{gap}} = \frac{\mathbf{p}_{\text{ext}} - \mathbf{q}_{\text{par}}}{\|\mathbf{p}_{\text{ext}} - \mathbf{q}_{\text{par}}\|} \,, \tag{5.38a}$$

$$\mathbf{p}_{\text{gap}} = \mathbf{q}_{\text{par}} \,, \tag{5.38b}$$

$$r_{\text{gap}} = \frac{r_{\text{par}} + r_{\text{ext}}}{2} \,, \tag{5.38c}$$

$$h_{\text{gap}} = \|\mathbf{p}_{\text{ext}} - \mathbf{q}_{\text{par}}\| \,. \tag{5.38d}$$

The cylinder relations are updated as well:

$$\text{parent}(C_{\text{ext}}) \leftarrow C_{\text{gap}} \,, \tag{5.39a}$$
$$\text{parent}(C_{\text{gap}}) \leftarrow C_{\text{par}} \,, \tag{5.39b}$$
$$\text{extension}(C_{\text{par}}) \leftarrow C_{\text{gap}} \,, \tag{5.39c}$$
$$\text{extension}(C_{\text{gap}}) \leftarrow C_{\text{ext}} \,. \tag{5.39d}$$

The segment containing the cylinder C_{ext} is merged with the segment containing the cylinder C_{par}.

Parent–child -stage Let C_{chd} be a cylinder without a parent for which an extension could not be found in the parent–extension -stage. Even though the cylinder is not an extension of any cylinder, it can still be a child of one. This means that the cylinder C_{chd} is the first cylinder in a child segment of some other segment.

As above, the process starts by finding candidate cylinders that satisfy certain conditions. The cylinder C_{can} is a suitable candidate if the following conditions are true:

$$r_{\text{chd}} \quad < \quad r_{\text{can}} \,, \tag{5.40a}$$
$$\sphericalangle(\mathbf{a}_{\text{chd}}, \mathbf{a}_{\text{can}}) \quad \text{is large,} \tag{5.40b}$$
$$d(C_{\text{can}}, \mathbf{p}_{\text{chd}}) \quad \text{is neither too large nor too small,} \tag{5.40c}$$
$$\mathbf{a}_{\text{chd}} \cdot \mathbf{p}_{\text{can}} < \mathbf{a}_{\text{chd}} \cdot \mathbf{p}_{\text{chd}} \quad \wedge \quad \mathbf{a}_{\text{chd}} \cdot \mathbf{q}_{\text{can}} < \mathbf{a}_{\text{chd}} \cdot \mathbf{p}_{\text{chd}} \,. \tag{5.40d}$$

The inequality (5.40a) states that the radius of the parent must be larger than the radius of the child. The second condition says that the axes of the two cylinders can not be too parallel. There must be a large enough angle between the child and its parent. The third condition (5.40c) means simply that the candidate cylinder must be close to the child cylinder, but again not too close so there is a gap to be filled. The last condition ensures that the whole candidate cylinder is behind the child cylinder in the axis direction of the child cylinder.

If none of the cylinders satisfy the given conditions, the gap can not be filled. The cylinders that passed the conditions need to be inspected a bit further. In order to do this a point called *attachment point* needs to be computed for each of the candidate cylinders.

If the child cylinder C_{chd} were extended backwards in its axis direction, the attachment point would be the point in which the starting point of the extended cylinder would meet the extended surface of the candidate cylinder (see figure 5.14). The attachment point $\mathbf{t}_{\text{can}}$ for the candidate cylinder C_{can} is computed as follows:

$$\mathbf{t}_{\text{can}} = \mathbf{p}_{\text{chd}} - \mathbf{a}_{\text{chd}} \frac{d(C_{\text{can}}, \mathbf{p}_{\text{chd}})}{\|\mathbf{a}_{\text{chd}} \times \mathbf{a}_{\text{can}}\|} \,. \tag{5.41}$$

When the attachment points for all the remaining candidate cylinders are computed, the cylinders that do not satisfy the following condition are excluded:

$$\mathbf{a}_{\text{can}} \cdot \mathbf{p}_{\text{can}} \leq \mathbf{a}_{\text{can}} \cdot \mathbf{t}_{\text{can}} \leq \mathbf{a}_{\text{can}} \cdot \mathbf{q}_{\text{can}} \,. \tag{5.42}$$

The inequality states that the projection of the attachment point on the axis of the candidate must be between the corresponding projections of the extreme points

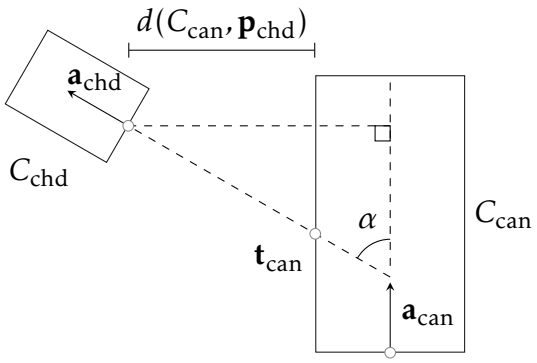


Figure 5.14: Attachment point of the child cylinder C_{chd} on the candidate cylinder C_{can}. The location of the attachment point $\mathbf{t}_{\text{can}}$ defines whether the candidate cylinder can be the parent.

of the candidate cylinder. The attachment point filtering ensures that the best candidate in the axis direction of the child cylinder is chosen.

If any candidates remain, the closest one C_{par} is chosen and the gap must be checked for blocking cylinders as in the parent–extension-stage. If the gap is blocked, the process is terminated. Otherwise, a cylinder with the following properties is added to the cylinder model:[8]

$$\mathbf{a}_{\text{gap}} = \mathbf{a}_{\text{chd}} \,, \tag{5.43a}$$

$$\mathbf{p}_{\text{gap}} = \mathbf{t}_{\text{par}} \,, \tag{5.43b}$$

$$r_{\text{gap}} = r_{\text{chd}} \,, \tag{5.43c}$$

$$h_{\text{gap}} = \|\mathbf{p}_{\text{chd}} - \mathbf{t}_{\text{par}}\| \,. \tag{5.43d}$$

The cylinder relations are updated as follows:

$$\text{parent}(C_{\text{chd}}) \leftarrow C_{\text{gap}} \,, \tag{5.44a}$$

$$\text{parent}(C_{\text{gap}}) \leftarrow C_{\text{par}} \,, \tag{5.44b}$$

$$\text{extension}(C_{\text{gap}}) \leftarrow C_{\text{chd}} \,, \tag{5.44c}$$

$$\text{children}(C_{\text{par}}) \leftarrow \text{children}(C_{\text{par}}) \cup C_{\text{gap}} \,. \tag{5.44d}$$

The new cylinder C_{gap} is added to the segment of the child cylinder.

Once all the cylinders without parents have been processed, the gap filling is complete. The possibly updated cylinder model is the final model that can be used for further analysis such as the branch size distribution.

[8] Another possibility is to extend the child cylinder backwards until it reaches the attachment point of the parent cylinder.

6 Validation and examples

This chapter validates the presented method by using several examples with both synthetic and real LIDAR measurements. In section 6.1, a synthetic test tree is first generated and measurements are taken from its surface. The measured points are then input to the algorithm to get an estimate of the surface, volume, and the branch size distribution of the tree. The results are compared to the respective, well-known properties of the original, generated tree.

In section 6.2, real measurements are used as input to show the capabilities of the algorithm. Since the exact volume and branch size distribution of a laser scanned tree are not known, the accuracy of the computed properties is estimated based on the visualization of the results.

6.1 Generated data

In this section, the accuracy of the implementation is tested with generated point cloud data. The data are generated from a cylinder model which has a structure similar to the resulting cylinder model. This makes it possible to study and visualize the differences in the original and the reconstructed model.

Section 6.1.1 explains how the point cloud data are generated from the original cylinder model. The generation is based on taking random samples from the surfaces of the cylinders in such a way that the sample density remains the same in all of the cylinders.

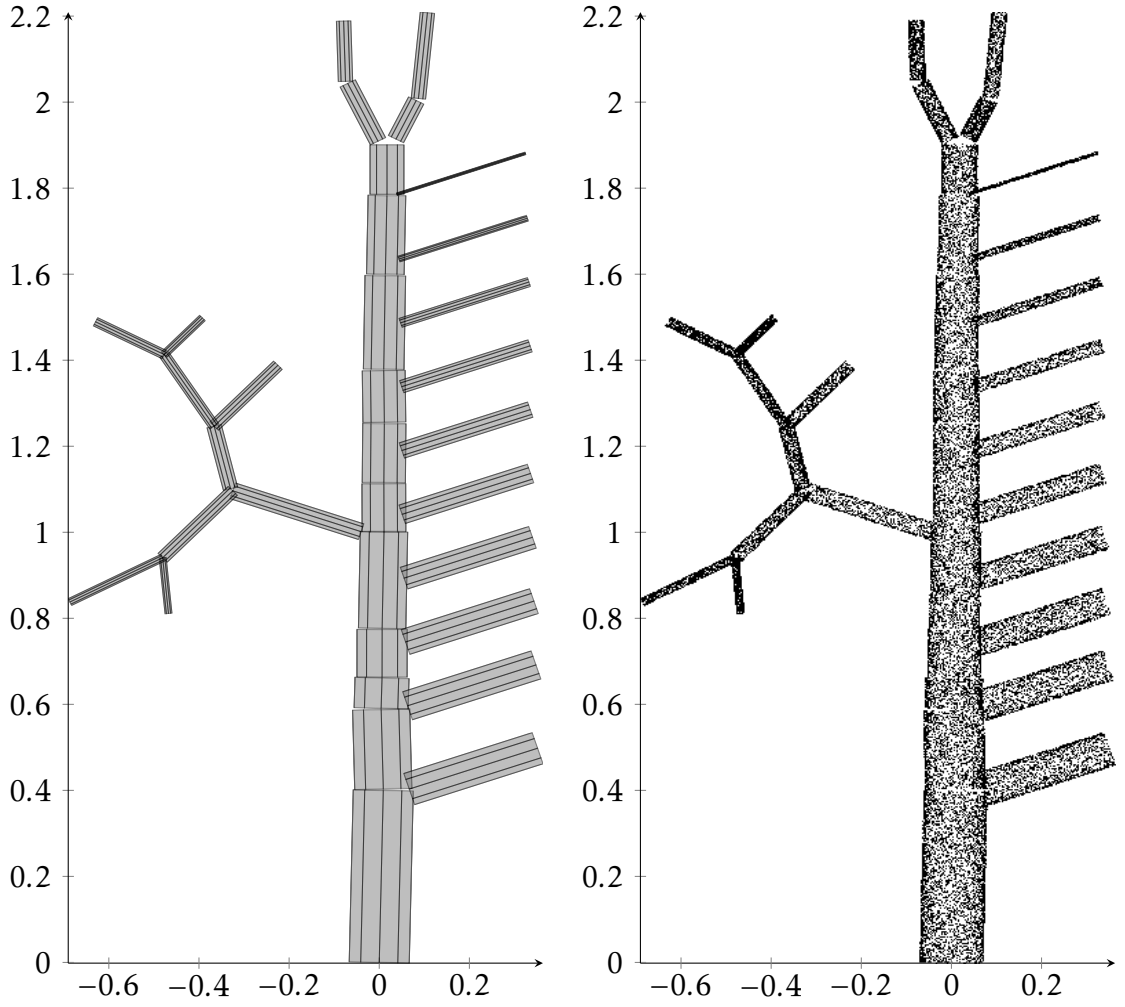
Once the data are generated, section 6.1.2 shows how good the results can be when the measurement data is noise-free. The intermediate results of the different stages are also presented, as are some key quantities such as the number of segments.

Later on in this section, the effects of measurement noise (section 6.1.4) and measurement density (section 6.1.5) are studied. The effect of the cover set radius is tested in a similar manner in section 6.1.3.

6.1.1 Data generation

For this example, a tree is defined as a collection of cylinders. The cylinders are chosen in a way that makes the complete cylinder model easy to visualize in two dimensions (see figure 6.1a). Furthermore, to demonstrate the capabilities of the implementation of the tree modeling algorithm, the cylinder model contains multiple bifurcations and cylinders with various size radii. To make the discussion simpler, the parts of the tree are named according to their purpose. The part names and descriptions are given in table 6.1.

To simulate the laser scanning measurements, random samples were taken from the surfaces of all the defined cylinders. To keep the point density constant, the number of samples in the largest cylinder was fixed and the number of samples in the other cylinders was scaled by the surface area of the correspond-



(a) Cylinder model **(b)** Point cloud

Figure 6.1: A visualization of the generated data. On the left: the cylinder model of the *tree* that is sampled and reconstructed. On the right: the point cloud of the samples taken from the cylinder model.

Table 6.1: Names, sizes and descriptions of the parts of the generated tree.

Part	Cylinders	Description
Trunk	11	In the middle of the cylinder model formed by the cylinders whose axis direction is close to the z-axis.
Treetop	4	The bifurcated top part, above the trunk.
Bifurcation area	9	The left hand side part with the multiple bifurcations.
Branch area	10	The right hand side part with the branches with increasing radius when moving from top to bottom.
	34	cylinders all together

ing cylinder. An example of a point cloud formed by the samples can be seen in figure 6.1b.

Furthermore, to simulate measurement noise, Gaussian noise was added to the measurements. The effects of the noise level on the reconstruction accuracy are discussed in section 6.1.4. But first the algorithm is run once on a point cloud without any noise to test the reconstruction detail in the case of optimal data.

6.1.2 Running the algorithm

In addition to the point cloud generated above, the MATLAB® implementation requires the following parameters:

r_{cov} Radius of the cover sets used in the tree analysis process.
r_{fil} Radius of the cover sets used in the initial filtering (see section 4.5.1).
d_{cov} Minimum distance between the center points of the cover sets (see section 4.2).
n_{fin} Minimum number of elements in the final cover sets.
n_{fil} Minimum number of elements in the filtering cover sets.

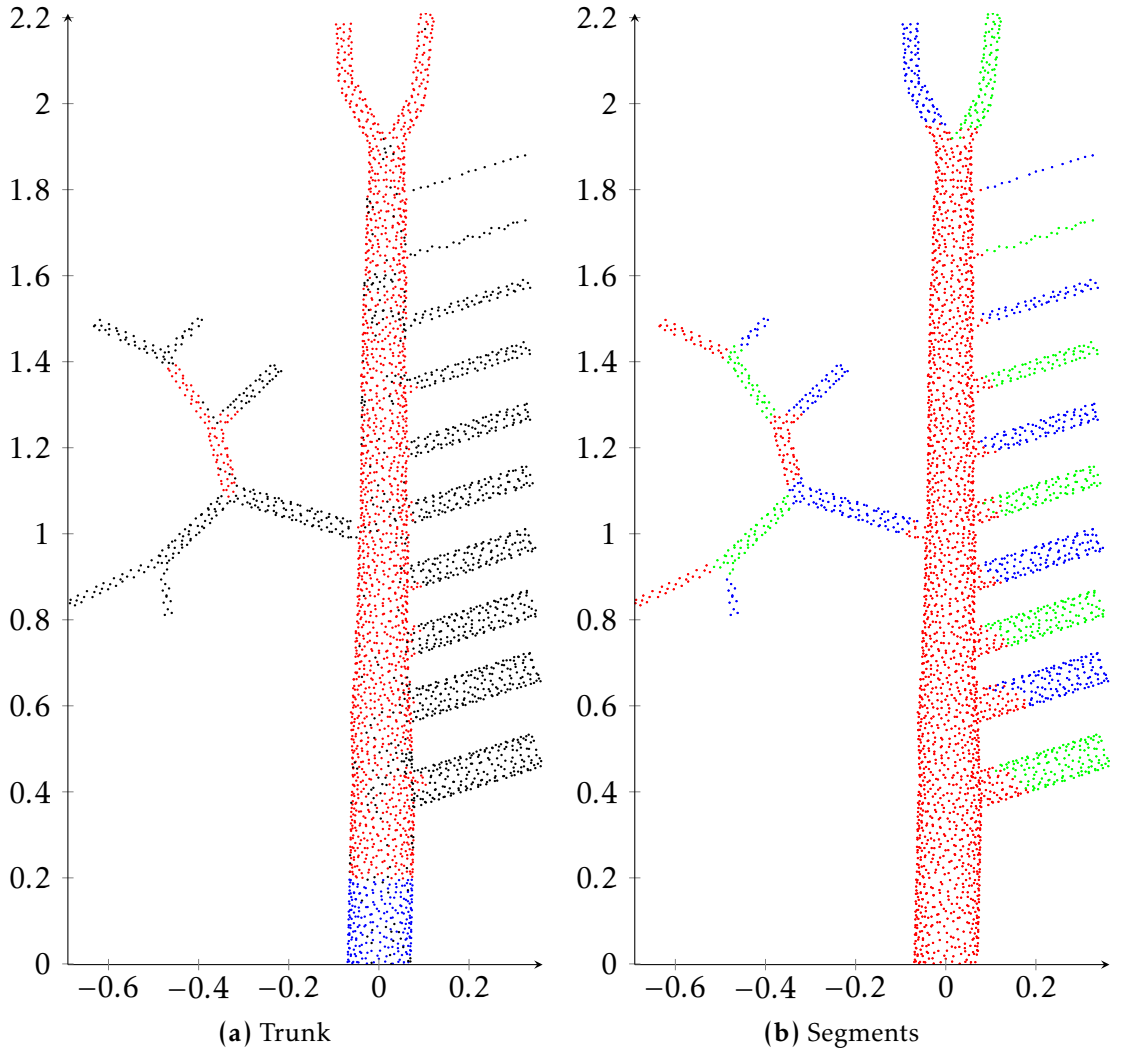
The following choices were made in this example to keep it as simple as possible:

$$r_{\text{cov}} = r_{\text{fil}} = d_{\text{cov}} = 0.016$$
$$n_{\text{fin}} = n_{\text{fil}} = 2$$

The value of the cover set radius was chosen by a method of trial-and-error. Usually a good rule of thumb is that the cover set radius should be close to the average branch radius. The minimum number of elements was set to a small value because in this case the data were known to be good and no filtering was required.

By using the above values, the number of cover sets was a little over 3400. The average cover set cardinality was 23.25 and the most common cardinality was 17.

The analysis began by the location of the trunk of the tree and the base of the trunk. The classification of the cover sets into trunk and non-trunk ones is



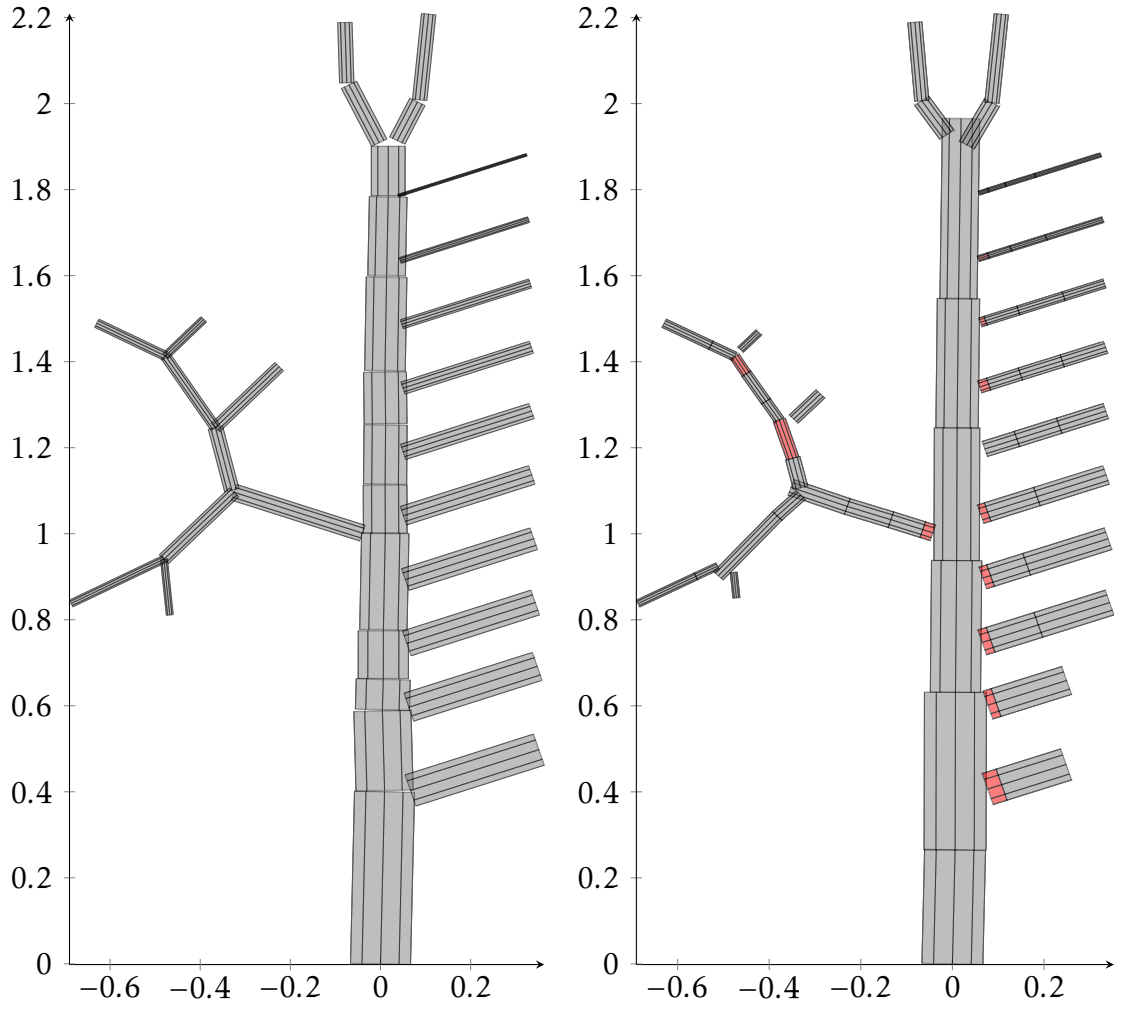
(a) Trunk **(b)** Segments

Figure 6.2: Classification of the cover sets during different stages of the algorithm. On the left-hand side, the cover sets classified as trunk are shown in red and the base of the tree shown in blue. On the right-hand side, the formed segments are shown in individual colours.

visible in figure 6.2a. The implementation correctly detected the measurements from the largest cylinders as the part of the trunk. However, some measurements in the bifurcation area and the treetop were erroneously classified as part of the trunk. The base of the tree was correctly found at the bottom of the trunk.

The segmentation process started from the base of the tree and formed a total of 22 segments. The formed segments are visualized in figure 6.2b. The segmentation process has worked correctly since each of the branches is detected. In the case of the segment covering the trunk, the implementation has, as expected, advanced quite far into the largest branches of the radius area before the study region has become disconnected. This causes gaps between the cylinders in the trunk segment and the largest branch cylinders. However, the gaps can be filled during the post-processing stage of the algorithm.

During the cylinder fitting process, 48 cylinders were successfully fitted to the



(a) Original **(b)** Reconstructed

Figure 6.3: Comparison of the original and the reconstructed cylinder model. Left: original model. Right: reconstructed cylinder model with the cylinders fitted during post-processing shown in red.

measurements. 12 additional cylinders were fitted during the post-processing stage. The result before and after the post-processing is shown in figure 6.3b. Most of the gaps (10 out of 12) were located between the trunk and the first order branches. The remaining two were in the bifurcation area. This was probably caused by inaccuracy during the segmentation process.

The original and the reconstructed cylinder model are shown side-by-side in figure 6.3. Additionally, the corresponding branch length and volume distributions are shown in figure 6.4.

When comparing the two cylinder models, it is easy to see that the reconstruction has worked extremely well. Most of the branches are reconstructed without any error and at least parts of every branch have been reconstructed. The most visible error seems to be in the two most bottom branches. They, and a few other branches in the bifurcation area, have been left a bit short. Since the segment formation was clearly successful, the error must be caused by incomplete checks

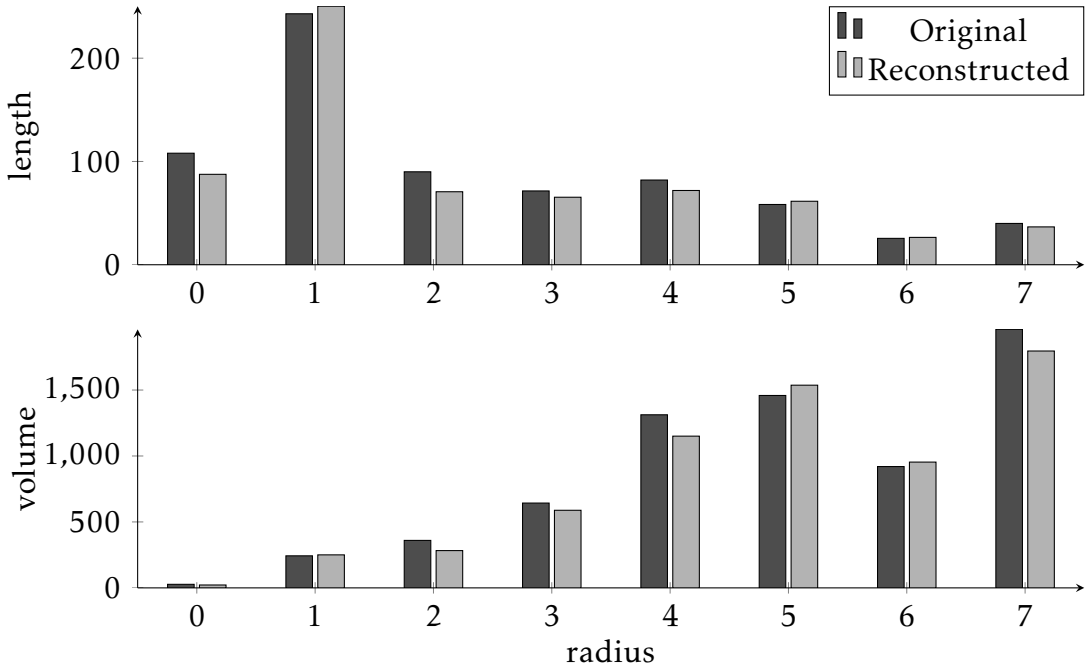


Figure 6.4: Length and volume distributions of the original and the reconstructed cylinder models.

in the current software and the chosen values of the parameters.

In the case of the trunk, the cylinders are a bit too large to properly follow the smoothly decreasing radius of the trunk. This has caused quite large differences in the radii of the consecutive cylinders of the trunk.

At the treetop, the trunk has extended a bit too far. All of the four cylinders in the treetop are reconstructed. The turning point in both of the branches is detected, but in the left hand side branch it is too low.

The purpose of the radius variation in the branch area was to test how small branches could be reconstructed, and in this case even the smallest branch ($r = 0.003$) was successfully reconstructed. However, the the radius of the reconstructed cylinder is slightly larger at 0.005. All the other branches have the correct radius.

In the bifurcation area, all the bifurcations have been detected and all the branches reconstructed. A couple of the branches have been left a bit too short and disconnected from the rest. Also, the first branch is too long since there is a lot of overlapping at the location of the first bifurcation.

The length distribution in figure 6.4 shows that in most of the size categories the added length of the cylinder in the reconstructed model was less than the one in the original model. This is consistent with the cylinder model visualization since some of the branches were left too short.

A possible source of error might also be that the cylinders in the branch area in the original model overlapped slightly with the trunk. This is not the case in the reconstructed model, which is good since in real measurement data such a phenomenon would be impossible.

As a total, the differences in the length and volume were

$$\begin{aligned}\Delta l &= l_{\text{orig}} - l_{\text{recon}} \\ &= 7.1839 - 6.7044 \\ &\approx 0.479 \\ \Delta V &= V_{\text{orig}} - V_{\text{recon}} \\ &= 8.4107 \cdot 10^{-3} - 8.3973 \cdot 10^{-3} \\ &\approx 0.0134 \cdot 10^{-3}.\end{aligned}$$

The relative error in the lengths is 6.7% and in the volumes 0.2%. The values are used to compare the results with different setting in the upcoming sections.

6.1.3 Effect of cover set radius

The radius of the cover sets used during the analysis is a crucial parameter. When the radius becomes smaller, increasingly finer details can be captured, e.g., two small branches growing side-by-side can be distinguished. However, at the same time the connectedness of the point cloud can change if it breaks into smaller components. In the extreme, case the radius would be so small that none of the points are connected to each other and the reconstruction would be impossible.

If the radius size increases, the level of detail becomes poorer. When the radius grows larger, the number of required cover sets becomes smaller, which in turn makes the computations faster.

Next, the effect of the cover set radius is studied in the case of the generated cylinder model. The reconstruction is run with radius values higher and lower than the optimal one to show how it affects the result. The key values from the fitting process with different cover set radius values are listed in table 6.2. The values from the optimal case, where the radius was 0.016, are highlighted in the table.

The table shows that as the cover set radius becomes larger, the number of cover sets naturally becomes smaller as fewer of them are required to cover all the points in the point cloud. At the same time, the average number of elements in the cover set rises as the radius becomes larger. The number of formed segments is a bit more unpredictable.

When the cover set radius is very small, $r_{\text{cov}} = 0.010$, the number of segments is very high. This happens because, as the cover sets become smaller, even the minor irregularities in the point cloud can be interpreted as bifurcations causing a new segment to be started. Many small segments are formed where they should, and the average segment size, measured in points, is almost half of what is with the second lowest radius 0.012.

As the cover sets radius starts to grow, the number of segments stays around 20 up to a radius value around 0.060. Similarly, the average segment size stays almost constant for the same radius interval. Only at the highest values does the number of segments start to drop quickly, and with the radius value 0.100, only six segments are formed, and only five cylinders are later successfully fitted to those segments.

With the smallest radius values 0.010 and 0.012, the number of successfully fitted cylinders is fairly large, but the relative volume error shows that the cylin-

Table 6.2: Quantities from the cylinder fitting process with different cover set radius values. The columns from left to right are cover set radius, number of cover sets, mean number of points in a cover set, number of formed segments, mean number of points in a segment, number of successfully fitted cylinder, total length of cylinder, difference to total length of original, relative difference in length, total volume of cylinders, difference to total volume of original, relative difference in volume.

r_{cov}	n_{cov}	$\overline{\text{cover size}}$	n_{seg}	$\overline{\text{seg size}}$	n_{cyl}	$\sum l\ (\cdot 10^2)$	$\Delta l\ (\cdot 10^2)$	$\Delta l / l_{\text{orig}}(\%)$	$\sum V\ (\cdot 10^6)$	$\Delta V\ (\cdot 10^6)$	$\Delta V / V_{\text{orig}}(\%)$
0.010	7214	9.18	62	1079.84	122	530.01	188.38	26.2	1060.68	7349.97	87.4
0.012	5486	13.03	25	2029.84	80	594.40	123.99	17.3	4431.21	3979.44	47.3
0.014	4238	17.80	21	3544.81	66	671.29	47.10	6.6	8188.85	221.80	2.6
0.015	3813	20.45	23	3356.65	74	716.19	2.20	0.3	8756.02	345.37	4.1
0.016	3414	23.25	22	3586.59	60	670.44	47.95	6.7	8397.28	13.37	0.2
0.017	3050	26.21	20	3978.30	55	620.13	98.26	13.7	8227.34	183.31	2.2
0.018	2762	29.48	18	4498.17	58	663.63	54.76	7.6	8821.58	410.93	4.9
0.020	2277	36.48	20	4143.45	53	617.29	101.10	14.1	7981.99	428.65	5.1
0.022	1908	43.78	21	3974.00	48	576.90	141.49	19.7	8790.22	379.57	4.5
0.025	1492	56.89	21	4030.14	38	495.93	222.46	31.0	7811.79	598.86	7.1
0.030	1059	82.16	20	4348.35	38	486.54	231.85	32.3	8382.39	28.26	0.3
0.035	770	111.24	19	4508.21	33	435.49	282.90	39.4	7977.30	433.35	5.2
0.040	589	144.31	18	4722.11	21	456.54	261.85	36.4	5750.98	2659.67	31.6
0.050	373	223.85	18	4638.67	30	463.73	254.66	35.4	8775.78	365.13	4.3
0.060	257	320.11	17	4839.29	15	379.51	338.88	47.2	5401.59	3009.06	35.8
0.070	184	432.60	11	7140.64	10	314.90	403.49	56.2	4886.31	3524.34	41.9
0.080	143	570.71	12	6800.92	8	129.53	588.86	82.0	1650.33	6760.32	80.4
0.100	94	851.80	6	13344.83	5	89.31	629.08	87.6	1352.42	7058.23	83.9
original						718.39			8410.65		

ders clearly have wrong sizes. The badly formed small segments actually caused some of the fitted cylinders to be inside one another, which probably contributed greatly to the error level. Even though the number of cylinders is large, the total volume is more than ten times too low, which shows that the cylinders are very much too small in diameter.

From the table it is fairly easy to see why the cover set radius value 0.016 was chosen as the optimal reconstruction. Both the length and volume errors are very small with this radius value. With the value 0.015 the error levels were equally low, but the visualization showed that several small cylinders were incorrectly placed inside larger ones as with the smaller radius values above.

It should be noted that the fitting error level is hard to predict. For example, when changing the cover set radius from 0.016 to 0.017, both the length and the volume error double, and furthermore drop to half when moving from 0.017 to 0.018. Even though the error level tells something about the success level of the fitting process, it should be remembered that visualizations are still the best way to compare the fitting results. Figure 6.5 shows the results where the error levels are small and close to each other but the visualizations are very different.

The results also show that, independent from the cover set radius, the total length of the fitted cylinders never rose above the total length of the original model. This means that, in the sense of the length, too many cylinders where never fitted. However, measured by the volume the case was different. In a couple of cases ($r_{\text{cov}} = 0.022, 0.050$) the total volume was more than the original which shows that the fitting is probably less accurate in the radial direction.

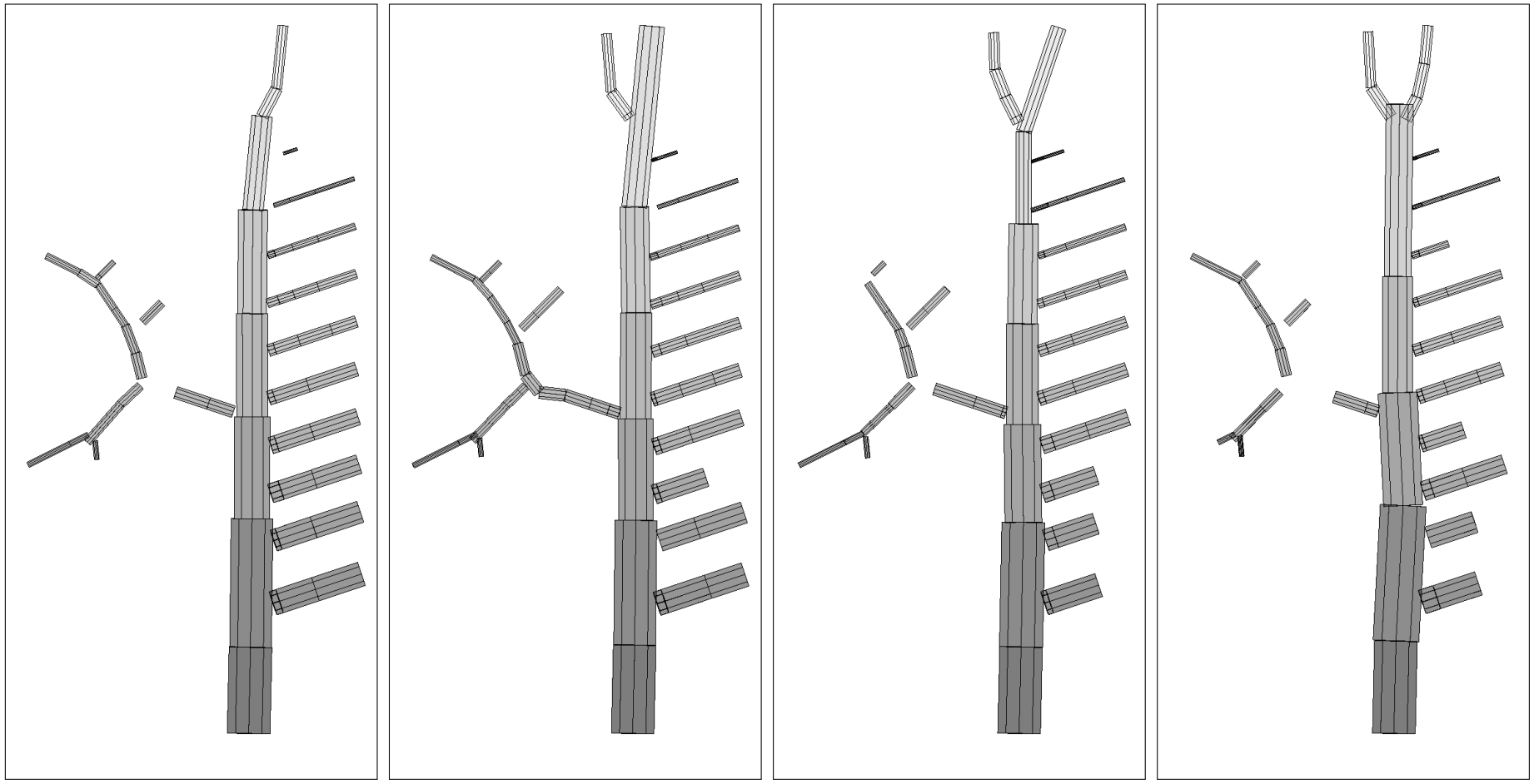
6.1.4 Effect of noise level

In this section, the same cylinder model that was generated in section 6.1.1 is used. In section 6.1.2 the algorithm was applied to an ideal set of data, whereas in this section the same process is repeated, but now an increasing amount of Gaussian noise is added to the random samples taken from the cylinders. The purpose is to find a limit for the level of the noise, below which the object is still reconstructible.

The noise is added in the radial direction. This means that in the ideal case, all of the measurements were on the surface of the original cylinder, but now the measurements are moved either towards or away from the axis in the radial direction but not in the axis direction. The number of measurements remains constant. The noise level is increased step-by-step from 0.001 to 0.012. At each noise level ε, the measurement noise is distributed in the interval $[-\varepsilon, \varepsilon]$. The measurement noise is absolute, so measurements from each cylinder have the same magnitude of error independent from the corresponding cylinder radius. The noise is measured in same length units as, e.g., the cylinder properties.

At each level, the cover set radius was adjusted by trial-and-error to minimize the length and volume differences between the original and the reconstructed cylinder model. The results are listed in table 6.3.

The results show that, as the noise level grows, the cover set radius must also be increased or the reconstruction may become poorer and the point cloud may become disconnected. Naturally, as the cover set radius grows, the number of cover sets decreases. The number of formed segments is again harder to pre-



(a) $r_{\text{cov}} = 0.017$ **(b)** $r_{\text{cov}} = 0.018$ **(c)** $r_{\text{cov}} = 0.020$ **(d)** $r_{\text{cov}} = 0.022$

Figure 6.5: Reconstructed cylinder models with various cover set radius values.

dict as it seems that the number of segments becomes more erratic as the noise becomes larger.

The relative length error increases steadily until the noise level 0.011. There is also a significant jump in both relative errors at 0.008. The volume error is naturally more dependent on the noise level since the noise was applied in the radial direction.

With the smaller noise levels ($\varepsilon < 0.006$) almost complete reconstruction was still possible, and even with the highest noise levels partial reconstruction was possible. The trunk and the largest branches very successfully reconstructed in most cases. The bifurcation in the treetop was the part where the algorithm struggled the most. At high noise levels, the smaller branches in the branch and the bifurcation areas were only partly reconstructed.

In real laser scanning equipment, the measurement error level is close to the beam diameter, which is 2–3 millimeters. If the units in this example are considered as meters, the real measurement error levels are much lower than what the modeling method can handle (8 mm). In addition to the equipment, other factors, such as wind conditions, can contribute to the measurement error as well.

6.1.5 Effect of point density

The measurement density plays a key role when trying to reconstruct a surface from a set of samples. Once the density gets under a certain point, it becomes impossible to reconstruct the finer details of the surface. Eventually, when the density falls below a certain limit, no part of the surface can be reconstructed accurately.

In this section, the point density of the generated samples is reduced step-by-step in order to find the minimum density with which the reconstruction is still possible. It should be noted that this minimum density is not the same for all surfaces. Naturally, the density for a single cylinder can be lower than with a complex tree model. The limit will, however, give some sense of the magnitude required in the case of tree surface reconstruction. During this test, the noise level remained constant at $\varepsilon = 0.003$.

The number of samples taken from the cylinder $C_{\max}$ with the largest surface area was fixed at each step to a number $x_{\max}$. The number of samples taken from the other cylinders was scaled such that the number of samples per area remained constant. The number of samples taken from the cylinder C_i is

$$n_{\text{pts}}^{C_i} = \frac{x_{\max}}{A_{C_{\max}}} \cdot A_{C_i} \tag{6.1}$$

$$= \rho \cdot A_{C_i}\,, \tag{6.2}$$

where ρ is the point density, and $n_{\text{pts}}^{C_i}$ is the number of samples taken from the cylinder C_i. In this particular example, the maximum cylinder envelope area is 0.1759. If the units are thought as meters then, for example, the point density corresponding to $x_{\max} = 4000$ is 2.27 points/cm^2 (a typical value in practice).

Results from reconstruction attempts with various sample densities are listed in table 6.4. The table shows the sample density, the total number of points, and the cover set radius that minimized the relative length and volume errors. The radius was found through trial-and-error. As the sample density becomes

Table 6.3: Effects of the measurement noise. Values of different quantities from during the tree analysis process with point clouds with different noise levels. The noise level ε is measured in the same length units as, for example, the cover set radius r_{cov}.

ε	r_{cov}	n_{cov}	n_{seg}	n_{cyl}	$\Delta l/l_{\text{orig}}$ (%)	$\Delta V/V_{\text{orig}}$ (%)
0.001	0.0153	3637	21	64	1.7	3.0
0.002	0.0157	3502	25	71	1.3	4.1
0.003	0.0157	3516	25	70	1.2	3.4
0.004	0.0155	3656	26	69	0.5	6.2
0.005	0.0168	3218	24	70	2.8	4.2
0.006	0.0176	3083	21	61	4.6	7.6
0.007	0.0195	2607	21	59	6.0	1.3
0.008	0.0208	2402	18	49	12.8	15.4
0.009	0.0215	2332	38	79	11.1	57.7
0.010	0.0220	2364	37	76	13.4	64.6
0.011	0.0199	3262	16	37	44.8	86.6
0.012	0.0244	2148	16	35	43.3	83.0

Table 6.4: Effect of the measurement density.

x_{max}	n_{pts}	r_{cov}	n_{cov}	n_{seg}	n_{cyl}	$\Delta l/l_{\text{orig}}$ (%)	$\Delta V/V_{\text{orig}}$ (%)
4000	36631	0.0151	3706	23	65	3.6	0.1
3000	27474	0.0153	3445	28	77	0.3	0.7
2500	22896	0.0176	2637	26	69	0.1	0.6
2000	18320	0.0172	2611	33	78	1.8	2.4
1750	16028	0.0181	2363	29	75	5.2	9.7
1500	13739	0.0216	1687	24	57	10.5	3.3
1250	11452	0.0212	1684	25	63	12.6	3.6
1000	9163	0.0225	1451	25	54	19.3	15.6
900	8247	0.0223	1436	50	93	12.6	42.3
800	7332	0.0241	1227	21	52	17.9	21.6
700	6414	0.0253	1093	45	88	9.4	25.6
600	5500	0.0269	947	23	55	21.1	24.9
500	4587	0.0298	782	19	42	29.9	24.1
400	3670	0.0281	791	5	7	86.8	95.6

smaller, the cover set radius grows larger and the number of cover sets decreases. The number of segments stays roughly between 20 and 30.

The relative error levels state that the complete reconstruction requires a density above 2000. If the density is greatly above 2000, the level of reconstruction does not improve. When using sample densities below 2000, the relative length error is the first to start growing. This means that some cylinders become impossible to reconstruct with the given number of samples. At the density 1000, the relative volume error also starts to grow. After that, the error levels stay around 20 to 25 percent, until at the density 400 the reconstruction becomes impossible as the errors grow close to 100 percent.

6.2 Measured data

In section 6.1, a point cloud based on a synthetic cylinder model was generated. In this section, the algorithm is applied to a real set of terrestrial laser scanning measurements. The difference between the two sources of data is that the real scanning measurements contain noise whose level is not constant, phantom measurements, and measurements from the ground. Also, the sample density varies a lot more.

The topological structure of the tree is more complicated than in the synthetic example. There will be a lot more than one component in the tree, and thousands of segments. The downside is that the exact structure of the tree is unknown. That means that the validity of the reconstruction can only be evaluated by comparing the visualizations of the point cloud and the produced cylinder model with each other.

The tree used in this example is a birch tree that has been scanned in the winter time without any leaves. The bottom half of the trunk of the tree has been omitted from the sample set. The laser scanning data consists of 1783261 samples. A thinned-out version of the point cloud has been visualized in figure 6.6.

As shown in the figure, the data contain phantom measurements so, unlike in the case of the synthetic data, filtering is required. Each component was required to have at least 20 cover sets in them to be accepted for analysis. Each cover set was required to have at least 5 points in them. These filtering conditions combined with the cover set radius caused 145549 points to be filtered out.

The cover set radius was chosen to be 0.027. This choice resulted in 197166 cover sets. The point cloud then consisted of 654 components which in turn were divided into 4638 segments. During the cylinder fitting process, 8050 cylinders were successfully fitted to the segments. Additional 746 cylinders were added during the post-processing stage. The resulting cylinder model consisted of 8796 cylinders. The model is visualized in figure 6.7 .

The largest branches and the trunk are segmented properly and reconstructed accurately by the algorithm. Many of the smaller branches are also completely reconstructed, but in many cases pieces are missing somewhere in between. The total length of all the cylinders is approximately 890 meters. The total volume is 0.28 cubic meters.

The total surface area of the cylinders is 79.18 square meters. The ratio between the number of samples and the surface area gives the sample density

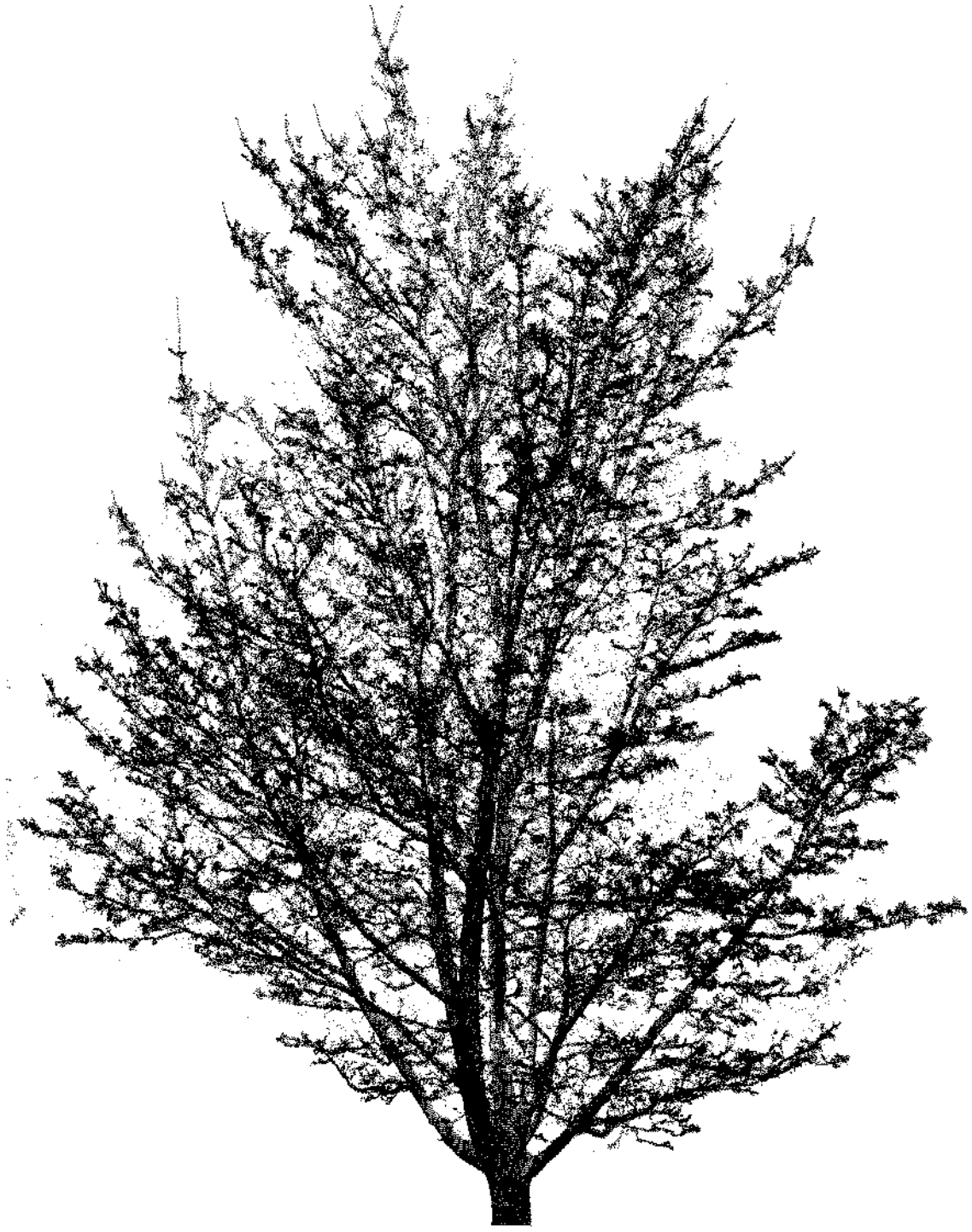


Figure 6.6: Point cloud produced by terrestrial laser scanning of a deciduous tree. The original scanning data consisted of 1783261 points but the visualized version has been thinned out to 594421 samples. The data are provided by the Finnish Geodetic Institute.

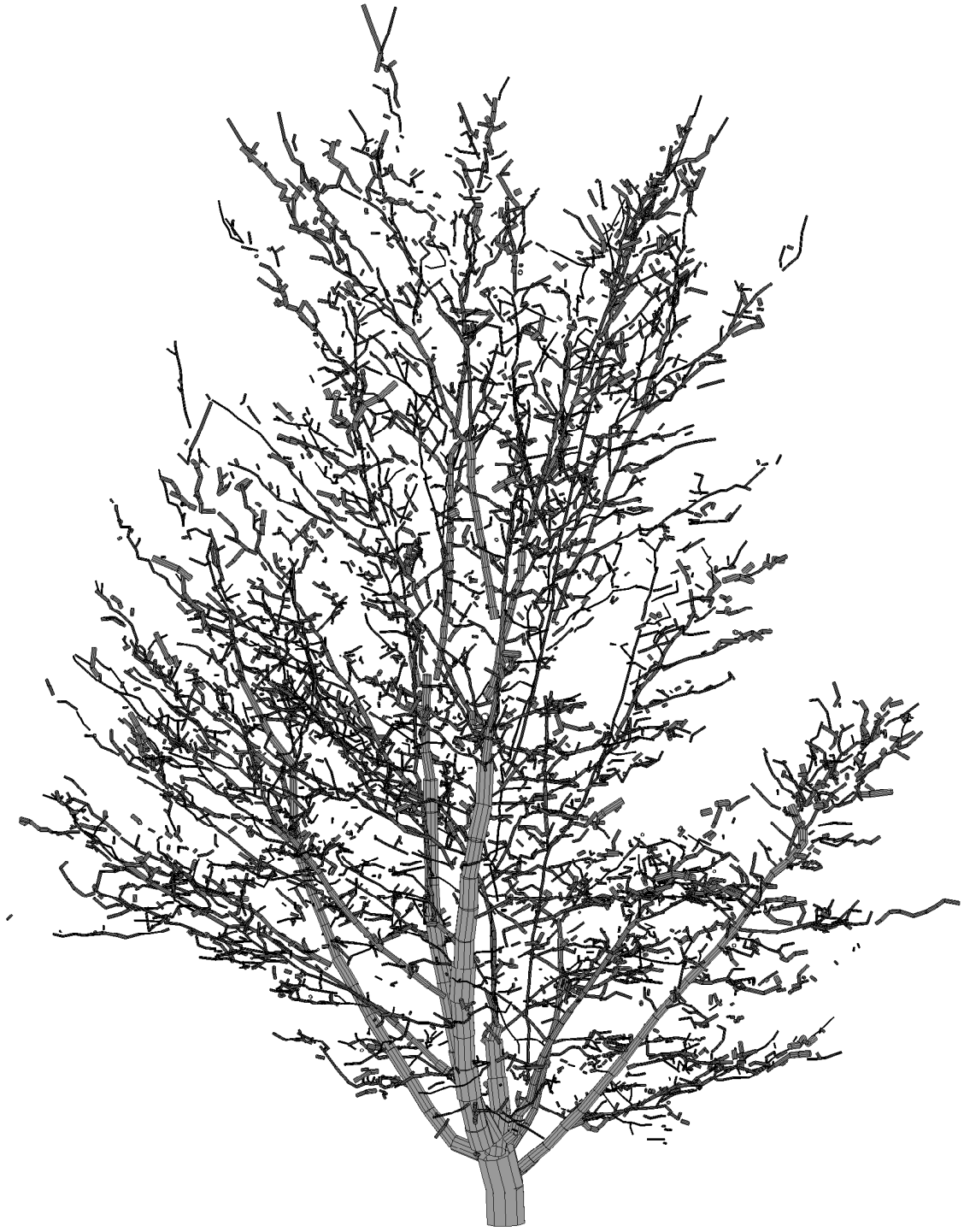


Figure 6.7: Cylinder model of a deciduous tree reconstructed from the point cloud. The model has a total of 8796 cylinders.

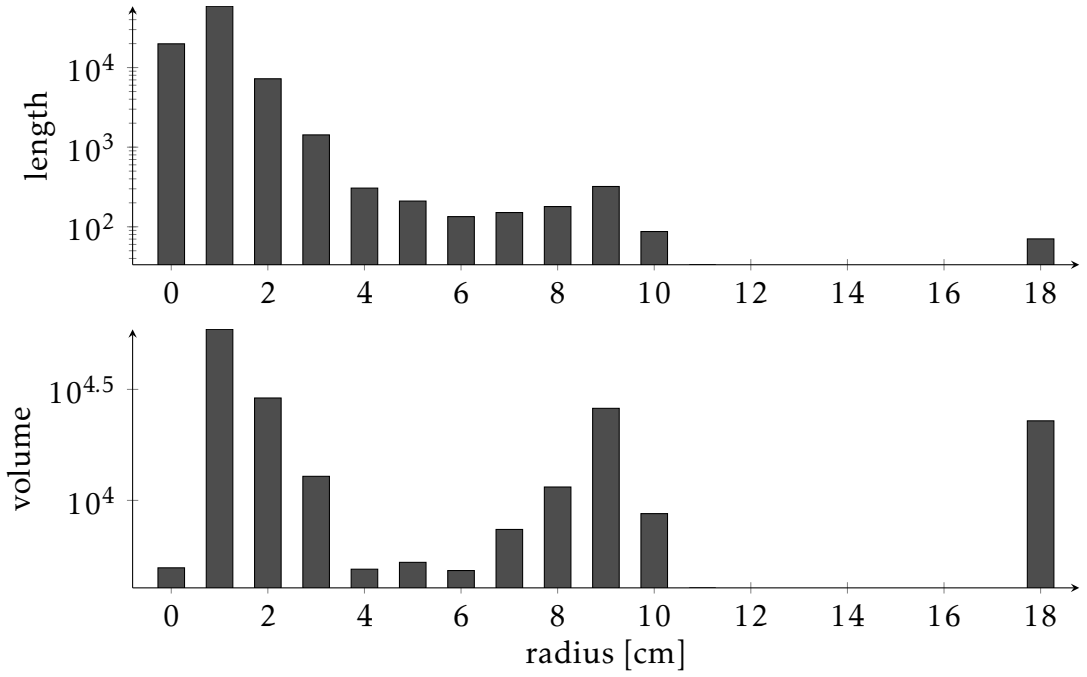


Figure 6.8: Absolute length and volume distributions for the reconstructed cylinder model. Note the logarithmic scaling of the vertical axis.

2.25points/cm^2. Since not all of the tree was successfully reconstructed, the surface area can only be considered as a lower bound and therefore the sample density as an upper bound. If this sample density is compared to the results of the synthetic case, it corresponds to a case where around 3960 samples were taken from the largest cylinder. This means that terrestrial laser scanning can provide the sample density required by the algorithm for a complete reconstruction of the surface of a tree.[1]

As for the relations between the cylinders, 92.3 percent of all of the cylinders have a parent set. Furthermore, 53.6 percent of cylinders are connected through a sequence of cylinders to the first cylinder which is at the base of the tree. This means that the structure of the tree is detected quite well. The longest sequence of cylinders goes from the base of the tree all the way to the top, and it is formed by 85 cylinders. The average parent sequence is 23 cylinders long.

The length and volume distribution computed from the cylinder model are visualized in figure 6.8. The distributions are shown in a logarithmic scale so the smaller values are also visible. Rather curiously, the model does not contain cylinders with radii $10.5 < r \leq 17.5$, but there are cylinders in the trunk of the tree with radius greater than 17.5.

6.2.1 Computational time

The relative times taken by the different stages of the tree analysis algorithm are shown in figure 6.9. To simplify the diagram, the cover set formation and characterization were combined, as were the cylinder fitting and the post-processing. In

[1] Even though the average sample density is not a problem, the uneven spread of the samples can be, as well as the measurement noise.

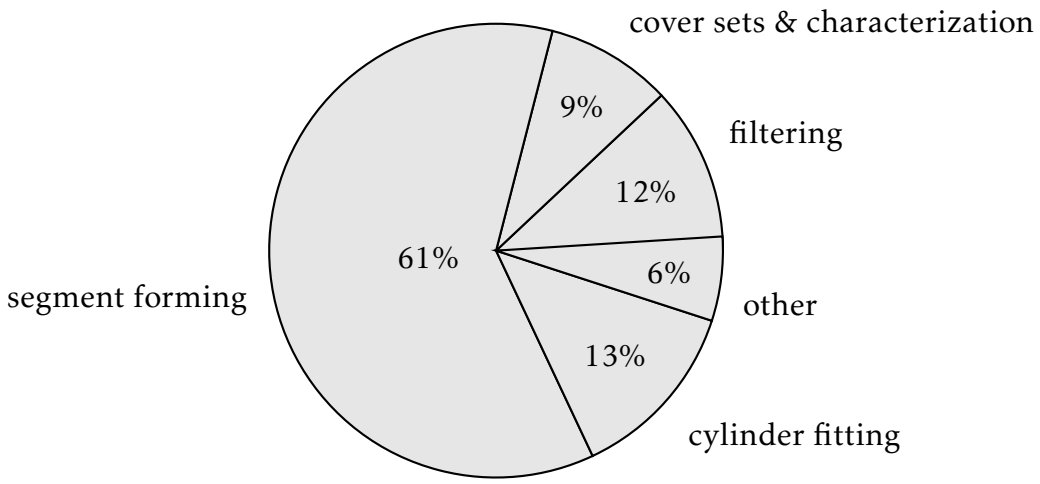


Figure 6.9: Relative computational time distribution between the different stages of the tree analysis process.

the latter pair, the cylinder fitting dominated the respective time distribution by 11 percentage units against 2 percentage units. In the first pair the percentages were 3 and 6, respectively.

The total process for the birch took 7 minutes and 3 seconds of CPU time on a 2.8GHz Intel Core i7 system. The segment forming is clearly the stage that takes the most time. Because the segment forming process goes through all the branches, its computational time is relative to the total length of the branches. Since this particular tree had a lot of branches measured in length, the segment forming took a lot of time. Notice that the filtering stage takes 12 percent of the complete time which is as much as the cylinder fitting stage.

7 Conclusion

In this thesis, a method for reconstructing a comprehensive quantitative tree model from a set of terrestrial laser scanning measurements was presented in detail. The reconstruction requires comprehensive measurements of the surface of the tree, and such measurements can be recorded using TLS. The measurement point cloud is processed using a topological approach which makes the method independent from scale. During the algorithm, the point cloud is segmented into branches, and the surface and volume of an individual branch is approximated by fitting multiple cylinders in to the measurement data. The result is a complete cylinder model that contains the location, size, and orientation of each branch and the trunk. Furthermore, the topological relations of the branches are also stored in the model, which means that the exact branching structure and properties like bifurcation angles are easily accessible.

The model can be used to compute properties, such as the partial volumes of the tree or the branch size distribution, easily and fast. Since the exact topological structure of the tree is known, it is also possible to do more extensive analyses of the properties of the tree on either the cylinder or the branch level.

The method is based on dividing the measurement point cloud into cover sets that conform to the surface of the tree. These intersecting sets are used in many stages of the reconstruction process to change the inspection scale from a local to a more global level and to make the analysis advance along the surface. The geometric properties of the cover sets are used to approximate the local properties of the tree surface, which are in turn used as initial values for the cylinder fitting.

The presented method was validated using generated cylindrical tree models, and real LIDAR measurements. A synthetic tree model was constructed from cylinders and random samples were taken from the model's surface to simulate laser scanning. The tolerance of the algorithm for measurement noise and sample density was studied with the help of the generated sample point cloud. Reconstruction of the cylinder model was tried with various noise levels and sample densities, and certain limits were identified. The reconstruction result was compared to the original cylinder model to evaluate the success level of the reconstruction.

The validation example was scale-free, but if the used units are considered as meters, the results show that the presented modeling method tolerates measurement error up to 7 millimeters. With greater levels the reconstruction level started to drop drastically. The measurement error caused by the equipment is usually a couple of millimeters, which is well in range for the algorithm to work.

A limit for the minimum sample density was found similarly. The sample density should be at least 1.14 points/cm^2. The reconstruction became better when the point density was increased up to 2.27 points/cm^2, but above that, a greater sample density did not improve the reconstruction notably any more. This is expected since the cylinder model is not designed to capture the smallest details of the surface.

The implementation was also used to reconstruct a tree from real terrestrial

laser scanning measurements. The reconstruction was a success as a total of 890 meters of cylinders were fitted to the point cloud. By using the surface area of the produced cylinder model and the sample count, an approximation of the sample density in the case of the real measurements was 2.25 points/cm^2. This means that, if the measurements can be spread out evenly enough, and the measurement noise can be kept low enough, the terrestrial laser scanning can provide suitable data for the algorithm to compute complete surface reconstructions.

The described tree modeling method has been implemented in MATLAB®. The complete reconstruction of the surface of a single tree can be done in a matter of minutes on up-to-date hardware (2.8GHz Intel Core i7, 8GB RAM). Even if the scanning data include millions of points, the analysis is fast because many of the stages of the algorithm only use the center points of the cover sets rather than all the points in the data set. Additionally, the use of partitioning makes the filtering and the cover set creation stages very fast. Currently, the segment forming stage takes most of the computational time, but the time distribution of the stages is naturally dependent on the branching structure of the tree.

7.1 Future work

In this thesis, the basis of the tree model algorithm was explained in detail. The development of the algorithm, however, still continues. Currently, the cylinder reconstruction is used solely to compute the branch size distributions, but it has much more potential. In the future, the algorithm can be used for multiple purposes, some of which are proposed next.

Analysing tree roots Currently only the visible part, i.e., trunk and branches[1], of the tree is reconstructed. The stump and the roots of the tree are not analysed even though the roots contribute largely to the biomass and the carbon footprint of a tree. It is possible to pull a stump out of the ground after the tree has been cut down and use a laser scanner to measure it. The characteristics of the measurements are similar to the ones from trees so, with small changes, the algorithm will be suitable for the roots as well.

By studying the size and structure of several stumps per every tree type, general assumptions could be made. The decomposition of the roots could be understood better, and the decisions could be made whether the stumps should be left in the forest beds or pulled out and burned for energy.

Different data sources The analysed measurements are received trough terrestrial laser scanning which currently is quite slow and the equipment is fairly expensive. To make accurate forest measurements available for everyone, cheaper and faster equipment is needed.

One of the future research goals is to study alternative ways to produce similar tree models from data sources other than terrestrial laser scanning. For example, high-resolution photographs taken from different directions are considered to be a valid study possibility. Another possibility is to use the combination of a camera and a low-end laser range meter.

[1] Since leaves are not cylindrical, they are best described as an additional statistical model.

Elliptic cylinders At the moment, the cross-section of the branches and the trunk is assumed to be circular ,so the fitted cylinders are also circular. This does not have to be the case, since once the fitting regions are formed it is very easy to change the shape of the surface to be fitted to the data.

For example, elliptic cylinders could be used instead of circular ones. This would mean that the cross-section could be an ellipse. By studying the ratio of the major and minor axes of the ellipses of the successfully fitted elliptical cylinders, one would know how close to circular the branches actually are. If they are clearly elliptical then the default fitting surface should be changed to an elliptic cylinder.

Wider analysis In this thesis, the main focus was on the algorithm and how to get from the point cloud to the cylinder model. But as mentioned earlier, the cylinder model is not the interesting result in itself. What is interesting is the information contained in the cylinder model. The presented branch size distribution is just one example of what can be done with the model. For example, the following interesting information is also easily computed from the cylinder model.

Number of child branches
How many child branches are bifurcated from a given branch? Is the number dependent on the radius of the branch? Is the count constant on every part of the tree? With several scans from the same tree, this quantity could be studied as a function of time, i.e., is the age of the branch a factor rather than its radius or length.

Bifurcation angles
When a child branch starts to grow from the parent branch there is always an angle between their axes. Is the angle constant or are there some geometric factors? Since the angle between the trunk and its child branches affects the quality of lumber, this quantity is very important to timber industry.

Parent–child radius ration
The radius of the child branch is smaller than the radius of the parent cylinder, but how much smaller? Is there a dependency? How does the radius of the child branch change with distance along the branch? The radii become smaller, but does the decrease follow some pattern? Could the number and size of the child branches be predicted before they become visible?

Statistical tree models
If the previous connections between the parent and child branches were combined and formulated as probabilities, a tree could be described with very few parameters as a statistical model. Furthermore, every tree would have such a statistical presentation and they could be compared. It would be interesting to see whether the statistical models from trees of the same type would be similar. Statistical models would also make it possible to generate trees that are visually very different but which have the same statistical properties.

Simulating trees and forests

By having the statistical models for different tree types, it would be possible to generate multiple instances of synthetic trees to simulate entire forests. Such simulations could be used to study how forests evolve over time. Another possibility would be to simulate airborne laser measurements and study how they are reflected from the synthetic forest, and how the measurements could be used to identify the tree types.

Bibliography

[1] T. Aschoff and H. Spiecker. Algorithms for the automatic detection of trees in laser scanner data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(Part 8):W2, 2004.

[2] K.H. Bae. *Automated registration of unorganised point clouds from terrestrial laser scanners*. PhD thesis, Curtin University of Technology., 2006.

[3] F.M. Danson, D. Hetherington, F. Morsdorf, B. Koetz, and B. Allgower. Forest canopy gap fraction from terrestrial laser scanning. *Geoscience and Remote Sensing Letters, IEEE*, 4(1):157–160, 2007.

[4] C. Dold and C. Brenner. Registration of terrestrial laser scanning data using planar patches and image data. In *ISPRS Comm. V Symposium "Iamge Engineering and Vision Metrology"*, *IAPRS*, volume 36, pages 25–27, 2006.

[5] W. Feller. *An introduction to probability theory and its applications*. Number vol. 2 in Wiley series in probability and mathematical statistics. Wiley, 1966.

[6] B. Gorte and N. Pfeifer. Structuring laser-scanned trees using 3d mathematical morphology. *International Archives of Photogrammetry and Remote Sensing*, 35(B5):929–933, 2004.

[7] C.W. Groetsch. *Inverse Problems: Activities for Undergraduates*. Classroom Resource Materials. Mathematical Association of America, 1999.

[8] A. Hartikka et al. Koealamittausperusteiset metsien inventointimenetelmät. Master's thesis, Mikkelin ammattikorkeakoulu, 2011.

[9] J. Hubbard. *Vector Calculus, Linear Algebra, and Differential Forms: A Unified Approach with Maple 10 VP*. Pearson Education, Limited, 2006.

[10] I.T. Jolliffe. *Principal component analysis*. Springer series in statistics. Springer-Verlag, 2002.

[11] L.C. Kinsey. *Topology of surfaces*. Undergraduate texts in mathematics. Springer-Verlag, 1993.

[12] A. Kirsch. *An Introduction to the Mathematical Theory of Inverse Problems*. Applied mathematical sciences. Springer, 2011.

[13] D.C. Lay. *Linear algebra and its applications*. Linear Algebra and Its Applications. Pearson/Addison-Wesley, 2006.

[14] G. Lukács, R. Martin, and D. Marshall. Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation. *Computer Vision-ECCV'98*, pages 671–686, 1998.

[15] M. Maltamo, P. Packalén, J. Uuttera, E. Äröla, and J. Heikkilä. Laserkeilaustulkinnan hyödyntäminen metsäsuunnittelun tietolähteenä. *Metsätieteen aikakauskirja*, 4(2008):304–309, 2008.

[16] B. Mendelson. *Introduction to topology*. Blackie & Son Limited, 1963.

[17] N. Pfeifer, B. Gorte, and D. Winterhalder. Automatic reconstruction of single trees from terrestrial laser scanner data. In *Proceedings of 20th ISPRS Congress*, pages 114–119. Citeseer, 2004.

[18] P. Raumonen, M. Kaasalainen, M. Åkerblom, S. Kaasalainen, H. Kaartinen, M. Vastaranta, and M. Holopainen. Comprehensive quantitative tree models from terrestrial laser scanner data. *Canadian Journal of Forest Research*, submitted.

[19] P. Raumonen, S. Kaasalainen, M. Kaasalainen, and H. Kaartinen. Approximation of volume and branch size distribution of trees from laser scanner data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2011.

[20] N. Ripperda and C. Brenner. Marker-free registration of terrestrial laser scans using the normal distribution transform. *Proceedings of the ISPRS working group*, 4, 2005.

[21] K. Ruohonen. *Vektorikentät*. Opetusmoniste, Tampereen teknillinen yliopisto. Tampere University of Technology, 2006.

[22] C. Schütt, T. Aschoff, D. Winterhalder, M. Thies, U. Kretschmer, and H. Spiecker. Approaches for recognition of wood quality of standing trees based on terrestrial laserscanner data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(Part 8):W2, 2004.

[23] M.Ó. Searcóid. *Metric spaces*. Springer undergraduate mathematics series. Springer, 2006.

[24] S.C. Sharma. *Topology: Connectedness And Separation*. DPH mathematics series. Discovery Publishing House, 2006.

[25] M. Thies and H. Spiecker. Evaluation and future prospects of terrestrial laser scanning for standardized forest inventories. *Forest*, 2(2.2):1, 2004.

[26] M. Vastaranta, M. Holopainen, H. Kaartinen, H. Hyyppä, and J. Hyyppä. Uudistuneet metsien maastomittaustarpeet. *Metsätieteen aikakauskirja*, 4(2009):370–374, 2009.

[27] L. Wasserman. *All of statistics: a concise course in statistical inference*. Springer texts in statistics. Springer, 2004.

[28] S. Willard. *General Topology*. Dover books on mathematics. Dover Publications, 2004.

[29] J.R. Wolberg. *Data analysis using the method of least squares: extracting the most information from experiments*. Springer, 2006.