Constraint-based automated reconstruction of grape bunches from 3D range data for high-throughput phenotyping

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von
Jennifer Mack
aus
Dieburg

Bonn, 2019

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn.

Erstgutachter: PD Dr. Volker Steinhage, Bonn
Zweitgutachter: Prof. Dr. Jürgen Gall, Bonn
Tag der Promotion: 23.09.2019
Erscheinungsjahr: 2019

Abstract

With increasing global population, the resources for agriculture required to feed the growing number of people are becoming scarce. Estimates expect that by 2050, 60 % more food will be necessary. Nowadays, 70 % of fresh water is used by agriculture and experts see no potential for new land to use for crop plants. This means that existing land has to be used efficiently in a sustainable way. To support this, plant breeders aim at the improvement of yield, quality, disease-resistance, and other important characteristics of the crops.

Reports show that grapevine cultivation uses more than three times of the amount of fungicides than the cultivation of fruit trees or vegetables. This is caused by grapevine being prone to various fungal diseases and pests that quickly spread over fields. A loose grape bunch architecture is one of the most important physical barriers that make the establishment of a fungal infection less likely.

The grape bunch architecture is mostly defined by the inner stem skeleton. The phenotyping of grape bunches refers to the measurement of the phenotypes, i.e., the observable traits of a plant, like the diameter of berries or the lengths of stems. Because of their perishable nature, grape bunches have to be processed in a relatively short time. On the other hand, genetic analyses require data from a large number of them. Manual phenotyping is error-prone and highly labor- and time-intensive, yielding the need for automated, high-throughput methods.

The objective of this thesis is to develop a completely automated pipeline that gets as input a 3D pointcloud showing a grape bunch and computes a 3D reconstruction of the complete grape bunch, including the inner stem skeleton. The result is a 3D estimation of the grape bunch that represents not only dimensions (e.g. berry diameters) or statistics (e.g. the number of berries), but the geometry and topology as well. All architectural (i.e., geometrical and topological) traits can be derived from this complete 3D reconstruction. We aim at high-throughput phenotyping by automatizing all steps and removing any requirement for interaction with the user, while still providing an interface for a detailed visualization and possible adjustments of the parameters.

There are several challenges to this task: ripe grape bunches are subject to a high amount of self-occlusion, rendering a direct reconstruction of the stem skeleton impossible. The stem skeleton structure is complex, thus, the manual creation of training data is hard. We aim at a cross-cultivation approach and there is high variability between cultivars and even between grape bunches of the same cultivar. Thus, we cannot rely on statistical distributions for single plant organ dimensions.

We employ geometrical and topological constraints to meet the challenge of cross-cultivar optimization and foster efficient sampling of infinitely large hypotheses spaces, resulting in Pearson correlation coefficients between 0.7 and 0.9 for established traits traditionally used by breeders. The active working time is reduced by a factor of 12.

We evaluate the pipeline for the application on scans taken in a lab environment and in the field.

Überblick

Angesichts einer zunehmenden Weltbevölkerung werden die landwirtschaftlichen Ressourcen zur Ernährung der wachsenden Anzahl von Menschen knapp. Schätzungen gehen von einem zusätzlichen Nahrungsbedarf von 60 % bis zum Jahr 2050 aus. Bereits heute werden 70 % des Frischwasservorrats durch Agrikultur verbraucht und Experten sehen keine Verfügbarkeiten für potentielle weitere Ackerflächen. Bereits existierende Felder müssen daher effizient und nachhaltig genutzt werden. Pflanzenzüchter arbeiten an der Verbesserung von Ertrag, Qualität, Resistenz gegen Krankheiten, sowie anderen wichtigen Charakteristiken von Nutzpflanzen.

Untersuchungen zeigen, dass im Weinbau mehr als dreimal so viel Fungizide benutzt werden wie im Anbau von Obstbäumen oder Gemüse. Der Grund dafür ist, dass Weintrauben anfällig gegen verschiedene Pilzkrankheiten und Schädlinge sind, die sich auf Feldern rasch ausbreiten können. Für Weintrauben ist eine lockere Architektur eine der wichtigsten physischen Barrieren, die den Pilzbefall unwahrscheinlicher macht.

Die Weintraubenarchitektur wird hauptsächlich vom inneren Stielgerüst definiert. Die Phänotypisierung von Weintrauben bezieht sich dabei auf die Vermessung der Phänotypen, d.h., der beobachtbaren Merkmale einer Pflanze, wie z.B. den Durchmesser einer Beere, oder die Länge von Stielen. Als verderbliche Ware müssen Weintrauben in relativ kurzer Zeit verarbeitet werden, genetische Analysen erfordern aber Daten von einer großen Anzahl verschiedener Trauben. Die manuelle Phänotypisierung ist fehleranfällig und erfordert einen hohen Zeit- und Arbeitsaufwand, daher ist die Entwicklung einer automatisierten Methode, die mit hohem Durchsatz betrieben werden kann, notwendig.

Das Ziel dieser Arbeit ist es, eine automatisierte Pipeline zu entwickeln, die eine Weintraube aus einer 3D Punktwolke vollständig rekonstruiert, inklusive des inneren Stielgerüsts. Das Resultat ist eine 3D Schätzung der Traube, die nicht nur die Dimensionen (z.B. Beerendurchmesser) oder Statistiken (z.B. die Anzahl der Beeren) repräsentiert, sondern auch die Geometrie und Topologie. Alle strukturellen (d.h. geometrischen und topologischen) Merkmale können aus dieser 3D Rekonstruktion abgeleitet werden. Um einen hohen Durchsatz zu erreichen, automatisieren wir alle Schritte und verzichten auf Interaktionen mit dem Benutzer, geben diesem aber über eine Schnittstelle die Möglichkeit, eine detaillierte Visualisierung zu erhalten und, wenn nötig, Parameter anzupassen.

Die Aufgabe bietet mehrere Herausforderungen: Reife Weintrauben weisen einen hohen Grad an Selbstverdeckung auf, daher kann das innere Stielgerüst nicht direkt aus der Punktwolke rekonstruiert werden. Die Struktur des Stielgerüsts ist komplex, was das manuelle Erstellen von Trainingsdaten erschwert. Der Ansatz soll für verschiedene Sorten geeignet sein. Zwischen verschiedenen Sorten und sogar zwischen verschiedenen Trauben der gleichen Sorte besteht eine hohe Variabilität. Daher können wir nicht auf statistische Verteilungen über die Dimensionen einzelner Pflanzenorgane zurückgreifen.

Wir verwenden geometrische und topologische Constraints, um die Methode für verschiedene

Sorten verwendbar zu machen und um effizientes Sampling für unendlich große Hypothesenräume zu erreichen, resultierend in Pearson Korrelationskoeffizienten zwischen 0.7 und 0.9 für traditionell von Züchtern benutzte Merkmale. Die aktive Arbeitszeit wird um den Faktor 12 reduziert.
Wir evaluieren die Pipeline auf Labor- und Feldaufnahmen.

Personal Information

Jennifer Mack
An der Josefshöhe 35
53117 Bonn

mack@cs.uni-bonn.de

Education

since 2016	Rheinische Friedrich-Wilhelms-Universität Bonn, PhD Student
2014	Rheinische Friedrich-Wilhelms-Universität Bonn, Master in Computer Science
2011	Rheinische Friedrich-Wilhelms-Universität Bonn, Bachelor in Computer Science
2008-2014	Rheinische Friedrich-Wilhelms-Universität Bonn, Studies of Computer Science
2008	Nicolaus-Cusanus-Gymnasium Bonn, Abitur

Acknowledgments

At first, I want to thank PD. Dr. Volker Steinhage, who gave me the chance to work on this project, supporting me on every step and offering new insights whenever required.

Also, I would like to express my gratitude to the members of my promotion commission: PD. Dr. Volker Steinhage, Prof. Dr. Jürgen Gall, Prof. Dr. Andreas Weber, and Prof. Dr. Reinhard Töpfer.

I would also like to thank the Deutsche Forschungsgemeinschaft, who funded this project titled ”Automated Evaluation and Comparison of Grapevine Genotypes by means of Grape Cluster Architecture” [funding codes STE 806/2-1, TO 152/6-1]. I want to express my gratitude to our colleagues and project partners at the Julius Kühn-Institut Geilweilerhof in Siebeldingen, Florian Rist, M.Sc., Dr. Katja Herzog, and the director of the Julius Kühn-Institut Prof. Dr. Reinhard Töpfer, for providing us with the 3D scans and the reference data used in this thesis, and offering invaluable insights from the user perspective. It was a great experience working with you all.

Another thank you goes to the students that I was able to (co-)supervise in the context of their Labs, Bachelor’s-, or Master’s theses, especially Christian Lenz, M.Sc., Johannes Teutrine, M.Sc., Lilli Bruckschen, M.Sc., Anatina Trakowski, B.Sc., Frank Schindler, B.Sc., Dirk Peters, M.Sc., Bernhard Japes, M.Sc., and Anne Hempel. Many times, their work helped further my own research.

I thank my colleague Pierre Barré, B.Sc., for fruitful discussions and providing the chance to rant about what went wrong whenever necessary.

I also want to thank Sabrina Amft, M.Sc., Lilli Bruckschen, M.Sc., Anne Hempel, Melanie Larws, B.Sc., Hoang Nguyen, B.Sc., and Dipl.-Übers. Helga Rumberg for proofreading my thesis.

A heartfelt thank you to my friends, especially Eva Hunger and Anne Hempel, my mother, sister, and my grandparents, who all supported me one way or the other, and to Lilli Bruckschen, who pushed me to follow my path into research.

Contents

Chapter 1

Introduction

For a growing world population, more and more resources are required to provide a healthy diet for all people. Estimates by the Food and Agricultural Organization of the United Nations (UN FAO) expect the population to rise to 9.7 billion by 2050, leading to an increase in the demand for food by 60 %. But resources for agriculture are already at their limits: 70 % of fresh water are currently used by agriculture, meaning that in 2050, 120 % more water would be required and 42 % more crop land, where there is not only no new land available today, but it is likely that land area will shrink due to rising sea levels and urbanization (Bhunnoo, 2018). While pesticides can be used to lessen yield shrinkage, they have a strong impact on the environment. Additionally, some species of fungi and weeds develop resistances, rendering the pesticides useless.

The objectives of plant breeders in this context are to improve the yield, quality, disease-resistance, drought and frost tolerance, and other important characteristics of the crops, to make it possible to cultivate them in a more efficient, but at the same time sustainable way.

A report on the use of plant protection products in the European Union (Muthmann, 2007) found that grapevine cultivation uses more than three times of the amount of fungicides than the cultivation of fruit trees or vegetables. This is caused by grapevine being prone to various fungal diseases and pests that spread quickly over fields. Especially bunch rot (Botrytis infection) is considered a high-risk fungus, as it is adaptive and builds up resistances against fungicides. A loose grape bunch architecture acts as a physical barrier that makes the establishment of a Botrytis infection less likely, making the breeding of grape bunches such that they show a loose architecture an important goal.

The architecture of a grape bunch is mostly defined by its inner stem skeleton. Depending on its structure, the berries are pressed together during growth or hang loosely. This effect is called the *bunch compactness*. Quantitative trait loci (QTLs) are sections of DNA correlating with quantitative traits in phenotypes of a grape bunch. Several QTLs responsible for grape bunch compactness were identified and have to be fine mapped based on phenotypes derived from grape bunches (Töpfer et al., 2011).

In this context, phenotypes are defined as the observable morphological attributes of a grape bunch, like the diameter of berries or the lengths of stems. Phenotyping is currently done manually by visually estimating attributes in regards to the OIV list (Organisation Internationale de la Vigne et du Vin, 2009). But QTL mapping requires data from a large number of grape bunches, as statistics get more robust with increasing example quantity. The manual phenotyping of grape bunches is highly labor- and time intensive (the so-called *phenotyping bottleneck*, (Furbank and Tester, 2011)), and additionally, subjective. Results can vary between different experts. Another problem is that grape bunches are a perishable product, meaning they have to be processed in a relatively short time. This contradicts the requirement of a large number of grape bunches, as only so many grape bunches can be processed during that time frame. Automated, high-throughput methods provide a way to, on the one hand, reduce the required processing time and, on the other hand, yield objective measurements. This is the objective followed in the project "Automated Evaluation and Comparison of Grapevine Genotypes by means of Grape Cluster Architecture" funded by the Deutsche Forschungsgemeinschaft (DFG) that this thesis is part of.

The development stages of grape bunches are described according to the Biologische Bundesanstalt, Bundesssortenamt and Chemische Industrie (BBCH) scale (Lorenz et al., 1994). We work on grape bunches of development stages BBCH 87 to 89. At this point, the grape bunches are ripe for harvest and all traits are fully developed.

Objectives: This thesis aims at the development of a completely automated pipeline that is applicable in the practice of phenotyping in grapevine breeding. It gets as input a 3D pointcloud derived by scanning a grape bunch from all sides. Based on this, a reconstruction of the complete grape bunch, including the inner stem skeleton, is computed. The reconstruction consists of geometric primitives approximating the plant organ shapes, like spheres, ellipsoids, and cylinders, allowing the derivation of established phenotypes.

Challenges: The reconstruction of grape bunches from 3D pointclouds presents several challenges.

1. The grape bunch stem skeleton has a complex structure, making it hard or even impossible to manually create training data in 3D pointclouds. Therefore, self-learning approaches are currently not applicable.
2. Grape bunches are subject to a high amount of self-occlusion, meaning that if grape bunches are scanned from the outside, only the berries and the upper part of the main stem (the peduncle) are visible. Therefore, the stem skeleton cannot be reconstructed directly from the data.

3. A cross-cultivar optimization cannot rely on statistical distributions derived for single plant organ dimensions or parameters, like, e.g., the diameters of berries, as there is a high variability between cultivars and in some cases even for grape bunches of the same cultivar. Additionally, the topology of the grape bunch and the relations between different plant organs have to be taken into account.
4. Different grape bunch organs have to be represented by different geometric primitives, e.g., the berries can be represented by spheres or ellipsoids, but the shape of stems is more similar to cylinders. To foster automated and reliable estimation of these primitives, it is important to use semantic labeling, i.e., to assign the points of the pointcloud to the respective grape bunch organs.

1.1 Contributions

This thesis meets the challenges named before using the following approaches:

1. We employ a model-based approach that incorporates constraint-based knowledge about the overall structure of the grape bunch.
2. Due to the massive occlusion of the inner stem skeleton, we use a Markov-Chain-Monte-Carlo approach for the selection of the correct model and the estimation of the parameters of the plant organs.
3. For the optimization, the structure of the grape bunch is represented by constraints that encode geometrical and topological relations between grape bunch organs.
4. We present and evaluate supervised and unsupervised methods for semantic labeling of dense high-precision pointclouds of grape bunches from different cultivars.

Like most parts of a plant, a grape bunch follows a regular growing structure. On this basis, it is possible to develop an approach that derives a first guess for the model underlying a given input and optimize it based on the constraints known to hold for the grape bunch. This first guess will be called ”initial hypothesis” in the following.

The inner structure of the grape bunch is unknown as it is completely occluded at development stages BBCH 87 to 89. The model dimension is, therefore, also unknown and has to be optimized together with the parameters of the model parts. For this, we employ a Reversible-Jump-Markov-Chain-Monte-Carlo (RJMCMC) optimization. In this thesis, we propose an adaption of the algorithm dividing it into phases. Simulated Annealing is used to foster efficient optimization. During approximately the first half of the optimization, measured in regards to the temperature, jumps that change the dimension of the model are followed by jumps optimizing the parameters involved in the jump. The second phase is restricted to a parameter optimization.

To include information about the topology of grape bunches in the optimization, we employ geometrical and topological constraints.

For semantic labeling, we derive a new RGB-D feature descriptor based on a combination of the Fast-Point-Feature Histograms (FPFHs) and a histogram derived from the hue channel of the HSI color space, representing shape and color features, respectively. We call this the Hue-Fast-Point-Feature-Histograms (HFPFHs). Due to its definition, the hue channel is well suited to applications where the classes contain objects with different shades of one color. We evaluate our new descriptor combined with a Support Vector Machine (SVM) for supervised learning and k-means for unsupervised learning and compare the results to established variants like the Point-Feature Histograms with RGB (PFHRGB) (Rusu et al., 2008) and Color-Signature of Histograms of Orientations (CSHOT) (Tombari et al., 2011) in the context of plant phenotyping.

Methods for the semantic labeling of 3D pointclouds based on deep learning have become an alternative to traditional descriptors during the last years, outperforming them in various applications. We choose to use a multi-view approach, as they are able to deal with dense pointclouds of objects with complex structures, like grape bunch stem skeletons. More concretely, we adjusted and optimized the SnapNet (Boulch et al., 2018).

1.2 Related work

In this section, we introduce related work concerning the phenotyping of grape bunches. For related work regarding the methods used in this thesis, please refer to sections 3.1 and 4.2 in the respective chapters.

We divide the proposed approaches into those working with 3D data and those using images, presenting the strengths and weaknesses of both approaches. Tello et al. (2016) compare methods using either 2D or 3D image analysis methods and evaluate them with respect to the morphological volume of grape bunches.

1.2.1 Phenotyping based on images

Grapevine berries: Many studies concentrate on the grapevine berries as important attribute for yield estimation. The corresponding images are taken at night with special lighting conditions (Font et al., 2014, Millan et al., 2018), in the field (Nuske et al., 2014, Roscher et al., 2014), with a rebuilt harvesting machine called PHENObot (Kicherer et al., 2015), or even with a smartphone (Aquino et al., 2017). Liu et al. (2015) take single images from one side of a grape bunch, reconstruct the berries and extrapolate the number of berries of the whole grape bunch from this value. Other studies derive the weight of berries (Cubero et al., 2014) or grape bunches (Diago et al., 2015, Liu et al., 2013). Newer approaches employ deep learning to detect grape bunches in images and count the berries with circular Hough transform (Rudolph et al., 2018), or even end-to-end learning to detect young shoots, berries, or pedicles (Grimm et al., 2019). Another approach using traditional descriptors and unsupervised learning to detect shoots in videos was introduced by Liu et al. (2017).

Yield estimation: Some studies aim at yield estimation at earlier development stages to provide the information before the beginning of the harvest. Aquino et al. (2018a) present a smartphone application to count young berries, using a black box as background, while Aquino et al. (2018b) take images with an all-terrain vehicle in the field. The same group (Aquino et al., 2015a) introduced a smartphone application that allows the user to take images in the field and counts the number of grapevine flowers in this image. Another work by Diago et al. (2014) does the same for images taken with a camera in the field, while Aquino et al. (2015b) provide a method to segment grapevine inflorescences. Liu et al. (2018) capture images in the field with a black backing board that is used to provide a controlled background.

Compactness: A study by Cubero et al. (2015) directly derives the bunch compactness by calculating the width of the bunch at different heights. A review over grape bunch compactness and different methods to derive it is given by Tello and Ibáñez (2017). Those methods compute indexes based on the visible outside of the grape bunch, using ratios between volume estimations, convex hulls, and the number of berries. None of those methods explicitly reconstructs the stem skeleton to include in the derivation of the compactness.

1.2.2 Phenotyping based on 3D data

There are comparatively few approaches for the phenotyping of grape bunches working on 3D data. Klodt et al. (2015) use RGB cameras and dense stereo reconstruction to derive traits like the 3D leaf surface area, or the fruit-to-leave ratio for two grape bunch breeding lines. Ivorra et al. (2015) derive a 3D model of a grape bunch based on stereo vision. This model yields traits like the number of berries, their sizes, weight, and the grape bunch compactness. Another study employs the Kinect to derive grape bunch volume and mass (Marinello et al., 2016).

Rose et al. (2016) generate a pointcloud using a multi-view stereo approach based on images with geotags. They classify the pointcloud and derive the berries with a Random Sample Consensus (RANSAC)-based approach, yielding the number of grape bunches, the number of berries, and the berry sizes.

A recent study by Milella et al. (2019) employs an Intel RealSense R200 sensor to detect and count grape bunches and estimate their weight.

To the best of our knowledge, the only publication that yields a complete derivation of the grape bunch, including the stem skeleton, is by Schöler and Steinhage (2015). They introduce a prototypic approach that uses statistical constraints to optimize a model of the grape bunch using RJMCMC optimization. As they test their method on only four grape bunches of the same cultivar that were chosen to be rather similar to each other, statistical constraints work well in their application. However, as mentioned before, even within the same cultivar grape bunches show a high within-class-scatter, rendering statistical constraints unusable in this application. Therefore, we use geometrical and topological constraints.

1.3 Conceptual design of the workflow

The pipeline, as depicted in Figure 1.1, consists of the following steps:

- Scanning and preprocessing: the grape bunch is scanned. The resulting dense high-precision pointcloud is thinned and outlier are removed to prepare it for further processing.
- Semantic labeling: each point of the pointcloud is assigned a label based on the biological structure of the grape bunch. In this context we introduce the new HFPFH descriptor combining FPFHs (Rusu et al., 2009) with a color descriptor (Mack et al., 2018). The results are compared to those achieved with a deep learning approach, more concretely, the so-called SnapNet (Boulch et al., 2018).
- Initial grape hypothesis: based on the semantic labeling, the berries of the grape bunch and the peduncle are reconstructed directly from the pointcloud using a Random

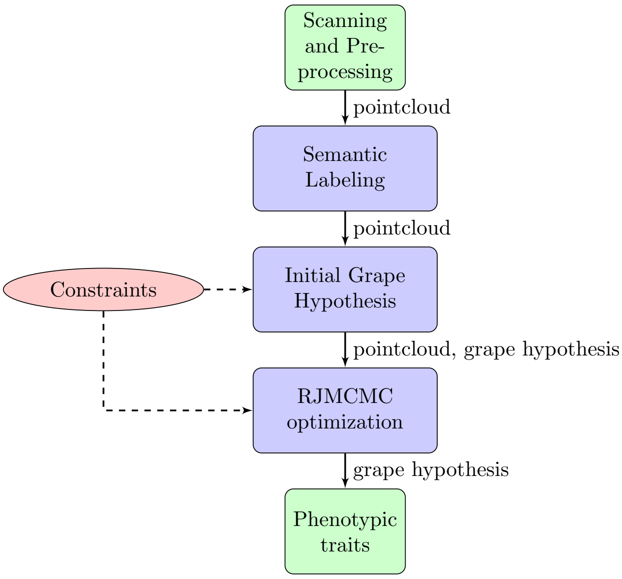


Figure 1.1: *The workflow of the proposed pipeline.*

Sample Consensus (RANSAC)-based procedure (Mack et al. (2017a), Mack et al. (2018), Rist et al. (2018)). Subsequently, an initial hypothesis for the complete grape bunch is created based on geometrical and topological constraints.

- RJMCMC optimization: the initial hypothesis is refined, employing geometrical and topological constraints to assess the quality of a new hypothesis compared to the current one (Mack et al. (2019)). The process is divided into two phases: during the first phase, each jump that changes the dimension of the model is followed by n jumps that perform a local optimization on the parts involved in the jump. This leads to a better adjustment of the parts and, thus, to a higher acceptance rate for topological jumps. The second phase assumes that the correct model has been chosen during the first phase. From this point on, only jumps are allowed that optimize the parameters of the model, but the model dimension is not changed.
- Phenotypic traits: from the 3D reconstruction, phenotypic traits can be derived, like berry diameters, or lengths of stems.

1.4 Thesis structure

The thesis is divided into seven chapters. Chapters 2 to 5 explain the algorithm on the basis of scans taken in a lab environment, while Chapter 6 shows first results for the application on field scans.
Chapter 2 describes the process of scanning the grape bunches, the preprocessing of the 3D pointclouds, and the derivation of reference data.
Chapter 3 presents the first step of a semantic labeling of the pointcloud in regards to the part of the grape bunch each point belongs to. For this, we evaluate the usage of the newly defined HFPFHs combined with supervised and unsupervised classifiers, and a Convolutional Neural Network (CNN) as well. The respective contributions were published in Mack et al. (2017a), Mack et al. (2017b), Mack et al. (2018), and Japes et al. (2018).
Afterwards, Chapter 4 introduces the constraint-based 3D object reconstruction approach we applied to automatically derive reconstructions from labeled 3D pointclouds of grape bunches. The method relies on constraints to evaluate the quality of a new hypothesis in comparison to the current hypothesis. A part of the method and its results was already presented in Mack et al. (2017a), Rist et al. (2018) and Mack et al. (2019).
In Chapter 5 we evaluate the methods presented so far on scans taken in a lab environment and discuss the results as well as the applicability to efficient phenotyping.
We apply the approach to field scans in Chapter 6, presenting first results and discussing them in comparison to the scans created in a lab environment.
Finally, Chapter 7 concludes the thesis and discusses possible extensions and enhancements of the developed methods.

Chapter 2

Data

This chapter covers the data generation and preparation process as well as the derivation of reference data to use for comparison in the evaluation. The first part presents the required nomenclature from viticulture. Afterwards, we describe the scanning process used to create dense high-precision 3D pointclouds that show the grape bunch from all sides. The next section introduces the preprocessing we apply to the pointcloud, consisting of a thinning procedure and a removal of outlier points. Finally, we describe the approach to derive different kinds of reference data for comparing the 3D reconstructions achieved with our optimization method with the original grape bunches. Both scanning and derivation of the reference data were carried out at the Institute for Grapevine Breeding Geilweilerhof in Siebeldingen.

2.1 Nomenclature from viticulture

We group the grape bunch organs according to the nomenclature commonly used in viticulture, as shown in Figure 2.1:

Peduncle: The peduncle is the former connection between the rachis of the grape bunch and the canopy.

Rachis: The rachis is the main stem of the grape bunch.

Twig: Twigs are the extensions of the rachis in the direction of the pedicles.

Subtwigs: Subtwigs are small twigs growing from twigs itself instead of the rachis. They can also be called ”second-order twigs”.

Pedicle: Pedicles are stalks that connect a berry to a twig or subtwig.

Berry: Berries are the fruits of a grape bunch.

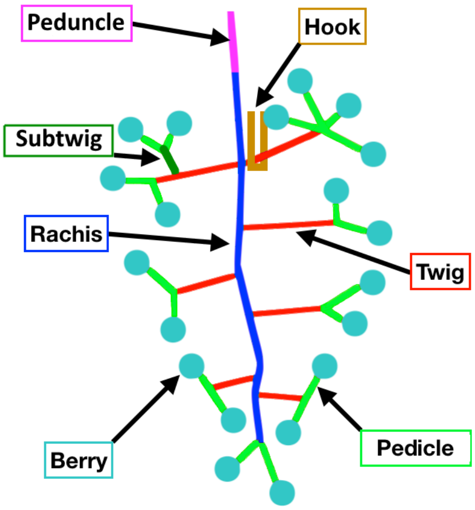


Figure 2.1: *The parts of a grape bunch including its stem skeleton labeled in different colors (the peduncle in pink, the rachis in blue, the twigs in red, the subtwigs in darkgreen, the pedicles in green, and the berries in turquoise). Additionally, the hook is shown with a brown color. To keep the image simple, berries are depicted smaller than they would be at development stage BBCH 89. This figure is an adjusted version from Japes et al. (2018).*

During the scanning process, the grape bunch is attached to a hook that is depicted in Figure 2.1. While it is not part of the grape bunch itself, we have to include it in the semantic labeling process to correctly recognize the respective points.
The transition from peduncle to rachis is not clearly defined in the literature. We choose to name the part of the main stem visible above the berries peduncle and the main stem below rachis. While the peduncle is usually visible in the scans of the grape bunch, as seen, e.g., in Figures 2.6a to 2.6c, the rachis is mostly occluded.

2.2 Scanning process

For each grape bunch, the scanning process can be divided into the following steps:

1. scanning of the complete grape bunch,
2. removal of berries,
3. taking a reference photo of the stem skeleton,
4. second scanning of the stem skeleton.

In the following, we describe those steps in more detail. It has to be noted that not all steps were performed on all grape bunches, as will be remarked in the description of the data sets.

Figure 2.2: *During the scanning process, the Artec 3D Spider Scanner (Artec 3D, 2019a) is placed in front of a grape bunch that is attached to a rotating hook. The grape bunch is scanned for slightly more than a full rotation period, generating a pointcloud that shows the grape bunch from all sides.*

(a) *Calardis Blanc* ***(b)*** *Dornfelder* ***(c)*** *Pinot Noir* ***(d)*** *Riesling*

Figure 2.3: *Examples for the cultivar set used as data material.*

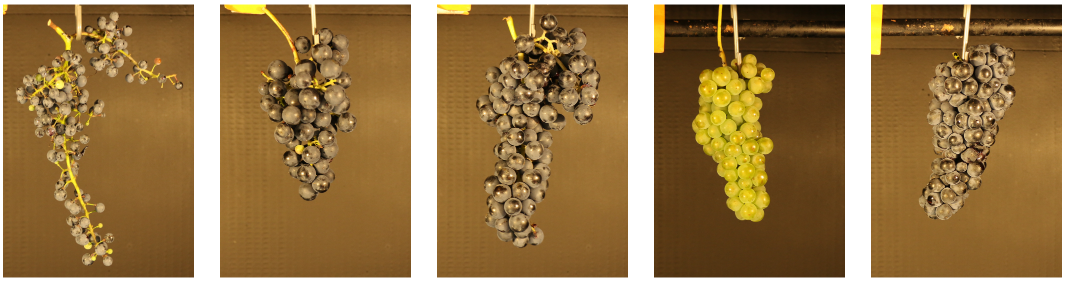
In a lab environment, the grape bunch is attached to a hook that rotates automatically with a relatively slow, uniform speed. The Artec 3D Spider Scanner (Artec 3D, 2019a) is placed in front of the rotating grape bunch, as shown in Figure 2.2. The bunch is scanned for slightly more than a full rotation period, generating a pointcloud that contains scans from all sides. Those scan slices are aligned on the fly by the Artec Scanning Software, generating a 3D pointcloud with a resolution of 0.1 mm and an accuracy of up to 0.05 mm.

After the first scanning, all berries are removed from the grape bunch. A photo of the resulting stem skeleton is taken to provide a basis for the reference data, as explained in Section 2.4. Then, it is attached to the hook again and scanned for a full rotation period, generating 3D pointclouds showing only the stem structure.

Together with the Institue for Grapevine Breeding Geilweilerhof in Siebeldingen, a number of 52 grape bunches of four cultivars (Calardis Blanc (9), Dornfelder (15), Pinot Noir (19), and Riesling (9)) were chosen for the initial screening such that they cover different grape

(a) *Sultana* **(b)** *Prime* **(c)** *Sugrathirteen* **(d)** *Ruby Seedless*

Figure 2.4: *Examples for the table grape set used as data material.*



(a) *Class 1* **(b)** *Class 3* **(c)** *Class 5* **(d)** *Class 7* **(e)** *Class 9*

Figure 2.5: *Examples for the variety set consisting of grape bunches from the five compactness classes.*

bunch shapes, sizes, and berry colors. Examples can be seen in Figure 2.3 and the resulting pointclouds in Figure 2.6. We call this the *cultivar set*.

To cover a variety of berry shapes and colors, steps 1 and 2 were done for 11 table grapes, including 5 grape bunches with green berries (1 Sultana, 2 Sugraone, 2 Prime) and 6 grape bunches from red table grape cultivars (3 Sugrathirteen, 3 Ruby Seedless). As can be seen in Figure 2.4 the ellipticity of the berries varies strongly. We will further call this the *table grape set*.

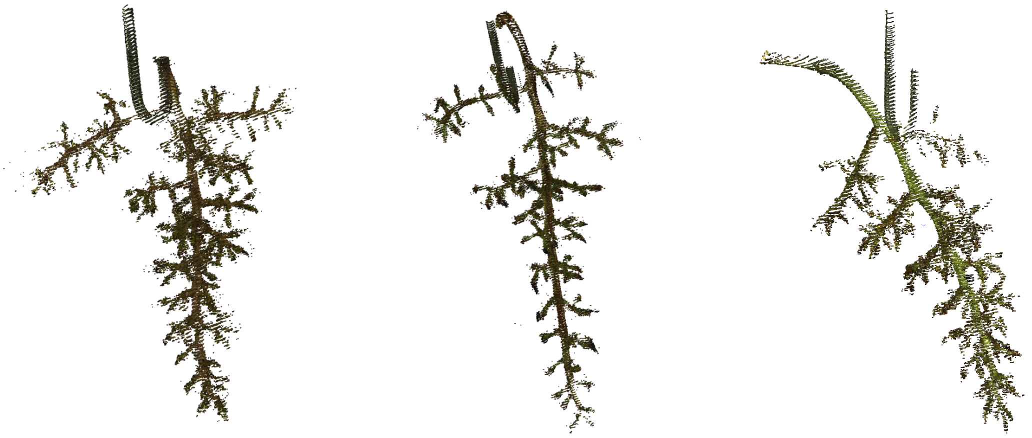
Finally, steps 1 to 4 were also applied to 10 grape bunches from a F1 progeny[1] of the crossing population of GF.GA-47-42 x ”Villard Blanc” that were chosen to represent the compactness classes from very tight to very loose, as shown in Figure 2.5. This will be called the *variety set* in the following.

All grape bunches are of development stages between BBCH 87 and 89, meaning that the berries are fully grown and ripe for harvest. At these stages, the stem skeleton is visible only in very loose grape bunches, as can be seen in Figure 2.5a, and even there parts of it are

---

[1] First generation of offspring

(a) *Calardis Blanc* ***(b)*** *Dornfelder* ***(c)*** *Pinot Noir*



(d) *Calardis Blanc* ***(e)*** *Dornfelder* ***(f)*** *Pinot Noir*

Figure 2.6: *Example pointclouds from grape bunches and grape bunch stem skeletons after preprocessing with outlier removal and thinning, as introduced in Section 2.3.*

occluded.

2.3 Preprocessing of pointclouds: outlier removal and thinning

The current pointclouds contain noise resulting from erroneous measurements during the scanning process, as can be seen in Figure 2.7a. To reduce this effect, we apply an approach to remove statistical outliers from the data: for each point, the distance to its k nearest neighbors is computed. Due to the nature of the scanning process, the resulting distribution

(a) *Original pointcloud* ***(b)*** *Pointcloud after outlier removal* ***(c)*** *Pointcloud after thinning*

Figure 2.7: *Example for the preprocessing of the Calardis Blanc grape bunch shown in Figure 2.3a.*

can be assumed to be Gaussian. Based on these distributions, a global mean and standard deviation for the distances can be computed. If the mean neighbor distance of a point lies outside the interval made up of the global mean distance and the global standard deviation, we consider it an outlier and remove it from the data.

After the outlier removal the pointclouds usually still have a size of between 300 K and 1 million points, often positioned very close to each other, as they originate from different scan slices. Thinning the pointcloud leads to an even spreading of the points over the surfaces. This increases the usability of a pointcloud, as reasonable values for parameters like the number of inliers for RANSAC in Section 4.3.4 can be predicted. Additionally, the reduction of the points leads to an overall speed-up of the process.

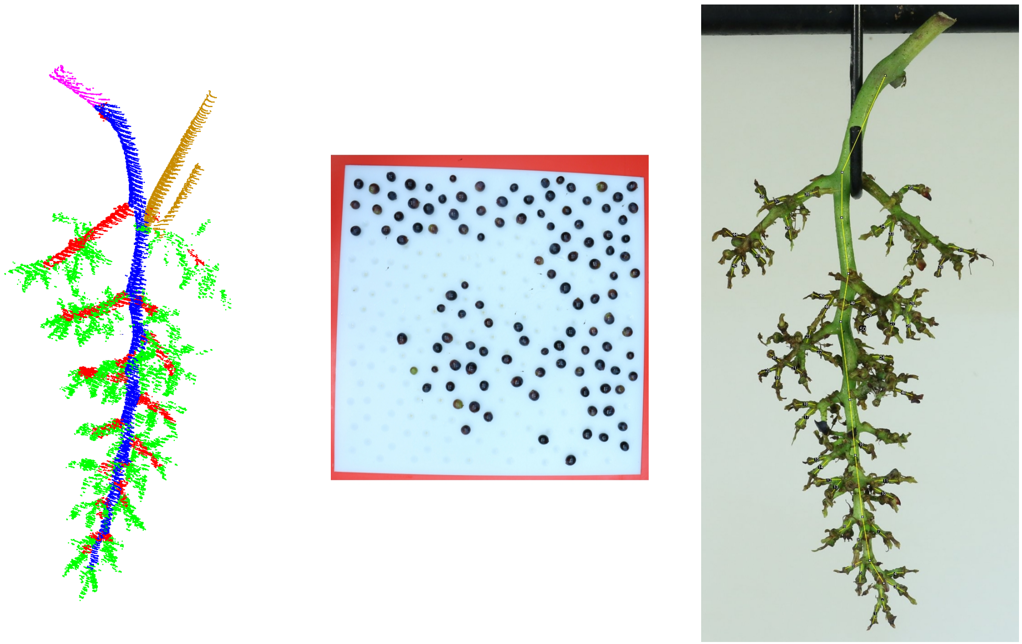
We apply a voxel-based thinning to the pointcloud that works as follows: the 3D pointcloud is divided into a so-called voxelgrid, where the edge length of each voxel, represented by the parameter $v \in \mathbb{R}$, is fixed to 0.4 mm. For each voxel, the center of mass of all points inside is computed and used to represent all points. This reduces the size of the pointclouds to between 150 K and 300 K.

Figure 2.7 shows the process for an exemplary Calardis Blanc grape bunch: the original pointcloud in Figure 2.7a shows outliers resulting from errors occurring during the scanning process that are removed in Figure 2.7b. At this time, the pointcloud size has been reduced from originally 350 K to 300 K. In the next step, the redundant points are removed, resulting in the pointcloud shown in Figure 2.7c with 100 K points. Visually, there is almost no difference between the pointclouds before and after thinning. The parameter v is chosen empirically. It was shown in Mack et al. (2017a) that adjusting this value in a small range

does not significantly change the result.
Figure 2.6 shows further examples for pointclouds after preprocessing.

2.4 Reference data and gold standards

We derive labeling gold standards and statistical reference data.
For the evaluation of the semantic labeling methods, labeling gold standards as shown in Figure 2.8a are created by manually labeling all points in the pointcloud following the stem skeleton structure introduced in Section 2.1, with subtwigs and pedicles summarized into one class, as they are hard to distinguish in the data. All in all, we provide labeling gold standards for 8 Calardis Blanc grape bunch scans, 10 Dornfelder scans, and 10 Pinot Noir Scans, each split up evenly between scans of the grape bunch and of the stem skeleton. Additionally, 5 Riesling grape bunch scans were labeled. It was not possible to generate labeling gold standards for the stem skeleton of the Riesling grape bunches, as those scans included a large



(a) *Labeling gold standard* *(b)* *2D image of the berries spread on a perforated plate* *(c)* *2D image with marked lengths*

Figure 2.8: *Examples for the labeling gold standard (Figure2.8a) and the basis for the derivation of statistical reference data for the berries (Figure 2.8b), the stem, and the pedicles (Figure 2.8c) derived for the Pinot Noir grape bunch in Figure 2.3c. The colors of the labeling gold standard relate to those shown for the structure of the grape bunch in Figure 2.1.*

number of small berries that would have been highly elaborate to remove, but cover large parts of the stem skeleton.

We derive statistical reference data in the form of phenotypes known from the OIV list (Organisation Internationale de la Vigne et du Vin, 2009), especially the number of berries, their average diameter, the length of the stem, and the average length of the pedicles. Those traits are established under grapevine breeders. At the Institute for Grapevine Breeding Geilweilerhof, two methods are used for the derivation of those values: for the berry reference data, the berries are removed from the grape bunch and spread on a perforated plate, such that each berry lies in one of the holes, as shown in Figure 2.8b. The number and average diameter of berries are derived from an image of the plate as shown by Kicherer et al. (2013). The lengths of the stem and the pedicles are estimated by taking photographs of the stem skeletons. The required lengths are then manually marked in each image, as can be seen in Figure 2.8c. This is a tedious process that takes a proficient user on average about 12 minutes. Additionally, the approach uses a 2D image, which has several drawbacks: Only a relatively small number of pedicles is completely visible. Parts of the stem can also be occluded. Due to the curvature in the stem skeleton and pedicles growing in different directions lengths are distorted. We try to compensate for this by choosing a good angle for the photo and measuring as many pedicle lengths as possible to achieve a stable average, still, some divergence cannot be avoided.

An alternative would be to directly measure the lengths in the stem skeleton with a ruler. While this method would provide exact measurements, it has to be done with a fresh stem skeleton, as the drying process leads to a shortening of the stems, introducing more errors. As explained in Chapter 1, the grape bunches are perishable products and, as such, have to be processed in a relatively small time frame. Therefore, a tedious process like this cannot be included in the harvesting process.

As shown in Table 2.1, the *cultivar set* can be evaluated with respect to all kinds of labeling and reference data.

During harvesting the staff is highly occupied with routine work, therefore, we reduced the amount of reference data collected for the other data sets. For the *table grape set* we created

	Special properties	Labeling g.s.	Berry ref.	Stem ref.
cultivar set	Four cultivars	X	X	X
table grape set	Elliptical berries	partly	X	
variety set	Compactness classes			X

Table 2.1: *The properties of the data sets. For the* table grape set*, the available labeling gold standard (labeling G.S.) only differentiates between berries and stems, instead of berries, peduncle, rachis, twigs, subtwigs, and hook. The shortcut "Berry ref." stands for berry reference, and "Stem ref." for stem reference, respectively.*

labeling gold standards for all 11 grape bunches, but only divided the points into two classes, with the first class consisting of berry surface points, and the second of points on the surface of stems or of the hook. There is no reference data available for the stems, as the focus of this set is the elliptical shape and the different colors of the berries.

The *variety set* shows grape bunches from all compactness classes. As the compactness mostly depends on the inner stem skeleton, we only provide reference data for the stems.

Chapter 3

Semantic labeling of 3D pointclouds

The automated reconstruction of plant organs with geometric primitives requires assigning the points of the pointcloud to the respective plant organs, as different plant organs have to be represented by different primitives. E.g., berries can be represented by spheres, but stems share more similarities with cylinders. This process of labeling points with respect to the surface of the plant organ they belong to is called semantic labeling and is especially challenging in precision farming due to high within-class variances, high degrees of occlusions and overlaps, self-similarities, and densely packed plant organs. E.g., the shape of berries can differ depending on how much pressure is between them and the neighboring berries.

There are two approaches to the semantic labeling of 3D pointclouds: Using either descriptors, or deep learning with a Convolutional Neural Network (CNN).

In the first case, descriptors are hand-crafted based on information derived from the pointcloud for each point. Afterwards, either a supervised classifier is trained, or an unsupervised classification is used to assign a label to each point depending on its descriptor. The two parts (descriptor and classifier) can balance each other out, i.e., when using a powerful descriptor, a simple classifier can be sufficient to provide a reliable division into the classes. On the other hand, if the descriptor only covers basic features, a more sophisticated classifier might still be able to provide robust results.

A CNN is a supervised classifier that automatically derives the required descriptive features during training. This requires a large amount of training data and is computationally expensive. Still, the high quality of results and applicability to a large number of classes make this method extremely powerful and in many cases the de-facto state-of-the-art.

In the first part of this chapter we give an overview of the related work regarding traditional descriptors on the one hand and CNN-based deep learning on the other hand. In Section 3.2 we introduce the Fast-Point-Feature Histogram (FPFH) descriptor that we use in this application, its extension with color information, resulting in the Hue-Fast-Point-Feature

Histogram (HFPFH) descriptor, and the different classifiers. Finally, in Section 3.3 we explain the architecture of the SnapNet and how it has to be adjusted to be used on pointclouds of grape bunches.

3.1 Related work

We divide the related work into the two categories of approaches mentioned above, with the first covering traditional descriptors, and the second one deep learning methods.

3.1.1 Descriptor-based semantic labeling

For different applications, descriptors have to be defined focusing on different, sometimes even contradictory key aspects.
A traditional field is the so-called scan matching. A 3D scanner generates slices of points at a fixed interval with resolution and range depending on the scanner itself. Those slices have to be aligned to generate a pointcloud. While the position of the scanner can be used to support this process, most localization methods are not exact enough to provide a perfect positioning. Therefore, descriptors are used in the context of scan matching to yield a better alignment. The Signature of Histograms of Orientations (SHOT, Tombari et al. (2010)) is an example for this category. They match surfaces to each other based on histograms of normal orientations computed in a local reference frame. The extension Color-SHOT (CSHOT, Tombari et al. (2011)) additionally includes information about the texture around each point. Nascimento et al. (2012) show that their Binary Robust Appearance and Normals Descriptor (BRAND, Nascimento et al. (2012)) is able to outperform the CSHOT by encoding point information as a binary string with fast performance and low memory consumption. Wu et al. (2017b) introduce a specifically designed combination of a descriptor and keypoint detector using texture and shape information. The Rotational Projection Statistics (RoPS) descriptor (Guo et al., 2013) is based on a triangle mesh and computes statistical information about the distribution of the points after projecting them onto normed planes. It was ranked on top for some studies providing an overview over scan matching descriptors (Yang et al., 2016a,b). Rusu et al. (2009) defined the Fast-Point-Feature Histograms (FPFHs) as an extension of the Point-Feature Histograms (PFH, Rusu et al. (2008)). They represent the local surface properties of a point based on the geometric relation between it and its neighbors.
While for scan matching it is important to distinguish parts of the data that are similar to each other, but not equal, plant phenotyping requires assigning the same label to different instances of parts of the plant, like different berries. Due to the natural growth, those parts usually show variations in color and shape. PFHs were shown to be suitable in this context by Paulus et al. (2013). They used PFHs to classify pointclouds of grape bunches and crop plants into a combined class for leaves and stems, and fruits as the second part. With this

semantic labeling, the result can be used to estimate the expected yield.

In an extension of the study, Wahabzada et al. (2015) clustered pointclouds into the plant parts described above using k-means. With this unsupervised method, they still reached robust results.

In Mack et al. (2017b), we compared the FPFHs, PFHs, SHOT, and RoPS descriptors as well as Spin Images (Johnson and Hebert, 1999) on the application of plant phenotyping and found FPFHs to be the most suitable choice. Similar to the CSHOT (Tombari et al., 2011) or the PFHRGB (Rusu et al., 2008) we try to enrich the FPFHs using color information, as explained in Section 3.2.2.

3.1.2 Deep learning-based semantic labeling

During the last years, deep learning approaches using convolutional neural networks outperformed traditional descriptors in the context of object recognition and semantic labeling (e.g., Girshick et al. (2014)). Following this development, a large number of studies have proposed architectures to deal specifically with image segmentation. One example is the SegNet (Badrinarayanan et al., 2017) that consists of an encoder and a decoder network, followed by a layer providing a pixel-wise classification. The U-Net (Ronneberger et al., 2015) was originally intended for biomedical image segmentation and is able to work with relatively few training images.

For the semantic labeling of 3D pointclouds, there are three categories of approaches: point-based methods, voxel-based methods, and multi-view methods.

Point-based methods work directly on the pointcloud. One prominent example is the Point-Net (Qi et al., 2017a) using the point coordinates as well as local and global information about the points in a fully-connected segmentation and classification network. In an extension to their work, the Point-Net++ (Qi et al., 2017b) additionally uses a hierarchical network.

Voxel-based approaches build an occupancy grid to use as neighborhood structure (Maturana and Scherer, 2015). The U-Net has been adapted to work on 3D data instead (Çiçek et al., 2016), but it requires a high amount of parameters, making it infeasible for usage on large pointclouds. The 3D Shape-Nets (Wu et al., 2015) learn 3D shapes from CAD-data for 3D object recognition.

Multi-view approaches render images from 3D pointclouds and classify them with CNNs (Su et al., 2015). Based on this, Boulch et al. (2018) introduced the SnapNet, that renders images from randomly chosen views and uses the U-Net to classify the images, before applying a backprojection to the 3D pointcloud. Guerry et al. (2017) optimize the SnapNet for the application on RGB-D data in robotics.

Considering that point-based methods have difficulties dealing with complex objects, like grape bunch stem skeletons, and voxel-based methods are highly computationally expensive, making them less suitable for our high-precision data, we decide to use a multi-view approach, more concretely, the SnapNet (Boulch et al., 2018).

3.2 Semantic labeling using descriptors

We introduce a new so-called HFPFH descriptor, combining shape and color information. The results presented in Mack et al. (2017b) show that the FPFH is the most promising descriptor in the context of plant phenotyping. We add color information with the objective to enhance its descriptive power. As second step, a classifier has to be chosen. We test a Support Vector Machine (SVM, Vapnik (1995)) as supervised method and, additionally, apply k-means++ (Arthur and Vassilvitskii, 2007) for an unsupervised clustering.

3.2.1 Fast-Point-Feature Histograms

Fast-Point-Feature Histograms (FPFHs) derive shape information for each point based on the geometric relations between the point itself and its neighbors.
At first, the normal n_i associated with each point p_i is computed. This can be done, e.g., using a Principal Component Analysis (PCA) based on the neighboring points $\{p_0, ..., p_k\}$ in a radius r_n. A local reference frame is needed to make the calculation rotation invariant. Here, p_s is the point with smaller angle between its associated normal n_s and the line connecting it to the other point, i.e., if it holds for points p_i and p_j with $p_{ji} = p_j - p_i$ and $p_{ij} = p_i - p_j$ that

$$\arccos(n_i, p_{ji}) \leq \arccos(n_j, p_{ij}), \tag{3.1}$$

then $p_s = p_i$, $n_s = n_i$, $p_t = p_j$ and $n_t = n_j$ and the other way round. Based on this, the Darboux frame (u, v, w) can be derived as local coordinate system:

$$u = n_s \tag{3.2}$$

$$v = \left(\frac{p_t - p_s}{||p_t - p_s||_2} \times u \right) \tag{3.3}$$

$$w = u \times v \tag{3.4}$$

To make u, v, and w an orthonormal basis, v and w have to be normalized.
Three angular features are used to express the difference between the two normals:

$$\alpha = v \cdot n_t \tag{3.5}$$

$$\phi = u \cdot \frac{p_t - p_s}{||p_t - p_s||_2} \tag{3.6}$$

$$\theta = \arctan(w \cdot n_t, u \cdot n_t) \tag{3.7}$$

Those angular features are computed for each point p_q and its neighbors in a radius $r_h \in \mathbb{R}$. The results are binned into a histogram with a number of $b_s \in \mathbb{N}$ bins. This is called the Simplified-Point-Feature Histogram (SPFH).

To take into account the influence between the neighboring points in a radius r_h, the SPFHs are combined into the Fast-Point-Feature Histogram as follows:

$$\mathrm{FPFH}(p_q) = \mathrm{SPFH}(p_q) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{w_i}\cdot \mathrm{SPFH}(p_i), \tag{3.8}$$

with k being the number of neighboring points and w_i the distance between p_q and its neighbor p_i. Thus, for each point p_q the FPFH has an influence region with a radius of up to $2 \cdot r_h$, taking into account the relationships between the point and its neighbors and between their neighbors as well.

3.2.2 Hue-Fast-Point-Feature Histograms

The colors of berries of grape bunches vary depending on the cultivar, as seen for the different data sets in figures 2.3, 2.4, and 2.5. For cultivars with green berries, the colors of berries and stems can be almost identical, while for cultivars with red berries the difference in color can be sufficient to distinguish the berries from the stems. Based on this observation, we follow a method similar to Rusu et al. (2008) adjusted to FPFHs. The RGB channels are binned into a second histogram and added to the FPFHs.

Compared to RGB the HSI (Hue, Saturation, Intensity) color model offers a representation that is more similar to the human perception. Especially the hue channel is due to its definition very suitable to distinguish between classes that contain objects with different shading variants of the same color. Therefore, we derive the Hue, Saturation and Intensity values for each pixel and add them to the extended histogram.

Each channel is divided into a number of $b_c \in \mathbb{N}$ bins. The color histogram is then added to the FPFH defined above, giving the final descriptor $b = b_s + b_c$ bins.

As we assume that the hue channel has the most potential to achieve a robust classification, we define our new RGB-D descriptor to be the hue channel combined with the FPFH descriptor, and call this the Hue-Fast-Point-Feature Histogram (HFPFH) descriptor in the following.

3.2.3 Classification

After the creation of a descriptor, it is necessary to use a classifier to semantically label the points. This can be done with supervised or unsupervised methods.

Support-Vector-Machines. As first approach, we use a Support Vector Machine (SVM, Vapnik (1995)) with linear kernel. A binary SVM separates data from two classes by deriving the hyperplane between them that has the largest margin to the closest training examples from each class. More formally, assume we are given n tuples of the form (x_i, y_i), with x_i denoting the feature vectors and y_i the supposed outcome as teacher value and $i = 0, \ldots, n$. Now we can derive the hyperplane, with w representing its normal vector and b its offset from the origin, by maximizing the margin while fulfilling the following set of inequalities to classify all data points correctly:

$$y_i(\langle x_i, w\rangle + b) - 1 = 0 \tag{3.9}$$

This can be combined in the primal form of the optimization problem using Lagrange multipliers α_i:

$$L_P(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i(y_i(\langle w, x_i\rangle + b) - 1) \tag{3.10}$$

This function has to be minimized with respect to w and b and maximized with respect to all $\alpha_i \geq 0$ to find the desired optimal values w^* and b^*. As the solution has to be at the same time a minimum and a maximum, it is a saddle point and, therefore, for the first derivative it must hold that $L'(w^*, b^*, \alpha) = 0$. Using this, we finally get the dual form of the optimization problem:

$$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j\rangle \tag{3.11}$$

This function now has to be maximized with respect to α and the resulting values can be used to compute the optimal parameters w^* and b^*:

$$w^* = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{3.12}$$

$$b^* = -\frac{1}{2}\langle w^*, x_k + x_l\rangle \tag{3.13}$$

where x_k, x_l are support vectors from one of the classes, i.e. $y_k = 1$, $y_l = -1$ and α_k, $\alpha_l > 0$. The decision function f computing the output for a new example z is then defined as follows:

$$f_{\alpha,b}(z) = \text{sign}(\sum_{i=1}^{n} \alpha_i y_i \langle xi, z\rangle + b) \tag{3.14}$$

The so-called kernel trick can be used on non-linear separable data. It projects the training examples into a higher-dimensional space where the separation can be easier. In this thesis, only linear SVMs will be employed.

k-means++. As unsupervised alternative, a k-means++ clustering (Arthur and Vassilvitskii, 2007) is introduced. As shown in Algorithm 1, the data points are repeatedly assigned to the closest cluster center and the cluster center is updated as the center of mass of the points assigned to it.

The distance measure used to assign points to cluster centers in line 3 plays an especially important part. In most applications, the Euclidean distance can be used, but we showed in Mack et al. (2017b) that for the histograms resulting from grape bunches, the χ^2-distance measure provides better results:

$$d_{\chi^2}(x, y) = \sum_{i}^{m} \frac{(x_i - y_i)^2}{x_i + y_i} \tag{3.15}$$

for $x, y \in \mathbb{R}^m$. The reason for this is that the values in the histograms are normalized, and, thus, tend to be very small, making the Euclidean distance prone to numerical errors.

The initialization of the cluster centers is usually done randomly. Instead, we follow the approach introduced by Arthur and Vassilvitskii (2007): only the first cluster center is chosen at random. Afterwards, all data points are weighted with their smallest distance to one of the already chosen cluster centers. This makes it more likely that the cluster centers are initialized as far spread over the points as possible, reducing the time required for the clustering and at the same time increasing the robustness.

We can assume that the amount of berry surface points in a pointcloud of a grape bunch is larger than the amount of stem surface points. Therefore, we set $k = 2$ and automatically assign the label ”berry” to the larger of the two resulting clusters and ”stem” to the smaller one, respectively.

Algorithm 1: k-means algorithm.

```
Input: $n$ data points, $k \in \mathbb{N}$
Output: $k$ cluster centers

1 Initialize $k$ clusters by using randomly chosen data points as cluster centers;
2 while the assignments of points to clusters changes do
3     Assign each point to the closest cluster center based on their distance;
4     Compute new cluster centers as center of mass of all points assigned to it;
5 end
```

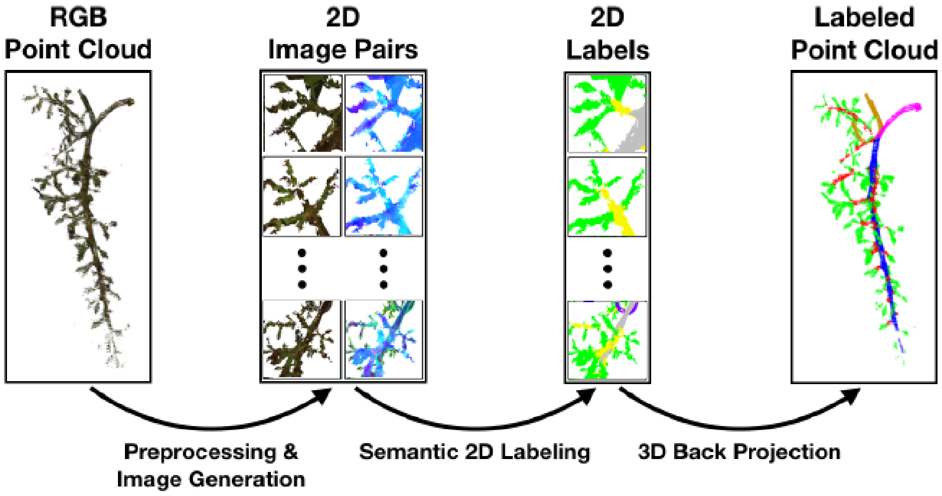


Figure 3.1: *The figure was adjusted from Japes et al. (2018). It shows the overall workflow of the SnapNet used for semantic labeling. First, 2D images are generated from the original pointcloud. Then, the CNN assigns semantic labels to the 2D images. Finally, the labels are transferred to the pointcloud using a 3D back projection. The colors of the 2D labels assigned by the CNN are chosen randomly for the visualization: in this case, yellow refers to the twigs, violet to the hook, grey to the rachis, and green to the subtwigs.*

3.3 Semantic labeling using Convolutional Neural Networks

We rely on a multi-view approach as described by Boulch et al. (2018) consisting of the following steps:

1. 2D images are generated from the pointcloud.
2. The SnapNet semantically labels those 2D images.
3. The labels are transferred to the pointcloud using a 3D back projection.

Figure 3.1 shows this workflow. We describe the steps in more detail in the following.

For the generation of 2D images, a RGB-mesh is derived from the pointcloud based on the greedy surface triangulation algorithm described by Marton et al. (2009), yielding a meaningful representation of the surfaces. To save some depth information even after the reduction to 2D, additional views containing geometric features are created. An encoded version of those views is shown in Figure 3.1 in the second image from the left. While the left views show the RGB colors, the right views represent depth information, namely the normal deviation (encoded on the green channel), local noise (encoded on the red channel), and the distance to the camera (encoded on the blue channel).

The views itself are generated by randomly choosing a point of the scene and creating a line through this point. Three points on the line in fixed distances from the point are used as camera positions, facing always towards the point itself and covering multiple scales. While

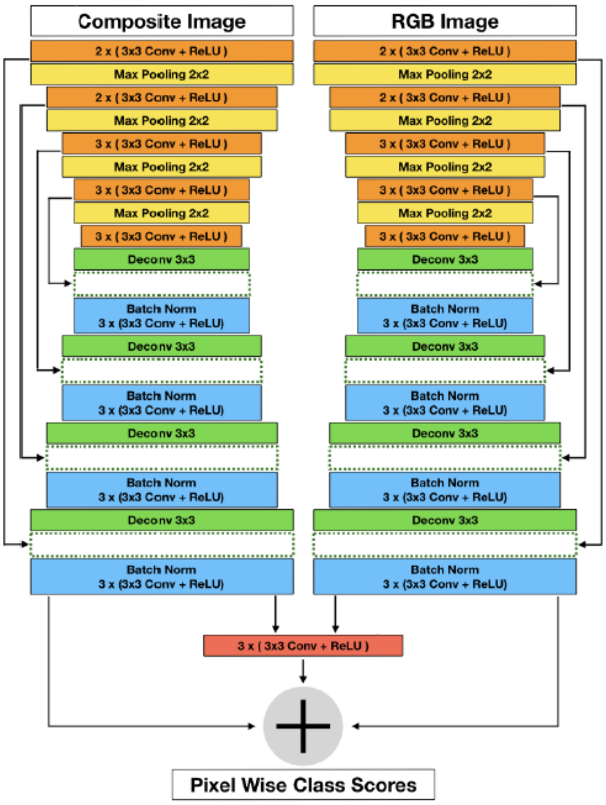


Figure 3.2: *The figure was taken from Japes et al. (2018). It shows the architecture of the U-Net based fusion network with residual correction making up the SnapNet, as introduced by Boulch et al. (2018).*

this is a rather simple strategy, tests performed by Barmscheidt (2018) during his master's thesis showed that distributing the viewpoints uniformly on a sphere around the pointcloud does not significantly improve the results.

The views created from the multiple camera positions are used as input for the CNN. Similar to Boulch et al. (2018) we use an U-Net-based fusion network with residual correction, as shown in Figure 3.2. The complete network consists of two U-Nets (Ronneberger et al., 2015), one of which is trained on the composite images including the depth information, and the other one on the RGB images. The U-Net architecture starts with a contracting part (shown in orange and yellow in Figure 3.2) followed by an expansive part (green and blue in Figure 3.2). The last feature maps of both networks are fed into a residual network (depicted in red in Figure 3.2) that learns a corrective term that can be applied to the predictions. The final result is then a pixel-wise score for each label.

In some more detail, the contracting part consists of multiple convolution layers (orange in

Figure 3.2). In each layer, two 3 × 3 unpadded convolutions are applied to the feature map, followed by a Rectified Linear Unit (ReLU) that serves as activation function for the neuron. Subsequently, a 2 × 2 max pooling operation (yellow layer in Figure 3.2) is performed as downsampling step.

The expansive part starts with an upsampling layer using a 3 × 3 deconvolution (green in Figure 3.2). The arrows in Figure 3.2 mark the copying of feature maps from the contracting part to the expansive part. The information from the upsampled feature map and from the one copied from the contracting part are concatenated and blended together using three 3 × 3 convolution layers (blue in Figure 3.2), each followed by a ReLU.

Finally, the fusion network consists of three 3 × 3 convolutions and a ReLU to learn the corrective term. It outputs class scores for each pixel in the image. To project this information back to 3D data, for each vertex of the mesh computed during image generation the score vectors of the pixels of each surface it is part of are summed up. The vertex is then given the label with highest class score. Based on this, the label for each point of the pointcloud can be derived as the label of the nearest vertex.

Chapter 4

Constraint-based 3D grape bunch reconstruction

Following the semantic labeling introduced in Chapter 3, the grape bunch is reconstructed based on a Reversible-Jump-Markov-Chain-Monte-Carlo (RJMCMC) optimization. We use constraints to steer the process towards an optimum. Section 4.1 introduces constraints in general, while Section 4.2 provides an overview over related work concerning 3D object reconstruction using constraints. Finally, we present our variant of the RJMCMC algorithm in Section 4.3.

4.1 Constraints

In general, constraints can be used to limit values that can be assigned to one or a combination of several variables. In our case, the goal is to avoid the requirement of teacher values by using an optimization procedure that relies on constraints to enforce restrictions on the target object instead.

One advantage of constraints is that they can be described in natural language, making them transparent and, thus, enabling discussions with experts from other application fields.

Constraint types vary highly ranging from simple equality and inequality constraints to complex relations between objects. E.g., the positions of parts of the objects can be restricted to lie in a fixed distance with respect to each other.

One classical example are statistical constraints as used, e.g., by Schöler and Steinhage (2015). They can be easily derived from examples and often provide a good representation of the object. But this approach fails for object classes with high within-class-scatter, like grape bunches. E.g., the lengths of twigs vary greatly, with upper twigs often having more than double the length of lower twigs. Thus, their mean value and standard deviation can not be used to restrict the length of a single twig. We introduce geometrical and topological constraints as an alternative, with geometrical constraints restricting the lengths and angles of parts of the model and topological constraints the whole topology of the grape bunch, e.g., allowing only connections between specific types of parts.

4.2 Related work

Constraints are employed in different contexts when it is necessary to restrict the hypotheses space, but also allow a certain variation. One example is the sampling of different environments, e.g., coffee shops and golf courses (Yeh et al., 2012), or of an animation (Chenney and Forsyth, 2000).

Another application field is the reconstruction of complex objects in 3D data. One class of methods aims at the reconstruction of buildings, focusing on aspects like the reconstruction of the building from airborne LiDAR data (Hu et al., 2018, Wu et al., 2017a), a detailed facade reconstruction (Dehbi et al., 2016) or the derivation of the building model based on pointclouds generated indoors (Ochmann et al., 2016). Dehbi et al. (2016) propose to derive geometrical, topological, and semantic constraints from examples to generate a better reconstruction. Hu et al. (2018) formulate constraints as descriptions of decompositions and topological relations between primitives.

In the field of indoor scene reconstructions scene databases with manually segmented and labeled mesh models are available and can be used to learn contextual relationships as constraints between objects (Chen et al., 2014). Those constraints are then used to deal with occlusion and a low-quality of the RGB-D data. Additionally, repetitive structures can be detected to accelerate the identification of objects in low-quality data (Kim et al., 2012).

The reconstruction of plants in 3D data is often done using parametric L-Systems (Huang et al., 2013, Xin et al., 2014). L-Systems are able to realistically simulate the growing process of a plant and can be set to develop in such a way that the final shape of the simulated plant comes close to the shape of a given real plant set by a user. Still, these methods do not provide a one-to-one reconstruction of a plant.

A prominent field in 3D plant reconstruction is the reconstruction of trees. Aiteanu and Klein (2014) introduce a hybrid method for sparse and dense pointclouds, fitting ellipses into cross sections and deriving a spanning tree for each group of branches. Afterwards, the subtrees are joined using a nearest neighbor approach. Similarly, Livny et al. (2010) and Preuksakarn et al. (2010) use a global optimization fitting a skeleton into a sparse, incomplete and noisy pointcloud. While those methods are suitable for unfoliated trees, they rely on the visibility of the largest part of the skeleton in the data.

The 3D reconstruction of grape bunches proves to be a rather challenging problem: other than in the other application fields, the visibility of the inner stem skeleton cannot be provided, as in the development stage BBCH 89 with berries ripe for harvest it is in most cases completely occluded by the outer berries. As already mentioned in Section 4.1, Schöler and Steinhage (2015) use statistical constraints to validate samples of a grape bunch architecture drawn by an RJMCMC algorithm. Compared to this, we aim to include more general constraints similar to those derived by Dehbi et al. (2016) that allow us to work on a larger variety of objects.

Several approaches exist to derive constraints automatically. Constraint programming has been applied to different pattern mining and rule learning problems to lead to more robust and flexible methods (De Raedt et al., 2010). Temple et al. (2016) infer constraints for product lines from randomly created examples that are classified according to an oracle. Another method using adversarial constraint learning is introduced by Ren et al. (2018).

Most current work focuses on the reconstruction of objects that are either mostly visible, or can be reconstructed using a data base with training examples. We assume that our data shows only the outside of the grape bunch, while the inner stem skeleton is completely occluded. Training data cannot be easily derived, as the removal of berries changes the topology of the stem skeleton, e.g., the angles of twigs can get distorted, as the weight of the berries is missing, or pedicles can be accidentally removed with the berries. Even so, the manual fitting of cylinders into the data would only be feasible for a low number of grape bunches.

4.3 Reversible-Jump-Markov-Chain-Monte-Carlo optimization

As the inner stem skeleton is not visible in the data, an optimization process has to be used that is able to do both, model selection and parameter estimation. The Reversible-Jump-Markov-Chain-Monte-Carlo (RJMCMC) was first introduced by Green (2003). It extends the Markov-Chain-Monte-Carlo algorithm to deal with hypotheses spaces for models that require changing the model's dimension and the model's parameters as well. The method uses jumps to switch between model states, converging to an optimum. Simulated Annealing is used to support the convergence.

The approach is presented in Algorithm 2.

The reconstruction problem is translated to a Bayesian decision problem with

- $\mathcal{D}$ representing the sensor data, e.g. a 3D pointcloud,
- S as the hypotheses space and
- $x \in S$ a hypothesis.

Each hypothesis $x = (k, \theta_k)$ consists of a model $M_k \in \{M_1, ..., M_n)$ from an enumerable set of models, defined using its index k, with dimension d_{m_k} and a set of parameters with dimension d_{n_k} relating to the respective model. The algorithm then in each step proposes a

Algorithm 2: Reversible-Jump-Markov-Chain-Monte-Carlo.

Input: initial hypothesis x_0, constant annealing factor c, end temperature t_f
Output: optimized hypothesis x^*

```
1  $t_0 = 1.0$;
2  while $t_i \leq t_f$ do
3      $m \sim j(xx')$;
4      $u \sim q_m(u|x, k')$;
5      $(\theta'_{k'}, u') = g_{kk'}((\theta_k, u))$;
6      $\mathcal{A}(x'|x) = \min\left(1, \frac{\pi(x'|\mathcal{D})^{\frac{1}{t_i}}}{\pi(x|\mathcal{D})^{\frac{1}{t_i}}} \cdot \frac{j'(xx')}{j(x'x)}\right)$;
7      if $a \sim U(0, 1) \leq \mathcal{A}(x'|x)$ then
8          $x_{i+1} = x'$;
9      else
10         $x_{i+1} = x_i$;
11     $t_{i+1} = c \cdot t_i$;
12 end
13 return $x_{n-1}$.
```

switch from state x to a new state x' with probability $j(xx')$. A random vector u is sampled from a distribution $q_m(u|x, k')$. To derive the new state x', a jump $g_{kk'}(\theta_k, u)$ is defined as a bijection between (θ_k, u) and $(\theta_{k'}, u')$. The vectors u and u' are used to match the dimension between the vectors, such that $d_{n_k} + d_{kk'} = d_{n_{k'}} + d_{k'k}$. This is called the *dimension matching condition*.

Assuming a current state x and a new state x', we formulate the probability of accepting the new state $\mathcal{A}(x'|x)$ based on the posterior distribution π and the proposal probability j:

$$\mathcal{A}(x'|x) = \min\left(1, \frac{\pi(x'|\mathcal{D})}{\pi(x|\mathcal{D})} \cdot \frac{j'(xx')}{j(x'x)}\right) \tag{4.1}$$

For the posterior distribution $\pi(x|\mathcal{D})$ it holds that

$$\pi(x|\mathcal{D}) \propto \pi(x) \cdot \mathcal{L}(\mathcal{D}|x) \tag{4.2}$$

with $\mathcal{L}(\mathcal{D}|x)$ being the probability of observing sensor data $\mathcal{D}$ given the hypothesis x.

Simulated Annealing is used to control the convergence, decreasing the acceptance probability over time. With the temperature t_i the acceptance probability for jump proposals decreases, depicted by the exponent $1/t_i$ added to the acceptance probability in line 6 of Algorithm 2. Line 11 shows the calculation of the new temperature following a geometric annealing.

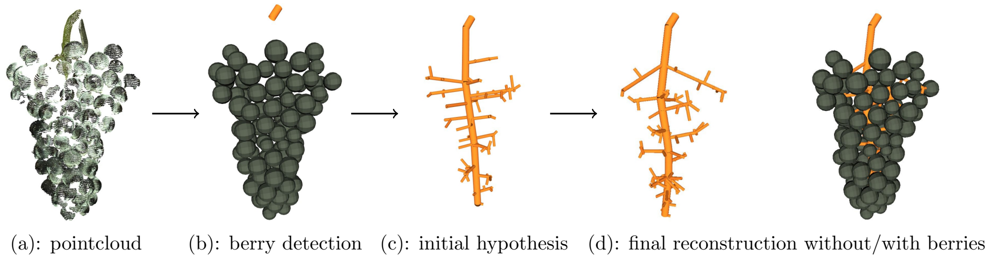


Figure 4.1: *The step-by-step workflow of the RJMCMC optimization shown on a Pinot Noir grape bunch. First, the visible grape bunch organs are derived from the pointcloud (a), namely berries and peduncle (b). Based on them an initial hypothesis is created (c) and finally optimized with the RJMCMC algorithm described in the following. The resulting grape bunch reconstruction is shown in (d). The right image shows the full reconstruction and the left only the stem skeleton, to provide a better overview.*

In the following, we give more details about the steps of the RJMCMC algorithm and how it is adjusted for the application to grape bunches. First, we show the set of jumps that is used to transform the current hypothesis to a new one in line 5 of Algorithm 2. Then, we explain the chosen constraints and how they can be used to compute the prior (line 6 of Algorithm 2). We explain our variant of an RJMCMC algorithm with locally optimized jumps in Section 4.3.3. Finally, section 4.3.4 presents the initial hypothesis and how it can be derived from the data.
Figure 4.1 demonstrates the reconstruction process on an example.

4.3.1 Jumps

Following the notation introduced in Algorithm 2, jumps are bijective functions $g_{kk'}$ that are used to derive a new state proposal x' from the former state x. The jumps presented here were derived in collaboration with the experts from the Institute for Grapevine Breeding Geilweilerhof in Siebeldingen. We divide the jumps into two classes: topological jumps change the dimension of the model by removing or adding parts. Geometrical jumps are applied to optimize the parameters, like lengths or angles of the model parts, but the model dimension itself remains fixed. Based on those categories, we define the following jumps:

Topological jumps:

1. Combine twigs: Combine two twigs with small distance between their startpoints on the rachis (less than 5 mm) into one.
2. Split twig: Split one twig into two new twigs. The endpoints of the twigs are chosen at random on a circle with radius between two parameters d_{min} and d_{max} around the old endpoint. We set those values to d_{min} = 1 mm and d_{max} = 20 mm.

3. Reassign berry: Assign a berry to another twig.
4. Reassign subtwig: Assign a subtwig to another twig.
5. Attach twig as subtwig: Replace a twig with a subtwig starting from another twig in the upper half of the grape bunch.
6. Detach subtwig as twig: Replace a subtwig with a twig.
7. Add subtwig: At a position with several pedicles with startpoint close to each other on the corresponding twig, add a subtwig and reattach the closest pedicles to it.
8. Remove subtwig: Remove a subtwig and attach the pedicles to the twig the subtwig was originally growing out of.

Geometrical jumps:

9. Move twig start/endpoint: Move the startpoint or, respectively, the endpoint of a twig.
10. Move subtwig start/endpoint: Move the startpoint or, respectively, the endpoint of a subtwig.

Each jump needs to have a reverse jump for the corresponding function to be bijective. For jumps 1 and 2 it holds that they are reverse jumps of each other, as are jumps 5 and 6, and 7 and 8. Jumps 3, 4, 9, and 10 are their own reverse jump.

4.3.2 Constraint processing

In Algorithm 2, an acceptance probability is computed in line 6, requiring the computation of the posterior distribution. Following Equation (4.2) in Section 4.3, we have to define the Likelihood of the data $\mathcal{L}(\mathcal{D}|x)$ and the prior $\pi(x)$.

The Likelihood $\mathcal{L}$ can be calculated as the distance between the surfaces of the cylinders making up the bunch reconstruction hypothesis and the non-berry points in the pointcloud. Berry points are ignored, as there is no jump to change the berries, thus there can be no change in the Likelihood for those parts. For points that cannot be assigned to the surface of a part of the hypothesis a penalty term is added.

The prior distribution $\pi(x)$ evaluates the quality of the current hypothesis. For this, we define constraints that have to be fulfilled.

For each instance of a constraint a probability C_i is computed. The prior π is then proportional to the product of these probabilities:

$$\pi \propto \prod_{i=1}^{n} C_i \tag{4.3}$$

Each probability C_i is expressed as a relation defined on a value π_i computed on the hypothesis.

$$C_i = \text{relation}(\pi_i) \tag{4.4}$$

This is either a unary or a binary relation and expresses in what way the constraint restricts the involved variables. The following relations are used in this thesis:

$$\text{similar}_\sigma(x, \mu) = \mathcal{N}(x, \mu, \sigma) \tag{4.5}$$

$$\text{smaller}_\sigma(x, y) = \mathcal{N}(x, \min(x, y), \sigma) \tag{4.6}$$

Equation (4.5) is used in constraints that restrict a value to be close to another given value μ. The strictness of the constraint can be adjusted using the standard deviation σ.
Equation (4.6) is defined on two different hypotheses that are evaluated relative to each other, i.e. x is the input computed on one of the hypotheses and y the input computed on the other hypothesis. Equation (4.6) assigns a higher probability to the hypothesis with smaller value. The definition of a relation that prefers larger values can be done analogously, but is not required here.

Constraints derived for grape bunches

Similar to the jumps, the constraints used in this thesis were derived in collaboration with the experts at the Institute for Grapevine Breeding Geilweilerhof in Siebeldingen. We categorize them into geometrical constraints, referring to lengths or relations between grape bunch organs, and topological constraints, concerning the topology of the whole grape bunch. In the following, we present the list of constraints chosen for our application.

Geometrical constraints:

Angle constraint:

Angles have to be about 90°.

$$G_1^{(k)} = \text{similar}\left(\sum_{(i,j)} \text{angle}(x_i, x_j), 90\right) \tag{4.7}$$

This constraint is applied between the rachis and twigs, and, additionally, to the twigs/subtwigs and pedicles.

Similarity constraint:

There should be no elements that are very similar to each other.

$$G_2^{(k)} = \text{similar}\left(\sum_{(i,j)} \text{count\_similar\_elements}(x_i, x_j), 0\right) \tag{4.8}$$

This constraint is applied to twigs, as very similar twigs can be combined. Here, we define ”similar” to be a short distance on the rachis of less than $d_{sim} = 10\,\text{mm}$ and a difference in orientation of less than $o_{sim} = 30°$.

Overlap constraint:

There can be no overlapping parts (with minimal distance between the parts smaller than the radius of the primitive).

$$G_3^{(k)} = \text{similar}\left(\sum_{(i,j)} \text{count\_overlapping\_parts}(x_i, x_j), 0\right) \tag{4.9}$$

We apply this constraint to all parts of the grape bunch.

Length constraint:

The lengths have to be as small as possible.

$$G_4^{(k)} = \text{smaller}\left(\sum_{i=1}^{n} \|x_i\|\right) \tag{4.10}$$

This constraint is applied to twigs, subtwigs and pedicles separately.

Deviation constraint:

The standard deviation has to be as small as possible.

$$G_5^{(k)} = \text{smaller}\left(\frac{1}{n}\sum_{i=1}^{n} \|x_i\| - \bar{x}\right) \text{ with } \bar{x} = \left(\frac{1}{n}\sum_{i=1}^{n} \|x_i\|\right) \tag{4.11}$$

This constraint is applied to pedicles.

Topological constraints:

Connection constraint:

Depending on the type of parts of the model, they have to be connected to each other. The relation type_of(x) returns the type of the grape organ x, e.g., rachis or pedicle.

$$T_1^{(k)} = \text{connection\_between}(\text{type\_of}(x_i), \text{type\_of}(x_j)) \tag{4.12}$$

This holds for the peduncle and a part of the rachis, parts of the rachis and twigs, twigs and subtwigs, and pedicles and berries. This constraint does not derive a cost, but it is checked whether or not it is fulfilled. If not, the hypothesis is invalid and has to be adjusted.

Potential connection constraint:

The parts of the skeleton involved in this constraint can be connected to each other. Again, the relation type_of(x) returns the type of the grape organ x.

$$T_2^{(k)} = \text{potential\_connection\_between}(\text{type\_of}(x_i), ..., \text{type\_of}(x_n)) \tag{4.13}$$

This constraint is applied to twigs and subtwigs (a twig can be connected to a subtwig, but there are twigs without subtwigs as well), and twigs/subtwigs and pedicles (a pedicle can be either connected to a subtwig or to a twig). All connections not covered by this constraint or in $T_1^{(k)}$ are not allowed, e.g., connecting a pedicle to the rachis. Similar to the connection constraint, this constraint does not derive a cost, but it is checked whether or not it is fulfilled. If not, the hypothesis is invalid and has to be adjusted.

Decreasing length constraint:

Lengths of elements should decrease from the top of the stem skeleton to the bottom.

$$T_3^{(k)} = \text{smaller}\left(\sum_{(i,j)} \text{length\_shorter\_elements}(x_i, x_j)\right) \tag{4.14}$$

This holds for the twigs, their length generally decreases.

Twig endpoint constraint:

There should be no twig endpoints outside the convex hull spanned by the berries.

$$T_4^{(k)} = \text{similar}\left(\sum_{i=1}^{n} \text{count\_if\_outside\_convex\_hull}(x_i), 0\right) \tag{4.15}$$

The exact definitions of the relations used in the similarity constraint, the overlap constraint, the decreasing length constraint and the twig endpoint constraint are given in Appendix A.

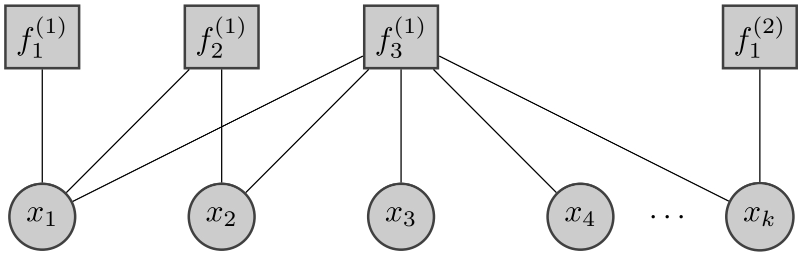


Figure 4.2: *Example for a factor graph with random variables* $x_1, ..., x_k$ *(spheres) and factors* $f_1^{(i)}, f_2^{(i)}, f_3^{(i)}$ *(boxes). Each factor encodes a constraint that is defined on one to* k *random variables, e.g.* $f_1^{(1)}$ *is defined on* x_1*, and a second instance of the same constraint* $f_1^{(2)}$ *on* x_k*.*

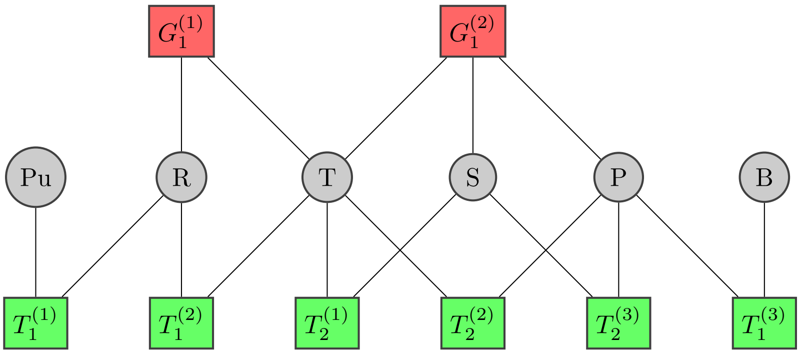


Figure 4.3: *Factor graph of the initial hypothesis including random variables (grey spheres), constraints (boxes, red geometrical and green topological), and the relations between them. The random variables represent the grape bunch organs (peduncle (Pu), rachis (R), twigs (T), subtwigs (S), pedicles (P), berries (B)). The constraints show connection constraints* $T_1^{(k)}$*, the potential connection constraints* $T_2^{(k)}$*, and angle constraints* $G_1^{(k)}$*.*

Factor graphs

We use a factor graph representation where the factors encode constraint instances and the random variables parts of the hypothesis that make up the scope of the constraints. Figure 4.2 shows an example for such a factor graph. This representation provides an efficient way to compute the prior from the joint probability of the factor graph due to offering a structured overview over the dependencies between constraints and random variables.

The factor graphs used in the creation of the initial hypothesis and afterwards for the optimization are shown in Figures 4.3 and 4.4. The types of random variables are the grape bunch organs (peduncle (Pu), rachis (R), twigs (T), subtwigs (S), pedicles (P), berries (B)). As mentioned before, we divide the constraints into two categories: geometrical and topological. Geometrical constraints are depicted with red boxes in the figures and concern lengths or relations of parts of the stem skeleton with each other and topological constraints are depicted with green boxes and restrict the topology of the whole stem skeleton.

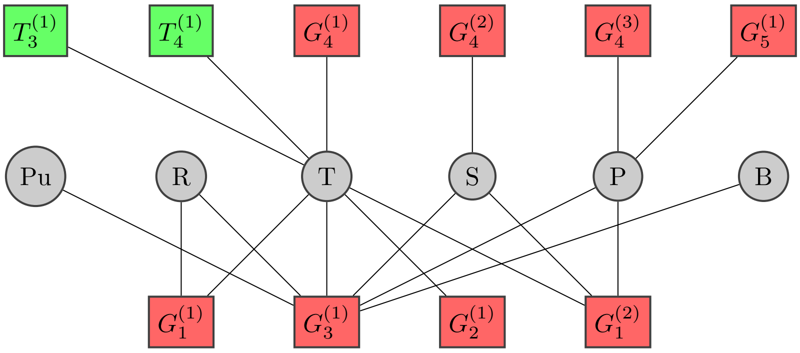


Figure 4.4: *Factor graph of the optimization including random variables (grey spheres), constraints (boxes, red geometrical and green topological), and the relations between them. The random variables represent the grape bunch organs (peduncle (Pu), rachis (R), twigs (T), subtwigs (S), pedicles (P), berries (B)). The constraints show angle constraints* $G_1^{(k)}$, *similarity constraints* $G_2^{(k)}$, *overlap constraints* $G_3^{(k)}$, *length constraints* $G_4^{(k)}$, *deviation constraints* $G_5^{(k)}$, *decreasing length constraints* $T_3^{(k)}$, *and twig endpoint constraints* $T_4^{(k)}$.

The set of constraints used during the generation of the initial hypothesis (Figure 4.3) and the optimization (Figure 4.4) show several differences:

- Connection and potential connection constraints are only used during the creation of the initial hypothesis. The reason for this is that we define the jumps in such a way that the connection constraints are still fulfilled by their result. E.g., for the jump ”Remove subtwig” a subtwig is removed from the hypothesis. The connection between the remaining parts of the hypothesis is kept by attaching the involved pedicles to the parent twig of the subtwig. In this way, the new hypothesis is kept consistent with the connection constraints, thus, as long as the initial hypothesis fulfils them, they do not have to be evaluated again.
- Length, deviation, and decreasing length constraints are only used during the optimization. The reason for this is that they are defined relative between two hypotheses. This makes it more suitable to apply them during an optimization procedure.
- Only connection, potential connection, and angle constraints are used during the creation of the initial hypothesis. While it would be possible to take into account all constraints, this would greatly increase the complexity of the generation of the initial hypothesis. We choose a middle path, where the generation is still rather simple, but only some of the constraints have to be initially fulfilled.

4.3.3 Partitioning into phases

The topological jumps that are used to select the correct model change a number of parameters that have to fit relatively well to yield a high acceptance probability. E.g., combining two twigs leads to the pedicles of both twigs being attached to the new twig, thus, potentially increasing their lengths. If the new twig endpoint in the mid point between the old twigs is not in a good position, this can lead to a rejection of the jump, even if moving it can potentially reduce the pedicle lengths again.

To solve this problem, we suggest to divide the RJMCMC algorithm into two phases. The first phase uses all jumps, but every jump that changes the dimension of the model, i.e., every topological jump, is followed by n geometrical jumps that locally optimize the involved parts of the model. A local optimization allows a better fitting of the involved object parts, increasing the acceptance rate for those jumps. The n geometrical jumps are performed taking the topological jump as basis. For the one topological and n geometrical jumps, an acceptance probability is computed and only the jump with highest acceptance probability is considered for acceptance.

In the second phase, only geometrical jumps are used. We assume that the model has been optimized during the first phase that concentrates on the topological jumps. Now, all that remains is a parameter optimization in the form of a fine-tuning of the positions of the parts.

4.3.4 Derivation of initial hypothesis

To start the optimization with a hypothesis that fits the data, we reconstruct the visible parts of the grape bunch with geometric primitives. The workflow is depicted in Figure 4.5. The semantic labeling of the pointcloud with the SnapNet explained in Section 3.3 yields labels classifying the points into the berries, peduncle, rachis, twigs, and the hook. The hook points are directly removed, as they are not part of the grape bunch itself. In most grape bunch scans, the only other visible parts are berries and the peduncle. The rachis and the twigs are

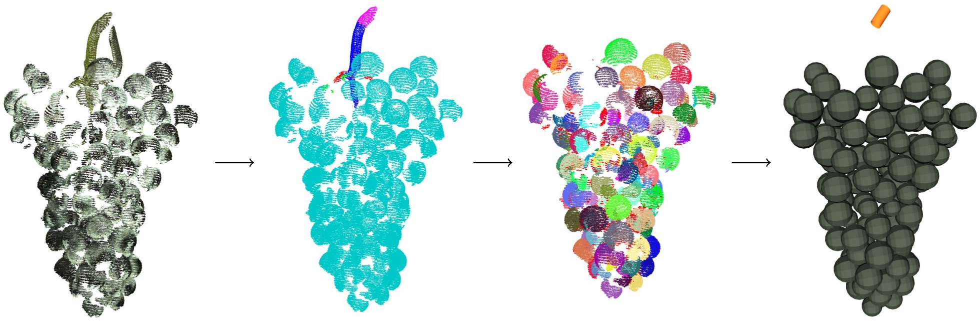

(a): pointcloud (b): semantic labeling (c): region growing (d): berry hypotheses

Figure 4.5: *The step-by-step workflow of the berry detection on a Pinot Noir grape bunch. First, the pointcloud (a) is semantically labeled with the SnapNet as described in Chapter 3, resulting in a labeled pointcloud (b). Region growing is applied to divide the pointcloud into regions, optimally yielding one region per berry (c). Finally, berry hypotheses are derived from the regions using a RANSAC-based approach (d).*

completely occluded by berries.

A well-known and robust method for the fitting of geometric primitives into pointclouds is the Random Sample Consensus (RANSAC) algorithm that is introduced in the following.

RANSAC

This algorithm was introduced by Fischler and Bolles (1981) and adjusted for the usage on pointclouds by Schnabel et al. (2007). The algorithm aims at fitting a geometric primitive (in our application usually spheres or cylinders) into a pointcloud. This method is described in Algorithm 3 and discussed in more detail in the following.

The algorithm requires as input the 3D pointcloud and, additionally, the primitive model that has to be fitted in the form of the required parameters. The number of iterations n_r the algorithm is allowed to try has to be restricted to terminate the method in case of data that do not fit to the model.

In a first step, k_r points are chosen randomly from the pointcloud. On the basis of these points, a model is calculated that represents a primitive that fits to this subset, where fitting is defined as lying in a fixed distance ϵ to the surface of the primitive. Afterwards, for all points it is tested whether they are part of the consensus set, again consisting of all points lying in a distance ϵ to the surface of the model. In the end, the model with largest consensus set is returned.

This method has been shown to run stable even on noisy data sets (Schnabel et al., 2007).

Algorithm 3: Random Sample Consensus.

```
Input: 3D pointcloud, model with $k_r$ parameters, number of iterations $n_r$
Output: model hypothesis $h$, consensus set $C$

for $i = 0; i < n_r; i++$ do
    Randomly select $k_r$ points of the pointcloud and store them in a set $K$;
    hypothesis $h_i$ = model with $d(p, \text{surface}(h_i)) < \epsilon$ for every $p \in K$;
    Create empty consensus set $C_i$;
    for each point p of the pointcloud do
        if $d(p, surface(h_i)) < \epsilon$ then
            Add $p$ to consensus set $C_i$;
    end
end
return Model hypothesis $h_i$ with largest consensus set $C_i$.
```

Berry detection

As the pointcloud with berry surface points includes a large number of berries, we try to make the detection more stable by dividing it into regions, aiming at providing one region for each berry. This is a challenging task, as the berries often touch each other, thus, there is a relatively smooth transition between them. As described in Mack et al. (2017a) and Rist et al. (2018), we use a Region Growing approach introduced by Rabbani et al. (2006) to separate the berries based on the smoothness of the curvature of the points. This curvature can be computed from the eigenvalues of the covariance matrix. Each region r is constructed starting from a seed point P_s that is chosen as the point with minimal curvature value that is not part of any region. For each point $p \in r$, every point $p_n \in N_k(p)$ is added to the region if the difference between its normal and the normal of p does not exceed a smoothness threshold t_s, with $N_k(p)$ being the k-neighborhood of p. Also, it is checked if the curvature value of such a neighborhood value is smaller than a curvature threshold t_c. If this is the case, it is added to the set of seed points P_s. After all seed points have been processed, the region r is closed and a new region r' started, again beginning with the point in P_s with minimal curvature value.

The smoothness threshold t_s has to be chosen such that it separates berries touching each other, but on the other hand does not split the same berry into several regions due to deformations or holes in the data. As these two requirements contradict each other, it is usually not possible to find a perfect value for this parameter. Thus, we have to deal with both oversegmentation (one berry divided into several regions) and undersegmentation (one region including several berries).

The *cultivar* and the *variety set* both contain exclusively grape bunches with spherical berries. Thus, their shape can be approximated by spheres. The *table grape set* includes grape bunches with elliptical berries that require an extension of the method that is discussed in the next section.

As explained in the previous section, RANSAC fits a parametric model into the data. In the case of spheres this parametric model consists of a center point and a radius. The hypothesis fitting the data best is derived as described in Algorithm 3. We use each region resulting from the Region Growing approach as input for RANSAC. To cope with undersegmentation, i.e., several berries in the same region, we use the following iterative strategy, yielding a set of berry hypotheses H:

1. While the number of points in the current region R is still larger than a threshold $t_i \in \mathbb{N}$ and RANSAC is able to find a hypothesis in the data, perform the following steps, otherwise, discard the region:
2. Apply RANSAC as described in Algorithm 3 to the region, yielding a berry hypothesis $h = (c, r)$ and the corresponding consensus set C.
3. Remove all points included in the consensus set from the region, i.e. $R' = R \setminus C$.
4. We apply constraints to the size of the consensus set (it has to be larger than a threshold

$t_i \in \mathbb{R}$) and the radius r (it has to lie in a range $r_{min} \leq r \leq r_{max}$). If both constraints are fulfilled, add the hypothesis to H, otherwise, discard it.

The constraint using the threshold t_i is used to make sure that the region still contains a sufficient number of points to yield a valid hypothesis in step 4. Additionally, the constraint on the radius of the berry hypothesis ensures that it is in a range natural for grape bunches and RANSAC does not fit a very large sphere into the data that covers many points, but does not provide valid dimensions.

The points from the consensus set are removed from the region in step 3. This makes sure that the next run of RANSAC uses points that were not covered by a berry hypothesis before. Still, deformations and oversegmentation can yield multiple hypotheses for the same berry. This is handled with a post-processing step. All hypotheses that overlap for more than $t_o = 25\,\%$ are compared with respect to the size of their consensus set. Only the hypothesis with largest consensus set is kept.

Elliptical berries

While RANSAC works for all primitive shapes, and thus, for ellipses representing elliptical berries as well, we pointed out in Mack et al. (2018) that the larger number of parameters and the non-perfect shapes of berries lead to a very long running time. Therefore, we present an alternative approach that uses least squares fitting (Li and Griffiths, 2004), as depicted in Figure 4.6. First, the approach described in the previous section is used to obtain regions, each containing exactly one berry. Then, an ellipsoid is fitted into each region following the procedure described by Li and Griffiths (2004), where an ellipsoid is defined as the solution

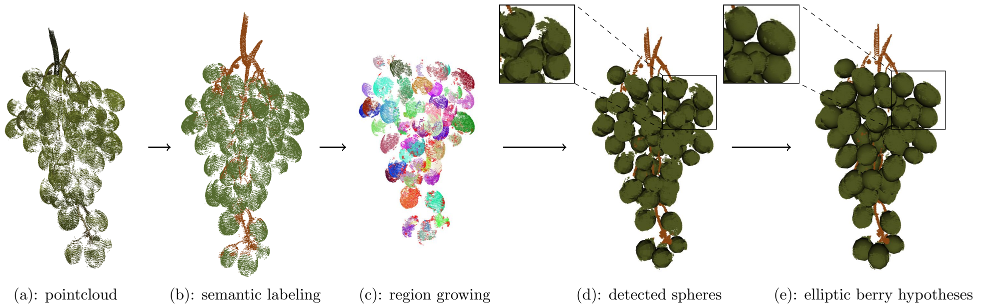


Figure 4.6: *The step-by-step workflow of the elliptical berry detection on a table grape. First, the points of the pointcloud (a) are labeled as berry or stem surface points as described in Chapter 3 (b). Region growing is applied to divide the pointcloud into regions, optimally yielding one region per berry (c). Berry hypotheses are derived from the regions using a RANSAC-based approach (d). On the regions covered by those berries a least squares fitting is applied to fit ellipses into the data (e).*

of:

$$(\boldsymbol{x}-\boldsymbol{m})^T A(\boldsymbol{x}-\boldsymbol{m}) = 1 \text{ with } A = \begin{pmatrix} a & h & g \\ h & b & f \\ g & f & c \end{pmatrix}, \tag{4.16}$$

with $\boldsymbol{m}$ being the center point of the ellipsoid and $\boldsymbol{x} = (x, y, z)$. A is positive definite. Solving the equation leads to

$$F(\boldsymbol{x}) = ax^2 + by^2 + cz^2 + 2fyz + 2gxz + 2hxy + 2px + 2qy + 2rz + d = 0. \tag{4.17}$$

The following definitions are required:

$$I = a + b + c \tag{4.18}$$

$$J = ab + bc + ac - f^2 - g^2 - h^2 \tag{4.19}$$

To yield parameters describing an ellipsoid, it must hold that $kJ > I^2$ with $k = 4$. With this, it is possible to estimate half-axis vectors and half-axis lengths of the ellipsoid that covers the largest number of points.

Peduncle detection

While the berries provide the endpoints of the pedicles, the peduncle represents the starting point of the stem skeleton as uppermost part of the main stem. It can usually be detected robustly by fitting a cylinder into the peduncle points with RANSAC. Again, constraints are used to limit the range of the cylinder radius.
In some cases, the peduncle is not visible in the data. Then, we place a cylinder in the center of mass of the upper layer of berries to simulate a peduncle in the most likely place.

Initial stem skeleton

Based on the detected berries and the peduncle, we create an initial stem skeleton to connect all parts with each other. For this the following steps are performed:

- Sidebunch detection: a RANSAC-based cylinder detection is performed on the point-cloud consisting of berry center points. To make up for the shape of sidebunches being only approximately cylindrical, we set the distance ϵ for the allowed distance of a point to the cylinder surface to a relatively large 10 mm and consider a cylinder a valid candidate for a sidebunch if the consensus set includes more than $t_{sb} = 20$ berries. We remove sidebunch candidates if their endpoints are closer than their combined radii to another sidebunch or the lowest point of the main bunch, as due to the volume of the berries the endpoints of bunches are always at a certain distance to each other. The startpoint of the sidebunch is set heuristically to a point directly below the peduncle and its endpoint to the center of mass of the berries in a radius of $r_{sb} = 15\,\text{mm}$ around the cylinder endpoint. If this point is inside a berry, it is moved upwards towards the startpoint.

- Twig endpoint detection: k-means is applied to the berry centers to cluster them with respect to their distance to each other. The retrieved cluster centers are used as candidates for twig endpoints. To avoid twig endpoints inside of berries, we compute a set of points inside the grape bunch, but with a certain distance to each berry. The points closest to the twig endpoint candidates are finally chosen as twig endpoints.
- Rachis insertion: the endpoint of the peduncle is used as startpoint of the rachis. The lowest twig endpoint detected in the step before defines the endpoint of the rachis. In between start- and endpoint, the rachis is estimated at one third and two thirds of the grape bunch length as the center point of the berries that lie inside a horizontal layer at the given height.
- Twig insertion: a twig is inserted at each twig endpoint, leading straight to the point at the same height on the rachis. If the distance between the twig endpoint and the closest point on a side bunch would be shorter, the twig is inserted there as a subtwig, instead.
- Pedicle insertion: a pedicle is inserted between each berry and the point on a twig or subtwig with smallest distance to it. While the pedicle is attached to the center point of the berry for the sake of simplicity, the radius of the berry is subtracted from its length to provide correct lengths.

Chapter 5

Experimental results

In this chapter, we present the results achieved applying our method to grape bunches. First, we give details about the implementation and explain the methods we used for the evaluation. Then, we show the results of the different approaches we used for semantic labeling: our HFPFH descriptor either with SVM or k-means, and deep learning. The results were partly published in Mack et al. (2018) and Japes et al. (2018). Afterwards, we present the results of the grape bunch reconstruction, divided into the initial hypothesis, the final hypothesis, the compactness, and the modification of the RJMCMC algorithm by partitioning it into phases. These results were partly published in Rist et al. (2018) and Mack et al. (2019). Finally, we discuss the results.

5.1 Implementation and evaluation methods

All methods were implemented in C++, using the Pointcloud Library (Rusu and Cousins, 2011) for Region Growing, Fast-Point-Feature Histograms, RANSAC, and for visualization purposes.

As presented in Table 2.1 in Section 2.4, labeling gold standards are available for 19 grape bunch scans (4 Calardis Blanc, 5 Dornfelder, 5 Pinot Noir, and 5 Riesling) and 14 stem skeleton scans (4 Calardis Blanc, 5 Dornfelder, and 5 Pinot Noir) from the *cultivar set* for the classes berries, peduncle, rachis, twigs, subtwigs, and hook. Additionally, the *table grape set* provides labeling gold standards for all 11 grape bunch scans, but here we only differentiate between berry and stem surface points.

As different stem types, like twigs or the rachis, are hard to distinguish based on their shape and color alone, we evaluate the HFPFH descriptor only for the two classes ”berry surface points” and ”stem surface points”. The stem skeleton scans show only different kinds of stems, thus, they are not used in the training of the HFPFH descriptor.

As deep learning approaches are deemed more powerful than traditional descriptors, we apply the SnapNet to all scans with labeling gold standard from the *cultivar set* using the full set of classes.

We evaluate all steps of the grape bunch reconstruction, i.e., berry detection, creation of initial hypotheses, optimization, and dividing the algorithm into phases, on all 52 grape bunch scans from the *cultivar set*.

The *table grape set* shows grape bunches with elliptical berries; therefore, we evaluate the presented method for the reconstruction of elliptical berries on this data set in comparison to the statistical reference data. For the stems, there is no reference data available.

The berries from the *variety set* are spherical and of similar sizes as those in the *cultivar set*, thus, we assume that they are detected with similar quality. The 10 grape bunches in this set are chosen such that the five compactness classes are covered by two grape bunches each. The compactness is mostly defined by the grape bunch architecture, thus, we take the stem reference data of this data set into account in the evaluation of the grape bunch reconstruction method.

The RJMCMC algorithm chooses the jumps randomly, hence, we present the average results from 10 runs for each grape bunch. Our variant of the RJMCMC algorithm using local optimization for topological jumps is evaluated by comparing the acceptance rates for topological jumps to the base version without local optimization. In the last step, we show a new method for the derivation of the compactness and its robustness on the *cultivar* and *variety set*.

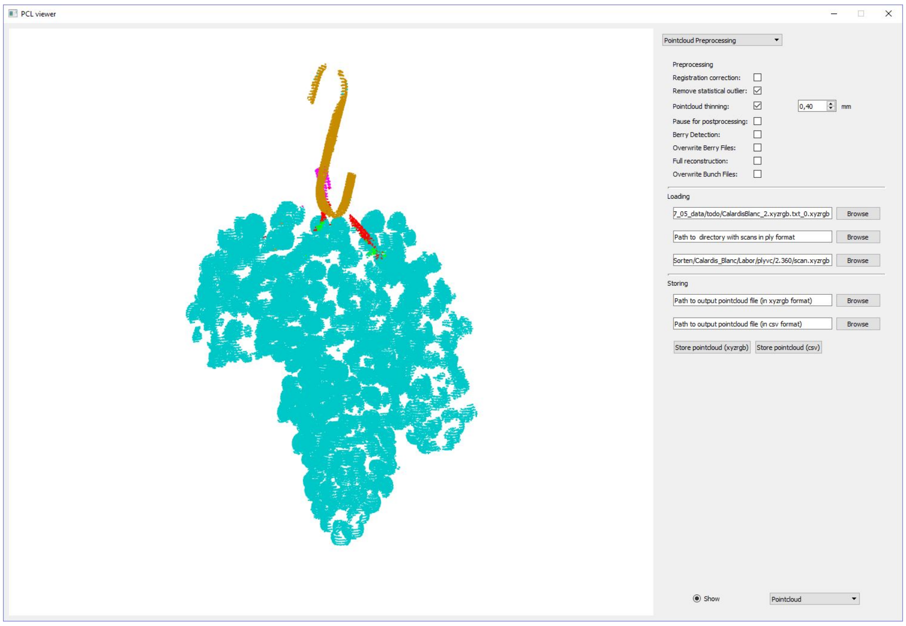
5.2 Graphical User Interface

Figure 5.1 shows the build-up of the graphical user interface. On the left hand side, a viewport provides the possibility to inspect the current status of the grape bunch reconstruction, i.e., the pointcloud, the labeled pointcloud, the berries, the peduncle, or the complete stem skeleton. In this figure, a pointcloud resulting from semantic labeling with the SnapNet is shown.

The right hand side gives the user access to parameters and the possibility to load pointclouds. Instead of loading just single pointclouds, the application of the pipeline to all grape bunches in a chosen directory is possible, which is important for the efficient phenotyping at the Institute for Grapevine Breeding Geilweilerhof in Siebeldingen.

A checkbox in the lower right corner gives the user the possibility to remove parts of the reconstructed grape bunch from view or add them again. This function is necessary to look at, e.g., the grape bunch stem skeleton without having the berries blocking the view. Figure 5.2 shows how different grape bunch organs can be removed from view.

The shown view depicts the panel for pointcloud preprocessing on the right side. In the upper right corner, other panels can be chosen for the detection of berries, the detection of



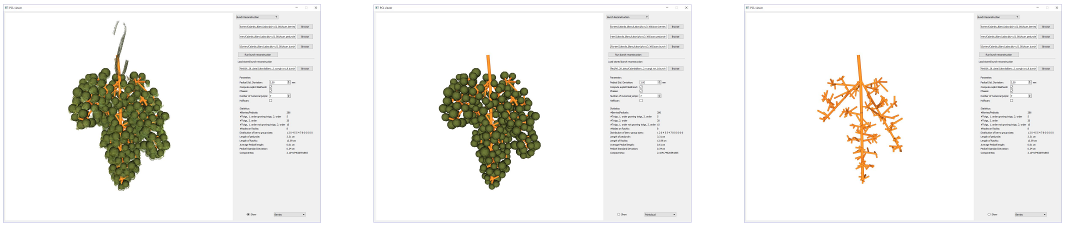
Preprocessing
Registration correction: ☐
Remove statistical outlier: ☑
Pointcloud thinning: ☑ 0,40 mm
Pause for postprocessing: ☐
Berry Detection: ☐
Overwrite Berry Files: ☐
Full reconstruction: ☐
Overwrite Bunch Files: ☐

Loading
7_05_data/todo/CalardisBlanc_2.xyzrgb.txt_0.xyzrgb
Path to directory with scans in ply format
Sorten/Calardis_Blanc/Labor/plyvc/2.360/scan.xyzrgb

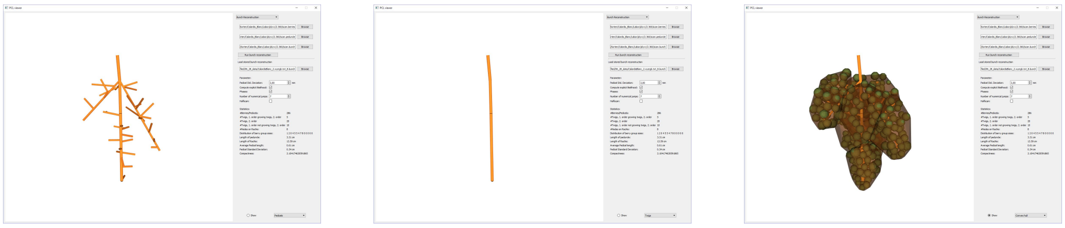
Storing
Path to output pointcloud file (in xyzrgb format)
Path to output pointcloud file (in csv format)

● Show

Figure 5.1: *The graphical user interface developed for the application.*



(a) *Pointcloud and reconstruction* ***(b)*** *Reconstruction* ***(c)*** *Reconstruction without berries*



(d) *Reconstructed rachis and twigs* ***(e)*** *Reconstructed rachis* ***(f)*** *The convex hull*

Figure 5.2: *Different grape bunch organs and the pointcloud can be removed from view if required. Figure 5.2a shows a reconstruction together with the respective pointcloud. In Figure 5.2b, the pointcloud has been removed from view. The following figures show the successive removal of other grape bunch organs from view: first the berries (Figure 5.2c), then the pedicles (Figure 5.2d), and finally the twigs (Figure 5.2e). Figure 5.2f shows the convex hull around the berries that is computed to estimate the volume of the complete grape bunch.*

the peduncle, and the bunch reconstruction, showing the respective parameters and statistical values.

5.3 Semantic labeling

As explained in Section 2.4, the manual creation of labeling gold standards is tedious. For the semantic labeling approaches introduced in Chapter 3, we present different variants of training the HFPFH descriptor while reducing the amount of required training data:

- 60 % training data and 40 % validation data for an SVM,
- 20 % training data and 80 % validation data for an SVM,
- no training data by using k-means for unsupervised clustering.

For the deep learning approach, namely the SnapNet, we keep the division into 60 % training data and 40 % test data. However, it offers the possibility to train more classes for the different types of stems; therefore, it is trained on a set containing both grape bunch and stem skeleton scans.

5.3.1 Hue-Fast-Point-Feature Histograms

The HFPFH descriptor is applied to the labeled grape bunch scans from the *cultivar set* and the *table grape set* separately. We divide the points in the pointclouds into two classes: one includes points that lie on the surfaces of the hook and the stems. The second contains points lying on the surfaces of the berries. In the scans from the *cultivar set*, the stem system is mostly occluded, thus, the number of berry points is significantly larger than the number of stem and hook points. In the *table grape set*, the same effect is visible to a lesser extent. This data set contains grape bunches with a wider range of berry colors and shapes. Additionally, larger parts of the stem system are visible. Therefore, this data set is on the one hand more challenging and on the other hand provides the possibility to test the effects of the color component of the descriptor. We use an SVM on the *cultivar set* and the *table grape set* and k-means only on the *table grape set*, as this set contains a larger number of stem points. The results are presented in the following.

SVM

For the evaluation, we divide the set of labeled pointclouds randomly into a training and a validation set. As mentioned in Section 2.4 we created labeling ground truths for 19 grape bunches from the *cultivar set*.

The first training set contains 60 % of the grape bunches, corresponding to 11 scans (2

Calardis Blanc, 3 Dornfelder, 3 Pinot Noir, and 3 Riesling). The remaining 8 grape bunches (40 %) make up the validation set.

The parameters are chosen empirically based on observations made in Mack et al. (2017a) and Mack et al. (2018). We set $r_n = 0.5$ mm and $r_h = 4.5$ mm for the normal calculation of the FPFHs in the descriptor and the binsizes b_s and b_c both to 33, leading to a total binsize of $b = 66$.

We use a linear SVM with standard parameters from the *svm-light* library (Joachims, 1999) and achieve a F1-score of 99.12 %. While this shows that the semantic labeling achieves almost perfect results, the creation of a labeling ground truth is time-intensive. Therefore, we created an additional training set containing only 20 % of the scans, i.e., one randomly chosen grape bunch of each cultivar. All 15 other grape bunches (80 %) were added to the new validation set. This combination resulted in an only slightly lower F1-score of 99.10 %. This shows that only a relatively small amount of training data is necessary to reach stable results. Similar experiences were reported in Mack et al. (2017a) on pointclouds generated with a 3D-laserscanner on grape bunches.

On the *table grape set*, we directly train on only one randomly chosen grape bunch per cultivar, leading again to a division into a training set with about 20 % of the data and a validation set with 80 %. As the berry shapes vary strongly and larger parts of the stem skeleton are visible, this data set is more challenging. Still, we achieve a F1-score of 94.05 %.

k-means

On the *table grape set* we evaluate the results of a k-means classification with $k = 2$. As the number of berry points is larger in all pointclouds, we assign the label ”berries” to the larger resulting cluster.

To validate our assumption that the hue channel is suitable to distinguish between objects of different colors, not different shades, we first evaluate the results achieved using only the different color channels, as depicted in Figure 5.3. We show the combined results on the whole data set and, additionally, the results on grape bunches with green berries and those with red berries. On the grape bunches with green berries, the hue channel clearly outperforms all other feature channels (88.10 ± 12.45 %), while the green channel leads to slightly better results on grape bunches with red berries (78.25 ± 17.61 %). Grape bunches with green berries often have a green stem system as well, making the hue channel the best choice to distinguish between the stems and the berries. The color difference between stem system and berries is usually larger for grape bunches with red berries; therefore, better results can be achieved with the green channel. On the combined data set, the hue channel performs best with 78.69 ± 17.75 %, followed by the green channel with 70.46 ± 18.37 %.

Based on these results, we combine the channels that showed the best results so far, namely hue and green channels, with the FPFHs. Figure 5.4 presents the respective F1-values.

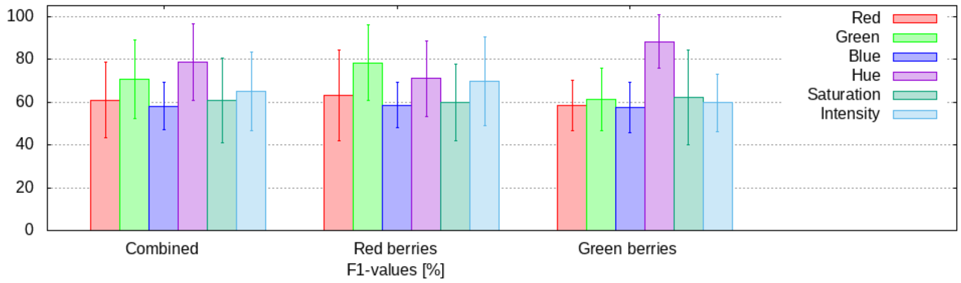


Figure 5.3: *The results achieved for the semantic labeling of the table grape set with k-means with only the separated red, green, blue, hue, saturation, and intensity channel.*

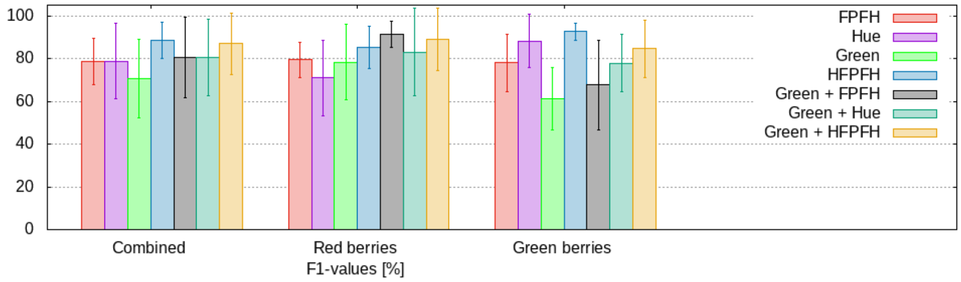


Figure 5.4: *The results achieved for the semantic labeling of the table grape set with k-means with a combination of the green or the hue channel and FPFHs.*

Again, the best semantic labeling on grape bunches with red berries can be achieved with a combination of the green channel and the FPFH descriptor (91.36 ± 6.27 %), but combining the hue channel with the FPFHs delivers results of 85.30 ± 9.75 % as well. For grape bunches with green berries, the green channel yields worse results than the FPFHs alone (67.68 ± 20.93% to 77.95 ± 13.35 %). The HFPFH descriptor is stable on all grape bunches, delivering a F1-value of 88.61 ± 8.54 % and is, thus, considered the best descriptor in this context.

We compare the results of the HFPFH descriptor with those of the CSHOT (Tombari et al., 2011), yielding a F1-value of 76.18 ± 4.97 %, and the PFHRGB (Rusu et al., 2008), delivering a F1-value of 80.84 ± 7.40 %, as shown in Figure 5.5. Our combination outperforms both in all cases with the final result of 88.61 ± 8.54 %, confirming our assumption that the hue channel is especially suitable if one class contains instances of objects with different shadings, but the same underlying color.

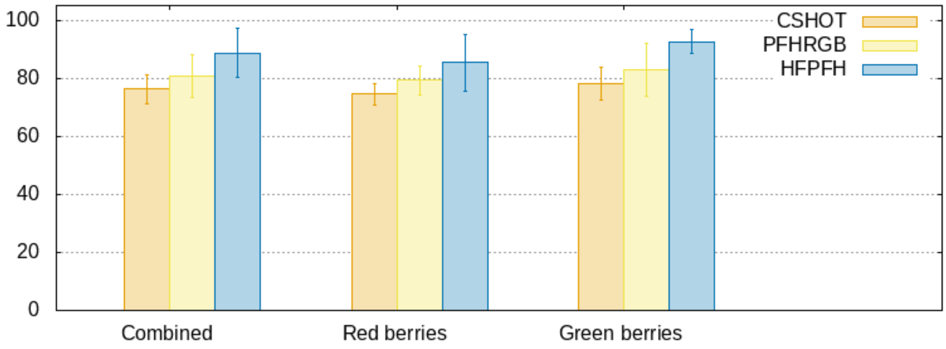


Figure 5.5: *The results achieved for the semantic labeling of the table grape set with k-means combined with the HFPFH descriptor compared to the PFHRGB and CSHOT descriptors.*

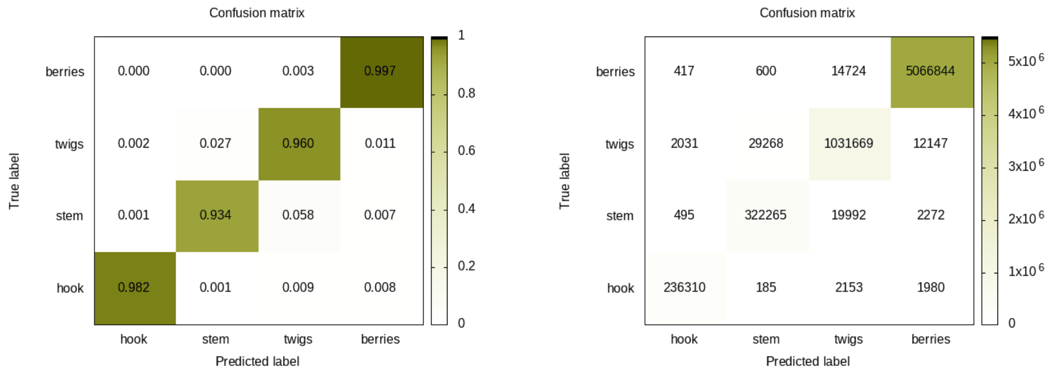
5.3.2 Deep learning

To adjust the SnapNet to the application on grape bunches, we empirically evaluated the respective parameters. As mentioned in Section 3.3, views are generated by randomly choosing a point of the pointcloud and drawing a line through this point. On this line, three points in fixed distances from the point are used as camera positions, facing always towards the point itself. We set the three fixed distances to $dv_1 = 20\,\text{mm}$, $dv_2 = 40\,\text{mm}$, and $dv_3 = 60\,\text{mm}$. The azimuth of the view is chosen uniformly at random between $av_{min} = 0°$ and $av_{max} = 360°$, and the elevation between $ev_{min} = -90°$ and $ev_{max} = 90°$, respectively.

The deep learning approach is applied to all scans of grape bunches and grape bunch stem skeletons with labeling ground truth from the *cultivar set*. We use a similar division as for the semantic labeling with the SVM: the training set contains all in all 11 labeled grape bunch scans (2 Calardis Blanc, 3 Dornfelder, 3 Pinot Noir, and 3 Riesling) and 8 labeled stem skeleton scans (2 Calardis Blanc, 3 Dornfelder, and 3 Pinot Noir). The validation set contains the remaining 8 grape bunch scans (2 of each cultivar) and 6 stem skeleton scans (2 each of Calardis Blanc, Dornfelder, and Pinot Noir). The number of classes is extended to berries, twigs (combining twigs, subtwigs, and pedicles), hook, and stem (combining the peduncle and the rachis). As mentioned in Section 2.1, the separation of peduncle and rachis is not clearly defined. While we train the classes separately, we evaluate them as one class.

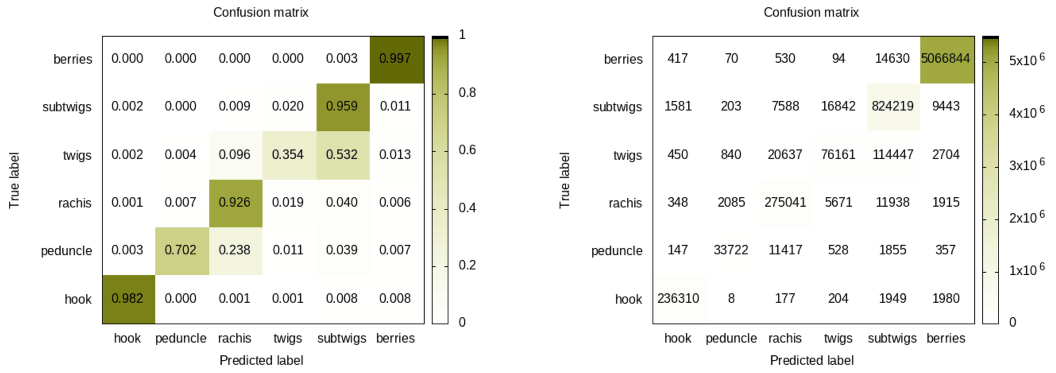
Figure 5.6 shows the confusion matrix. As can be seen the approach yields results of more than 93 % accuracy for all classes.

As shown in Section 2.1, the grape bunch structure usually shows a further division of the twig class into twigs, subtwigs, and pedicles. While in grape bunch scans, the only visible parts of the skeleton are usually the peduncle, sometimes parts of the rachis, and few twigs, the stem skeleton scans provide the possibility of a further division into a class combining subtwigs and pedicles and a twig class. We expect that distinguishing between subtwigs and



(a) *Relative accuracy* **(b)** *Absolute numbers*

Figure 5.6: *Results achieved for the semantic labeling of grape bunches and grape bunch stem skeletons from the cultivar set, showing the absolute numbers of points assigned to each label (5.6b) and the relative accuracies (5.6a).*



(a) *Relative results* **(b)** *Absolute results*

Figure 5.7: *Results achieved for the semantic labeling of grape bunches and grape bunch stem skeletons from the cultivar set with division of the twig class into twigs and subtwigs, showing the absolute numbers of points assigned to each label (5.7b) and the relative accuracies (5.7a). Twig points are often misclassified as subtwig points, as the amount of training data is relatively low (214 K twig points). The peduncle is the uppermost part of the rachis, thus, peduncle points are often assigned to the rachis class, instead.*

pedicles is not feasible, as there are usually only very few short subtwigs visible and they are hard to identify even for the human eye. Therefore, we did not further divide the classes. Figure 5.7 shows the results for such a larger number of classes. For further information, we also divide the stem class into peduncle and rachis. As can be seen, the true positive values stay at more than 93 % for all classes beside the twigs and the peduncle. Only a relatively small number of 35 % of twig points are correctly assigned to their class, the rest is labeled as subtwig. We can trace this to the fact that twigs are often occluded by subtwigs and pedicles in the views, thus, they are not as well represented in the data. Figure 5.7b shows a number of 214 K twig points, which is relatively low. While the number of hook points (240 K) and stem points (345 K) is not significantly larger, those classes show clear features, e.g., the stem is always oriented vertically and follows a relatively straight line, and the hook as an artificial object always has the same shape in all scans. The twigs do not show such clearly defined attributes, thus, more training data would be required to robustly distinguish them from subtwigs.

The rachis points keep the same labeling quality as the combined stem class, but only 70 % of the peduncle points are correctly labeled. 23 % of the missing points are assigned to the rachis class instead. As mentioned before, this is due to the unclear definition. Still, this does not lead to problems, as even if all peduncle points would be wrongly labeled as rachis points, it would be possible to label the highest part of the rachis as peduncle points.

A discussion of the semantic labeling results and a comparison between the descriptor-based and deep learning-based methods presented in this thesis will be given in Section 5.5.

5.4 Grape bunch reconstruction

We evaluate the steps of the bunch reconstruction divided into the berry detection, the initial hypothesis, and the final reconstruction.

5.4.1 Parameters

If not noted differently, all parameters in this section were derived empirically.

Berry detection

For the detection of berries, several parameters are required, as mentioned in Section 4.3.4:

1. $t_s = 15°$: the smoothness threshold for region growing,
2. $t_c = 3.0$: the curvature threshold for region growing,
3. $\epsilon = 0.05$ mm: the maximal distance between a point and the surface of the model to be counted as part of the consensus set by RANSAC,
4. $n_r = 500$: the number of iterations that RANSAC is allowed to use,
5. $t_i = 70$: the threshold on the number of points in the consensus set to consider a berry hypotheses valid,
6. $r_{min} = 1$ cm, $r_{max} = 9$ cm: the minimal and maximal berry radii

Some of these values can be derived from the data, as described in Mack et al. (2017a). The minimal and maximal berry radii r_{min} and r_{max} are set very loosely, such that r_{min} is smaller and r_{max} larger than any berry in the grape bunches in our data sets. The reason for this is that some berries are deformed, making them seem almost flat in the data. Therefore, RANSAC finds a basis for fitting a huge berry that covers the flat points very well, as the curvature gets very small. As such a berry cannot occur, we prevent the creation of such hypotheses.

Initial hypothesis

For the computation of the initial hypothesis, k-means is employed to cluster the berry centers. The resulting cluster centroids are then used as potential twig endpoints. The parameter k has to be derived such that it gives a good starting point for the number of twigs. For this, we set it as follows:

$$k = n_b / ((V_b \cdot V_c) \cdot w_t) \tag{5.1}$$

with n_b being the number of berries, V_b the summed up volume of all berries, and V_c the volume of the convex hull around the grape bunch. $w_t = 2$ is a weighting constant. To restrict the number to a reasonable range, we make sure that the condition $k_{min} = 20 \leq k \leq k_{max} = 35$ holds.

RJMCMC optimization

The RJMCMC optimization requires the definition of the annealing factor $c = 0.999$ and the end temperature $t_f = 10^{-5}$. The probability distribution $j(xx')$ is chosen such that each jump is assigned the same probability.

Most constraints work with a normal distribution, requiring a standard deviation σ that is used to set the strictness of the constraint. Table 5.1 shows the choice of those parameters. We allow a 10° deviation for the angles between rachis and twigs (angle constraint $G_1^{(1)}$), or

	$G_1^{(1)}$	$G_1^{(2)}$	$G_2^{(1)}$	$G_3^{(1)}$	$G_4^{(1)}$	$G_4^{(2)}$	$G_4^{(3)}$	$G_5^{(1)}$	$T_3^{(1)}$	$T_4^{(1)}$
σ	10°	10°	1	5	10 mm	3 mm	3 mm	1 mm	10 mm	0.5

Table 5.1: *The standard deviations derived empirically for the normal distributions of the angle constraint between rachis and twigs $G_1^{(1)}$, the angle constraint between twigs, subtwigs and pedicles $G_1^{(2)}$, the similarity constraint $G_2^{(1)}$, the overlap constraint $G_3^{(1)}$, the length constraints applied to twigs $G_4^{(1)}$, subtwigs $G_4^{(2)}$, and pedicles $G_4^{(3)}$, the deviation constraint $G_5^{(1)}$, the decreasing length constraint $T_3^{(1)}$, and the twig endpoint constraint $T_4^{(1)}$. The connection and potential connection constraints $T_1^{(k)}$ and $T_2^{(k)}$ control the connections between plant organs and need no additional parameters.*

twigs, subtwigs, and pedicles (angle constraint $G_1^{(2)}$). In some cases, especially the upper twigs are drawn downwards by the weight of the berries. Therefore, we allow some deviation, while still keeping the majority of the twigs at about 90° angle. The constraint responsible for avoiding similar twigs $G_2^{(1)}$ is set relatively strict to 1 to make sure that such twigs are combined. The constraint on overlapping parts $G_3^{(1)}$ is chosen less strict with 5. The reason for this is that in some cases overlap appears by approximating curved twigs by straight ones. While the algorithm should still reward hypotheses that avoid such overlap, setting the standard deviation too strict might make this constraint useless, as a minimal probability would be returned for all hypotheses, because the number of overlapping parts in the initial hypothesis is too large.

For the length constraints, the different sizes of components have to be taken into account. Setting the standard deviation for the constraint restricting twig lengths $G_4^{(1)}$ to 10 mm rewards smaller twig lengths, but still allows the algorithm to add a new twig where necessary. Subtwigs and pedicles on the other hand (constraints $G_4^{(2)}$ and $G_4^{(3)}$) are comparably short; therefore, the standard deviation has to be chosen stricter. Pedicle lengths usually show only a small standard deviation, hence, we set the respective deviation constraint $G_5^{(1)}$ rather strict to 1.0 mm.

The topological constraints T_1 and T_2 control the connections between elements. They do not require additional parameters. The decreasing length constraint T_3 that makes sure that twig lengths decrease from top to bottom could be implemented with a *similar* relation with the mean value μ set to 0. However, this can be too strict, as in the initial hypotheses all berries are attached to twigs and the missing subtwigs can lead to wrongly alternating twig lengths. Therefore, we use the *smaller* relation instead, but still make sure that adjusting twigs such that they decrease from top to bottom of the grape bunch is sufficiently rewarded by setting the standard deviation to 10 mm. Finally, twig endpoints outside of the convex hull can appear if twigs are elongated unnecessarily such that they are no longer inside the berries. Such a hypotheses should be rejected by the constraint T_4, therefore, we set it strictly to 0.5.

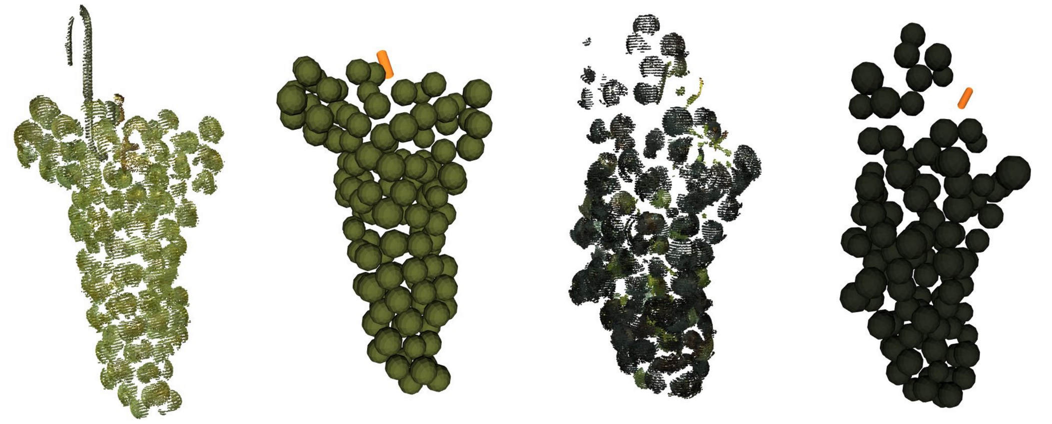
According to our modification of the RJMCMC algorithm, during the first phase each topological jump is followed by $n = 7$ geometrical jumps.

5.4.2 Berry detection

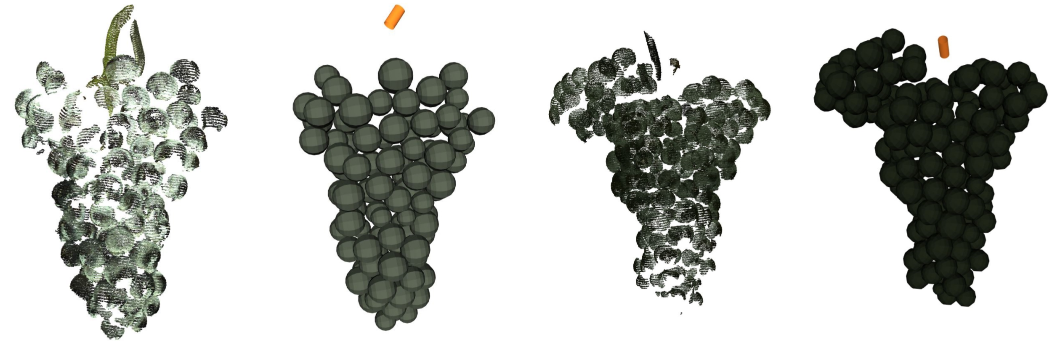
We evaluate the berry detection approach, presented in Section 4.3.4, on the *cultivar set* by comparing our results to reference data derived manually for the number of berries and their average diameters, as described in Section 2.4. Additionally, we present qualitative results for grape bunches chosen from the different cultivars.

Qualitative evaluation

Figure 5.8 shows exemplary results for the berry detection method on grape bunches from the different cultivars in the *cultivar set*. Visually, the result is in all cases very close to the original pointcloud. In the case of the Dornfelder pointcloud (Figure 5.8c and 5.8d), it is obvious that the upper left part of the grape bunch was not optimally scanned, resulting in berries that are scanned incompletely. Although most of these berries were recovered, we have to assume that some will be missing. Another factor that leads to missing berries is that some of them grow inside the grape bunch, being completely or almost completely occluded. For the examples shown in the figure, we detect 112 out of 151 berries for the Calardis Blanc grape bunch, 100 out of 126 for the Dornfelder grape bunch, 97 out of 114 for the Pinot Noir grape bunch, and 150 out of 196 for the Riesling grape bunch. For the Riesling grape bunch, it is important to note that this cultivar grows a large number of very small berries



(a) *Calardis Blanc pointcloud* ***(b)*** *Detected berries* ***(c)*** *Dornfelder pointcloud* ***(d)*** *Detected berries*



(e) *Pinot Noir pointcloud* ***(f)*** *Detected berries* ***(g)*** *Riesling pointcloud* ***(h)*** *Detected berries*

Figure 5.8: *Exemplary results achieved on grape bunches from the cultivar set for the berry detection.*

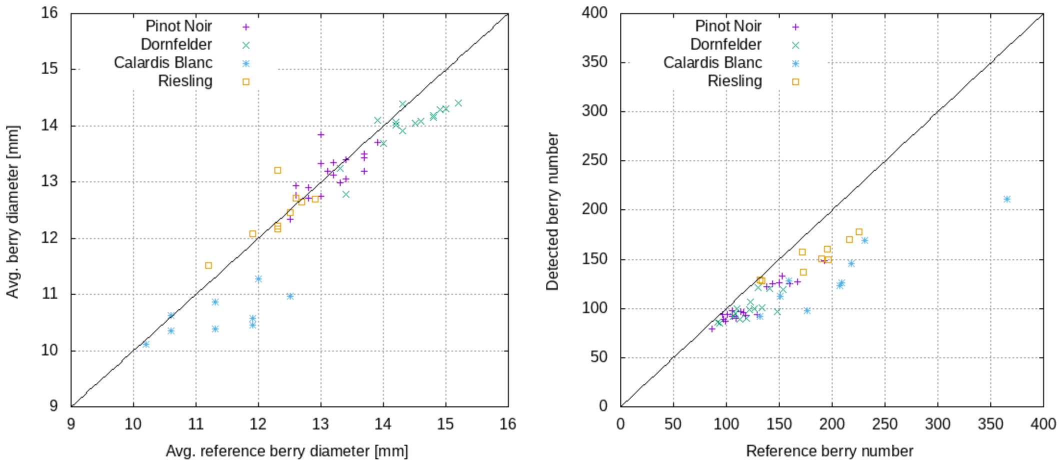
at the inside. Those berries were not included in the reference measurements, as they do not provide the required genetic information, but they are also not visible in the scan, thus, leading to no deviation of the results.

The average diameter of the berries shows only a small mean percentage error for all grape bunches (3.9 % for Calardis Blanc, 3.56 % for Dornfelder, 0.63 % for Pinot Noir and 7.4 % for Riesling). In all cases, the absolute error is less than 1 cm and for Pinot Noir even less than 1 mm.

Quantitative evaluation

The results of the quantitative evaluation can be seen in Figure 5.9.

We achieve a Pearson correlation coefficient of 0.9 for the number of berries and 0.92 for the average berry diameters. While this is a robust result, the plot shows that the number of berries is constantly underestimated. As mentioned in the qualitative evaluation, this is related to the structure of the grape bunch: while most berries are found on the outside, some grow inside the grape bunch, being occluded by other berries. Thus, we cannot detect them in the pointcloud. This effect gets stronger for larger grape bunches containing more berries, which is the case mostly for grape bunches of the Calardis Blanc cultivar. Because of the same reason, the average diameters are underestimated for Calardis Blanc grape bunches, still, there is a high correlation.



(a) *Average berry diameters*

(b) *Number of berries*

Figure 5.9: *Results achieved on the cultivar set for the berry detection, covering the average berry diameters (Pearson correlation coefficient 0.92) and the number of berries (Pearson correlation coefficient 0.9). The number of berries is underestimated, as berries growing at the inside of the grape bunch are not visible in the pointcloud.*

Elliptical berries

For elliptical berries, we start with spherical berries and then apply a least square fitting, as described in Section 4.3.4. While the spherical berries in the *cultivar set* are defined by one diameter, the elliptical berries in the *table grape set* require two diameters. We name them in Figure 5.10. The diameter relating to the furthest elongation of the berry starting at the pedicles is called ”length”, and the diameter orthogonal to it ”width”. For the used data set it always holds that the width diameter is smaller than the length diameter. We apply the detection to the pointclouds after the semantic labeling with k-means introduced in Section 5.3.1.

Figure 5.11 shows exemplary results on the table grapes depicted in Figure 2.4. Especially in the Ruby Seedless grape bunch in Figure 5.11d, few berries are not detected, as they are only partially included in the pointcloud. Still, for Sultana 54 out of 55 berries are retrieved, for Prime 76 out of 78, for Sugrathirteen 88 out of 90, and for Ruby 45 out of 48. The reason

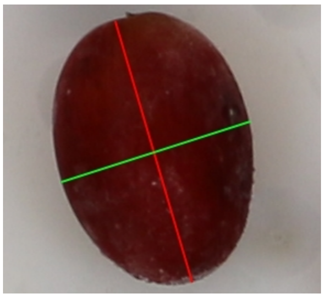


Figure 5.10: *This figure shows the lengths considered for the evaluation of elliptical berries: the width is depicted by a green line and the length by a red line.*

(a) *Sultana* ***(b)*** *Prime* ***(c)*** *Sugrathirteen* ***(d)*** *Ruby Seedless*

Figure 5.11: *Results achieved on the table grape set when detecting the elliptical berries with least square fitting on the pointclouds generated for the grape bunches in Figure 2.4. The pointclouds are left visible for a better overview over the quality of the least square fitting.*

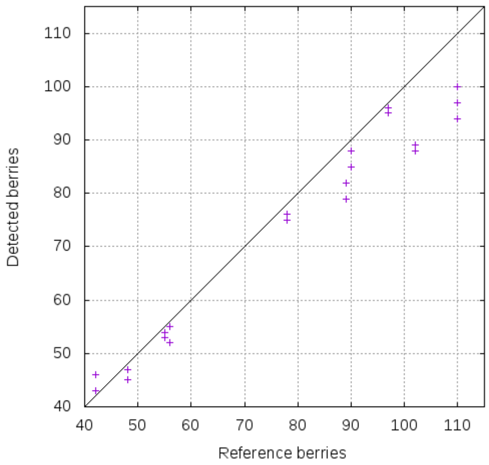


Figure 5.12: *The results achieved for the number of detected berries on the* table grape set *(Pearson correlation coefficient 0.99).*

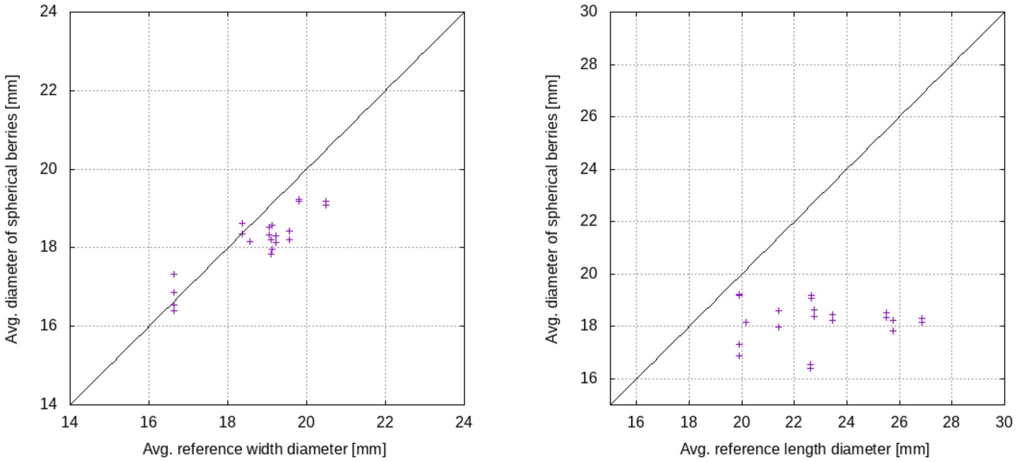
for this is that table grapes have a rather loose architecture with large berries, thus, most of them are visible and can be detected.

The ellipticity of the berries in the pointcloud is usually represented well, with few deviations, e.g., in the lower part of the Prime grape bunch (Figure 5.11b). The low mean percentage error of the length diameters validates this: for Sultana, it is 4.40 %, for Prime 0.73 %, for Sugrathirteen 8.61 %, and for Ruby 2.84 %.

The results for the number of berries on the *table grape set* can be found in Figure 5.12. The Pearson correlation here is with 0.99 even better than on the *cultivar set*, which is mostly due to the fact that the table grape bunches mostly have comparably few berries, ranging between about 40 and about 110 instead of 80 and 370 on the *cultivar set*. Thus, the underestimation effect on this data set is relatively small.

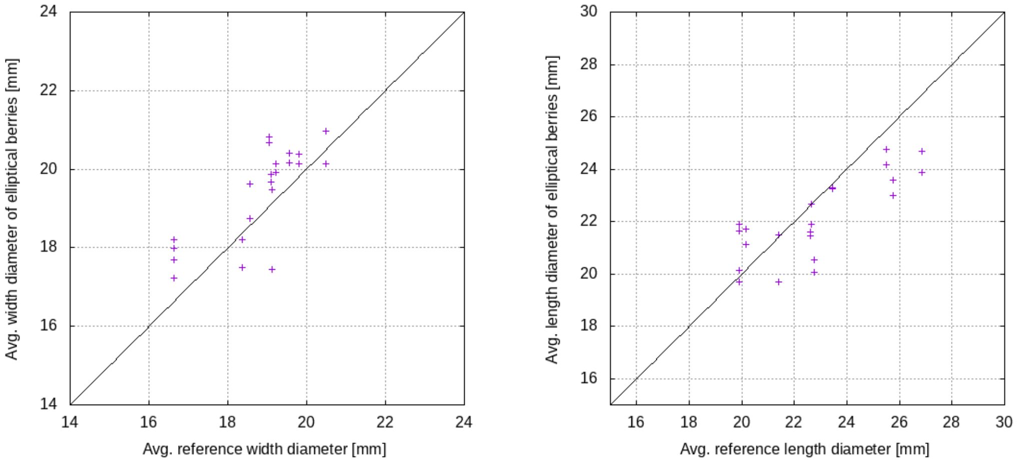
Figures 5.13a and 5.13b show the results achieved when representing the berries with spheres. The spheres are fitted into the data such that they cover the largest amount of points possible. Thus, the width is covered with a Person correlation coefficient of 0.9. The length of elliptical berries differs from the width, leading to an underestimation of this diameter and showing hardly any correlation with a coefficient of 0.04. This shows that elliptical berries cannot be represented by spheres with a sufficient quality.

Extending the method with a least squares fitting yields the results presented in Figures 5.13c and 5.13d. The length diameter is clearly improved with a Pearson correlation coefficient of 0.8. The results for the width diameter on the other hand are slightly worse, showing a Pearson correlation coefficient of 0.77. The shape of elliptical berries offers more points that support the fitting of the length diameter and less for the width. Therefore, the fitting of the width can be less exact. Still, this problem can be easily solved by combining the width diameter achieved with the sphere detection with the length diameter from the least squares



***(a)** Avg. width diameter for spheres*

***(b)** Avg. length diameter for spheres*



***(c)** Avg. width diameter for ellipses*

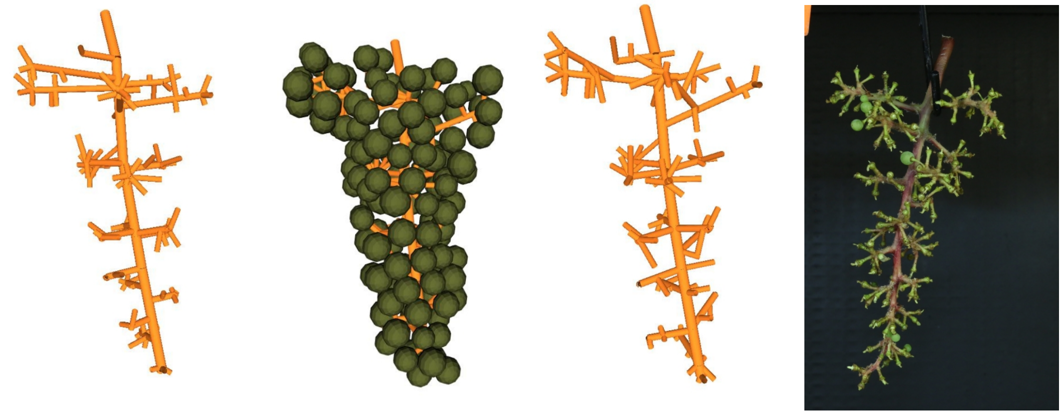
***(d)** Avg. length diameter for ellipses*

Figure 5.13: *Results achieved on the* table grape set *when detecting the elliptical berries with spheres (Figures 5.13a and 5.13b) or ellipses (Figures 5.13c and 5.13d). For spheres, the width parameter results in a Pearson correlation coefficient of 0.9 (Figure 5.13a) and 0.04 for the length (Figure 5.13b), as spheres can only cover one diameter of the elliptical berries and the width is supported by a larger number of points. For ellipses, the width parameter results in a Pearson correlation coefficient of 0.77 (Figure 5.13c) and 0.8 for the length (Figure 5.13d).*

fitting.

5.4.3 Reversible-Jump-Markov-Chain-Monte-Carlo optimization

In this section, we evaluate the RJMCMC optimization step as introduced in Section 4.3.

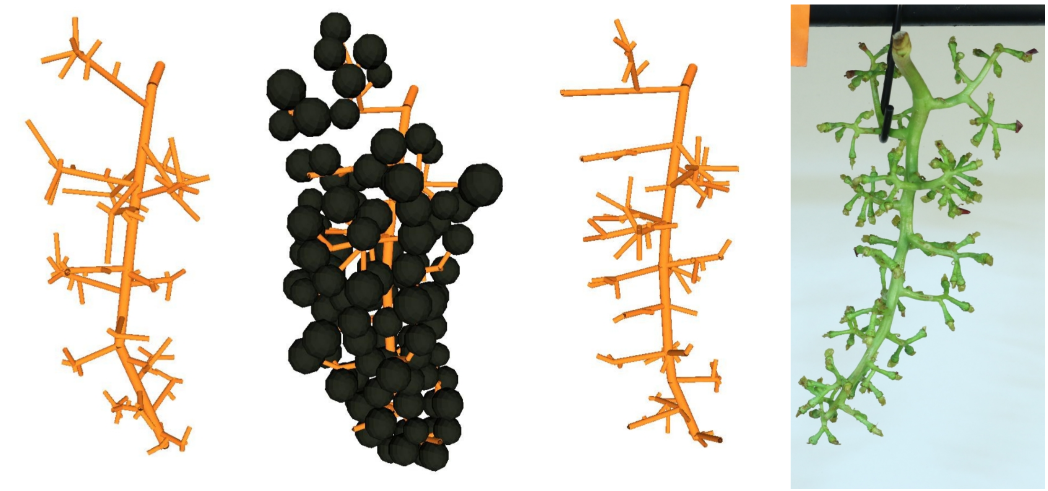


*(**a**) Calardis Blanc initial hypothesis*

*(**b**) Final reconstruction*

*(**c**) Final reconstruction without berries*

*(**d**) Stem skeleton*



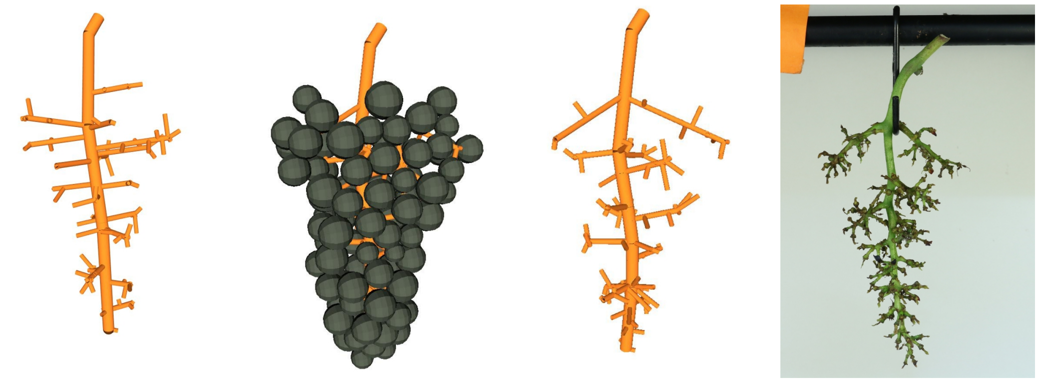
*(**e**) Dornfelder initial hypothesis*

*(**f**) Final reconstruction*

*(**g**) Final reconstruction without berries*

*(**h**) Stem skeleton*

Figure 5.14: *Exemplary results achieved on Calardis Blanc and Dornfelder grape bunches from the cultivar set for the grape bunch reconstruction.*



*(**a**) Pinot Noir initial hypothesis* *(**b**) Final reconstruction* *(**c**) Final reconstruction without berries* *(**d**) Stem skeleton*

*(**e**) Riesling initial hypothesis* *(**f**) Final reconstruction* *(**g**) Final reconstruction without berries* *(**h**) Stem skeleton*

Figure 5.15: *Exemplary results achieved on Pinot Noir and Riesling grape bunches from the cultivar set for the grape bunch reconstruction.*

Qualitative evaluation

Figures 5.14 and 5.15 show the results achieved on different grape bunches from the *cultivar set*. On all grape bunches it can be seen that the number of twigs is reduced from initial to final hypothesis and the pedicles are more evenly spread over the twigs. This effect is especially strong on the Pinot Noir grape bunch (Figures 5.15a to 5.15c) and also mirrored in the numbers: for the initial hypothesis, the average pedicle lengths are underestimated for all grape bunches, for Calardis Blanc by 13.73 %, for Dornfelder by 10.53 %, for Pinot Noir by 21.95 %, and for Riesling by 37.77 %, leading to an absolute error between 7 and 22 mm. With the optimization, this error is reduced to 4.59 % for Calardis Blanc, 6.12 % for Dornfelder, 1.47 % for Pinot Noir, and 2.79 % for Riesling, or an absolute error of less than 5 mm.

The length of the stem is not changed by jumps, thus, it remains almost stable during the optimization. Both initial and final hypotheses show relative errors of 4.26 % for Calardis

Blanc, 26.66 % for Dornfelder, 11.14 % for Pinot Noir, and 15.69 % for Riesling. The absolute errors are 0.56 cm for Calardis Blanc, 3.87 cm for Dornfelder, 1.56 cm for Pinot Noir, and 2.01 cm for Riesling. While this error seems high compared to the pedicle lengths, there is one important factor that comes into play for the stem length: the reference measurements are taken in images, meaning that distorted angles are an obvious source for errors. While this holds for pedicle lengths as well, they are small and measurements are taken for a large number of them, leading to a smoothing of the error. In the case of Dornfelder, showing the largest error, the image of the stem skeleton in Figure 5.14h makes it seem likely that the stem of the grape bunch has a relatively strong backwards curvature. Additionally, the peduncle is crooked, pointing towards the camera, and is thus missing in the measurement. Tests on the *variety set* showed that such distorted angles can lead to deviations of up to 4 cm. Another problem that occurs for almost all grape bunches is that the lower end of the rachis is usually completely occluded, making it impossible to see and detect it in the data. Thus, we have to use heuristics. Finally, some grape bunches were not optimally scanned at the lower end, as can be seen, e.g., for the Riesling grape bunch in Figure 5.8g. When berries in this part of the pointcloud are missing, this has a critical effect on the length of the rachis that can then be underestimated.

Because of these error sources, both over- and underestimation can occur for the length of the stem.

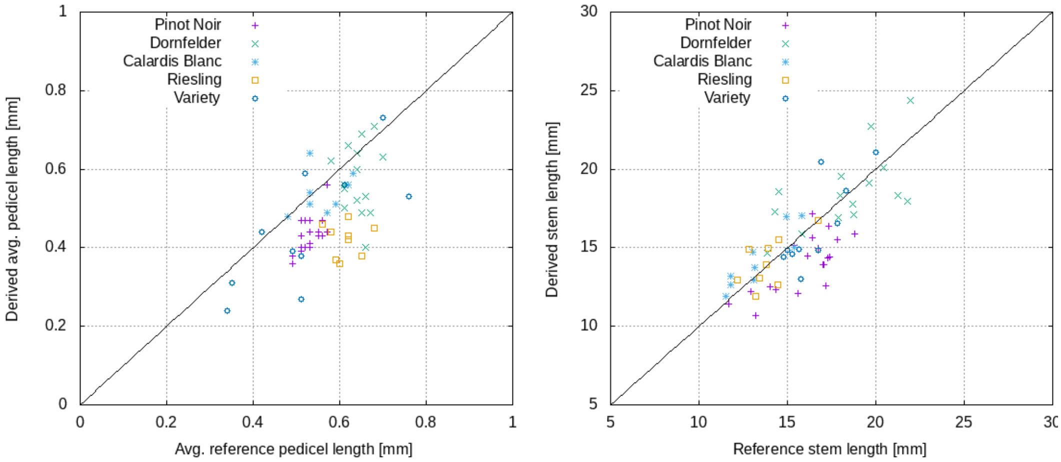
Quantitative evaluation

We evaluate the RJMCMC optimization quantitatively by comparing the average results from 10 independent runs on the *cultivar* and the *variety set* to those derived manually, as described in Section 2.4. First, we show the results for the initial hypotheses, and then the improvement for the final reconstructions.

Figure 5.16 depicts the results achieved with the initial hypotheses. While the stem length shows a Pearson correlation coefficient of 0.76, the coefficient for the average pedicle lengths is significantly lower with 0.59. This confirms the assumption made during the qualitative evaluation that the initial hypotheses contain too many twigs, leading to very short pedicles.

As can be seen in Figure 5.17, this changes for the final reconstructions. The optimization combines twigs where necessary and leads to longer pedicles during the first phase. During the second phase, only the start and endpoints of twigs are moved, leading again to a slight decrease of pedicle lengths. Finally, we achieve a Pearson correlation coefficient of 0.71 that is still not optimal, but considered a stable result in this context.

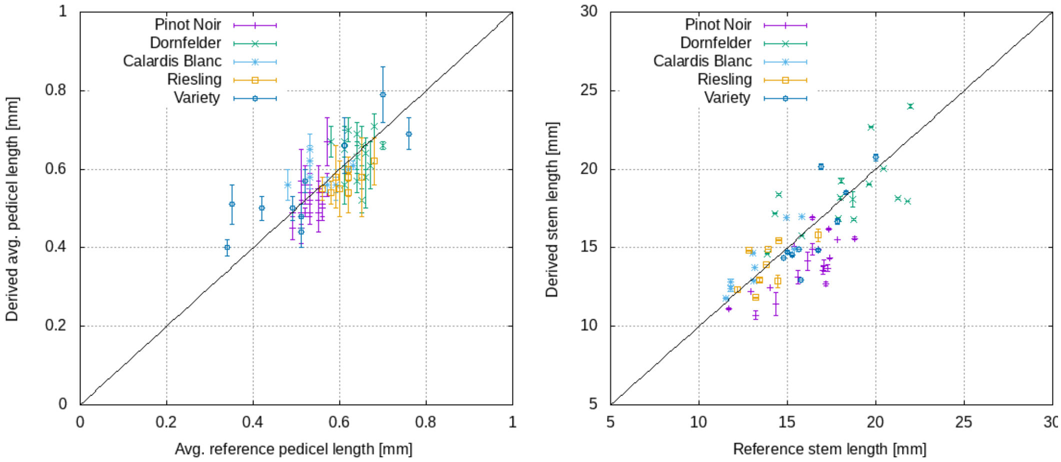
As there is no jump that adjusts rachis or peduncle, the stem lengths remain stable with a Pearson correlation coefficient of 0.76 for initial hypotheses and final reconstructions. The plot shows that, as explained in the qualitative evaluation, both over- and underestimation of the stem length occur because of the distortion of angles in the reference measurements and



***(a)** Average pedicle lengths*

***(b)** Stem lengths*

Figure 5.16: *Results achieved on the cultivar and variety set for average pedicle length (Pearson correlation coefficient 0.59) and stem length (Pearson correlation coefficient 0.76) of the initial hypothesis. Too many twigs are inserted, leading to an underestimation of the pedicle lengths. This is corrected in the final reconstruction (Figure 5.17).*



***(a)** Average pedicle lengths*

***(b)** Stem lengths*

Figure 5.17: *Results achieved on the cultivar and variety set for average pedicle length (Pearson correlation coefficient 0.71) and stem length (Pearson correlation coefficient 0.76) of the final hypothesis.*

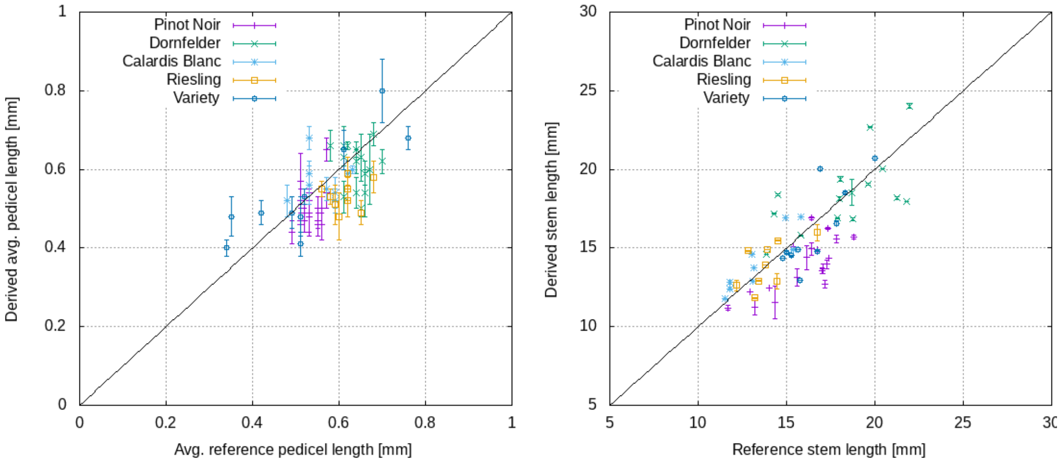
missing berries at the end of the grape bunch. Still, for most grape bunches the error remains less than 1.5 cm.

Partitioning into phases

The effects of the modification of the RJMCMC optimization introduced in Section 4.3.3 are shown by comparing the results introduced in the section before using the modification with the results of the original algorithm.

Figure 5.18 shows the results in comparison to the reference data. Again, the stem lengths change only slightly as there is no explicit jump to adjust the rachis or peduncle. For the pedicle lengths, the Pearson correlation coefficient decreases to 0.65. The effect is also visible in the plot mainly for Riesling grape bunches. For this cultivar, the pedicle lengths are underestimated, showing that without the local optimization, there are only few topological jumps accepted. Thus, the underestimation of pedicle lengths visible for the initial hypothesis (Figure 5.16a), especially for Riesling grape bunches, cannot be sufficiently compensated.

Figure 5.19 shows the acceptance rate of jumps for the optimization with partitioning into phases and without it. For all topological jumps and over all cultivars, the acceptance rate increases using the modified algorithm with the partitioning into phases. It can also be seen that the jumps with highest acceptance rates are *Combine jumps*, *Attach twig as subtwig*, and *Reassign berry*. This fits to the observation that the initial hypotheses usually show too many similar twigs that have to be combined or transformed into subtwigs. If a twig is changed in such a way, it influences the lengths of pedicles attached to it. In some cases, moving the pedicle to another twig, i.e., reassigning the berry, decreases the pedicle length. Therefore, the higher acceptance rate for combining twigs and attaching twigs as subtwigs leads to a higher number of *Reassign berry*-jumps.



(a) *Average pedicle lengths*

(b) *Stem lengths*

Figure 5.18: *Results achieved on the cultivar and variety set for average pedicle length (Pearson correlation coefficient 0.65) and stem length (Pearson correlation coefficient 0.76) of the final hypothesis using the unmodified RJMCMC algorithm.*

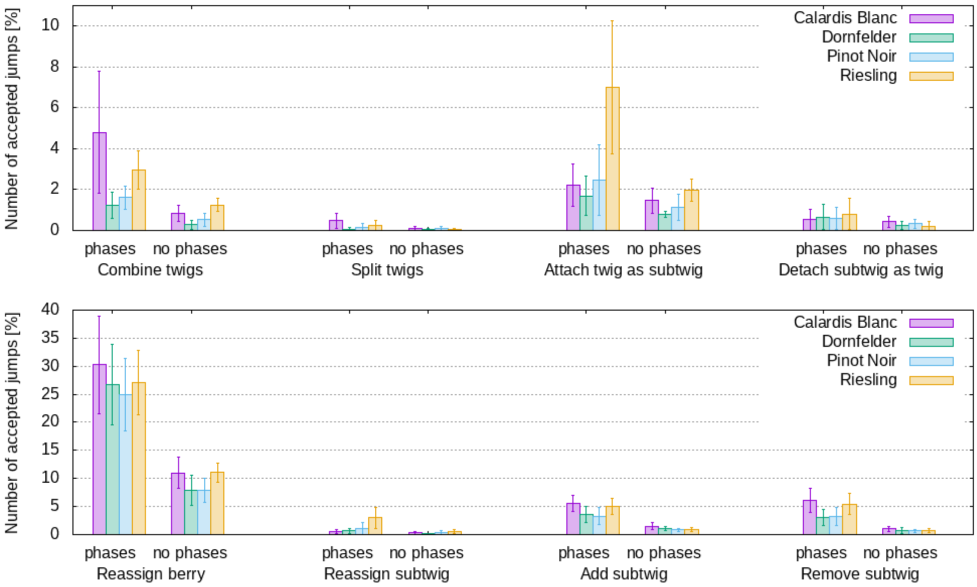


Figure 5.19: *Acceptance rates of jumps for the optimization with partitioning into phases and without it.*

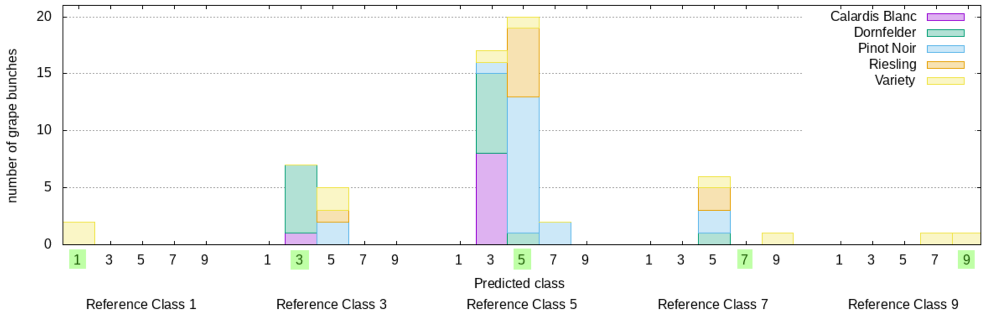
On average, using the modified RJMCMC optimization increases the acceptance rate for topological jumps between 60 and 75 %.

Compactness

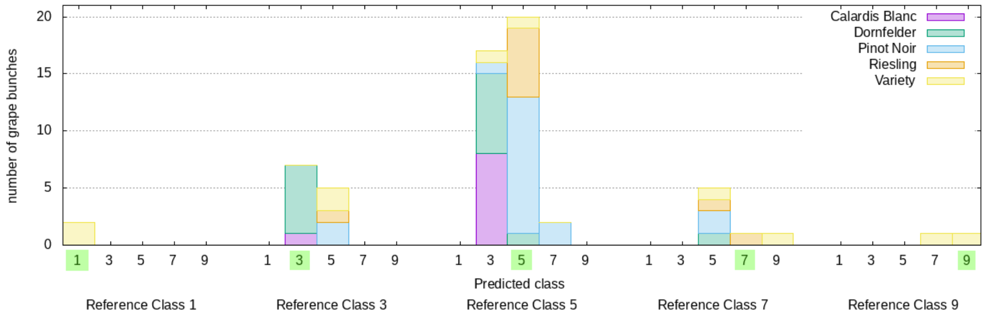
As mentioned in Chapter 1, the compactness is an important trait in the phenotyping of grape bunches. In collaboration with with our project partners at the Institute for Grapevine Breeding Geilweilerhof in Siebeldingen we developed a new formula to assign one of the five classes defined in the OIV list (Organisation Internationale de la Vigne et du Vin, 2009) to a grape bunch reconstruction x:

$$\text{compactness}(x) = \frac{V_s(x) + V_b(x)}{V_c(x)}. \tag{5.2}$$

Here, V_s is a function that derives the summed up volume of all cylinders making up the stem system, i.e, of peduncle, rachis, twigs, subtwigs, and pedicles. V_b sums up the volume of all berries. Finally, V_c represents the volume of the convex hull around the grape bunch. We compute this convex hull from the center points of the berries. As a first step, a k-means clustering with $k = 10$ is applied to the center points. Then, a convex hull volume is computed for each of the clusters. The reason for this partitioning is that because of sidebunches and



(a) *Initial compactness*



(b) *Final compactness*

Figure 5.20: *Results achieved on the cultivar and variety set for the compactness of the initial and final hypothesis.*

shoulders the convex hull of the complete grape bunch can stretch over empty spaces. This effect is reduced with the clustering.

After computing the compactness value, we use thresholds to divide them into the 5 OIV classes: compactness values between 0 and 2.0 are assigned to class 1, values between 2.0 and 3.5 to class 3, 3.5 to 4.5 to class 5, 4.5 to 4.8 to class 7, and everything above 4.8 to class 9. The ranges are non-linear, as is the relation between the volumes: for two grape bunches with same volume of the convex hull, but different compactness, the more compact grape bunch has a larger volume of berries and stems. But because of the nature of the grape bunch, this development has to be less than linear, as the berries need sunlight for their growth, thus, they have to be positioned on the outside of the grape bunch, limiting the possible volume.

Applying formula 5.2 to the initial and final hypotheses of the RJMCMC optimization leads to the results depicted in Figure 5.20. As can be seen, for both initial and final hypotheses, there is no deviation of more than one class from the expert opinion. For the final hypotheses, 50 % of the grape bunches (31 out of 62) are assigned the correct class, 37.10 % (23 grape bunches) are assigned to one class below and 12.9 % (8 grape bunches) to one class above the expert estimate. As this is a subjective measurement and deviations of one class happen regularly between experts as well, our estimation here is comparable to that of an expert.

5.5 Discussion

Grape bunches are a challenging application field as they vary strongly in shape and sizes. The structure of a stem skeleton is complex, but subject to the rules imposed to them by their biology.

Semantic labeling: We showed a semantic labeling of the pointclouds with a new RGB-D descriptor on the one hand, and a deep learning method using the SnapNet on the other hand. Even with only one grape bunch per cultivar as training data, an SVM trained on the features derived with the HFPFH descriptor achieves F1-values of more than 99 % on the *cultivar set*, more than 94 % on the *table grape set*, and of more than 88 % with k-means clustering on the *table grape set*. While the results with k-means are slightly lower, this method offers the advantage that no training data is required.
The SnapNet as a supervised method can be directly compared to the SVM by summing up the results for the detection of stem, hook, and twigs, as they correspond to the stem class used by the SVM. This leads to an F1-value of 99.68 %, thus, both methods are able to solve this classification task very well. The similarity of different parts of the stem skeleton, like the rachis or twigs, makes it hard to distinguish them with the HFPFH descriptor relying on shape and color. It would be possible to train an additional SVM for the classes ”stems” and ”hook”. As the surface of the hook is smooth and, thus, different from the stems, this approach seems promising and the removal of hook points is important to achieve a completely automatic pipeline with no manual input required from the user. However, the SnapNet allows us to train those and even more classes in one go.
There are two new possible extensions of the approach based on the larger number of classes that can be distinguished with the SnapNet: the semantic labeling of stem skeleton scans could be used to develop a method for the automatic reconstruction of stem skeletons (Hempel, 2019) that in turn could provide the basis for an automatic derivation of constraints, e.g., by finding regularly appearing structures. Another possibility is the extension to grape bunches in earlier development stages, e.g., to BBCH 73 with groat-sized berries, where most of the stem skeleton is visible. Here, a semantic labeling of the parts of the stem provides a robust starting point for the derivation of an initial hypothesis (Peters, 2018b).
The division of the combined twig class into one containing only twigs and another containing subtwigs and pedicles yields results of lower quality compared to collecting all twig, subtwig, and pedicle points into one class, as the twigs are often occluded in views and are, thus, not as thoroughly trained as the other classes. Also, the comparatively low number of twig points makes their recognition harder. While Barmscheidt (2018) showed that an equal distribution

of the generated views around the grape bunch does not boost the results, a larger number of training data might yield some improvement.

Berry detection: The evaluation of the berry detection shows very good results for both number and average diameter of berries, even though the berry numbers are constantly underestimated. This is due to berries being occluded and thus not detectable in the data. Still, the shape of the pointcloud is visually close to that made up by the detected berries. While it would be possible to define a jump that adds inner berries, it cannot be evaluated, as the respective berries are not visible in the data.
It is possible to derive a statistical correction term for the number of berries. However, experiments showed that this does not change the results for the QTL mapping, as the correlation is already very high with more than 0.9.

Grape bunch reconstruction: While the average pedicle lengths are clearly improved with the optimization, the stem lengths are not changed, as no jump is defined to do so. While creating such a jump would be possible, the initial hypothesis already uses all visual clues to fit the rachis into the data as good as possible. Fine changes to, e.g., the curvature, can not be evaluated with the current set of constraints or the likelihood. While the achieved Pearson correlation coefficient is still robust with 0.76, the absolute error can get rather large on few grape bunches with up to 4 cm. However, the process of generating reference data shows problems in this regard too: the reduction from 3D to 2D leads to a distortion of angles. Thus, the curvature of the stem can lead to measurement errors of up to 4 cm. Further sources of errors are the hidden lower end of the rachis, that can only be derived from the positions of berries in this part, and lower berries not being detected because of a suboptimal scanning process leading to holes (see, e.g., the Riesling pointcloud in Figure 5.8g). As the experts in this application field ask for Pearson correlation coefficients of at least 0.7, all measurements delivered by our approach can be considered sufficiently good.
While we estimate the rachis from several points at different heights, as described in Section 4.3.4, leading to a curvature where necessary, the twigs, subtwigs, and pedicles are all represented as a single cylinder. As subtwigs and pedicles are relatively short, this approximation is realistic. The longer twigs in the upper part of the grape bunches on the other hand often show some curvature, as the weight of the berries pulls them downwards. It would be easy to represent such a curvature using, e.g., B-Splines, but there is no clear visual clue in the pointclouds how exactly a twig lies underneath the berries. This makes the modelling approach complicated, as it has to take into account the weight of the berries, their distribution, and the pressure of the berries against each other as well. As we do not currently evaluate the twig lengths and they are so far not considered an important trait in the phenotyping of grape bunches, we currently choose not to include such a model. The missing curvature does have an effect on the lengths of pedicles attached to the twig, but we

assume that it is usually balanced out, as pedicles on one side are longer than they should be, but pedicles on the other side shorter. This leads to a larger standard deviation for pedicle lengths, but the average can still be estimated reliably, as shown in Figure 5.17a.
While a one-to-one evaluation of the optimized stem skeleton to a ground truth would be interesting, such an approach is not possible in the field of grape bunch reconstruction. The reason for this is that with the removal of berries, the stem skeleton is often damaged. The weight of the berries is missing and, additionally, stem skeletons dry out fast, leading to a change in the curvature of twigs, reduced lengths, and a higher fragility, making further damage likely. Another problem is visible in the stem skeleton images in Figures 5.14d, 5.14h, 5.15d, and 5.15h. When the berries are removed, some residue of the berries is left at the end of the pedicles that cannot efficiently be removed. While in images it can be recognized and ignored for the measurements, in 3D scans this is not possible. As the amount of residue varies for each berry, it is also not possible to use a correction term. Thus, the lengths of pedicles derived from stem skeleton scans would be distorted, highly reducing their usefulness.
We used the same set of parameters, introduced in Section 5.4.1, for all grape bunches and achieved stable results. This makes the system especially user-friendly, as the parameters are either set automatically or chosen such that they work on a wide range of examples.

Partitioning into phases: The modification of the RJMCMC algorithm using locally optimized topological jumps leads to an increase of the Pearson correlation coefficient for the average pedicle lengths from 0.65 to 0.71. The reason for this is that the local optimization of topological jumps during the first phase raises their acceptance rate. This effect can be seen for all grape cultivars, but especially for Riesling grape bunches, as those grape bunches show a rather complex stem skeleton with a large number of subtwigs. These subtwigs have to be added using topological jumps, thus, a higher acceptance rate leads to better results on such grape bunches.

Application to other fields: An application to other fields is possible, as long as the underlying objects follow rules that can be formulated as constraints. One close example is the reconstruction of foliated trees. Their growth structure shares similarities with grape bunches, with obvious differences in the scale and growth direction. The challenge here would be the derivation of suitable endpoints. For grape bunches, the berries can be derived and for the initial hypothesis we use the knowledge that each berry has to be attached to a twig by a pedicle. Transferring this idea to foliated trees would require the detection and reconstruction of leaves, which is more challenging because of their often irregular shapes. An alternative would be to start with a minimal hypothesis, e.g., the usually visible tree stump, and derive the rest of the tree during the optimization using jumps. However, this would significantly lengthen the optimization process.

Other examples that allow a more similar approach are the application to apple trees, as apples are attached to twigs similar to grapevine berries, or tomato plants, with tomato panicles being comparable to grape bunches.

Efficiency: To compare the efficiency of our method to the one currently used at the Institute for Grapevine Breeding Geilweilerhof in Siebeldingen, we consider the steps that require manual labor: both approaches require that the grape bunch is harvested and brought inside a lab environment. For our method, the grape bunch needs to be scanned from all sides. This process takes about one minute on average. For the currently used manual method, as described in Section 2.4, all berries have to be removed from the grape bunch. Then they are distributed on a perforated plate, such that each of the berries lies in one of the holes, making the automatic derivation of the berry number and diameter from an image possible. For the stem measurements, an image is taken from the stem skeleton and the required lengths are marked manually in this image. The whole process takes a skilled user on average 12 minutes. Thus, our approach reduces the active working time by a factor of 12.

Chapter 6

Application of the grape bunch reconstruction in the field

Scanning grape bunches in the field offers several advantages: on the one hand, this makes the approach non-invasive, enabling the scanning of the same grape bunch at several points during the development. On the other hand, the grape bunch no longer has to be harvested and taken inside, further reducing the required active working time. Additionally, there is no longer the risk of the grape bunch getting damaged on the way inside.

The Artec 3D Spider Scanner (Artec 3D, 2019a) is highly suitable for outdoor scanning as it uses structured light that is not distorted by strong ambient light. It is also relatively small with measurements of $190 \times 140 \times 130\,\mathrm{mm}$ and light-weight with 0.8 kg, making it easily transportable. Using a battery makes the sensor independent from external power sources.

Tests in practice showed one additional challenge compared to the scanning in a lab environment: in the lab, the grape bunch is rotated by 360°, allowing the generation of full scans. While it would be possible in theory to move the scanner around the grape bunch in the field as well, there is usually not enough space between grape bunch and canopy to actually do this. In most cases, only scans from the front are possible.

Together with our project partners at the Institute for Grapevine Breeding Geilweilerhof in Siebeldingen it was decided to currently require the user to attach a cardboard behind the grape bunch during the scanning process, so that the background can be easily extracted from the scan. This process can be seen in Figure 6.1.

Figure 6.1: *In the field, the grape bunch is scanned from the front, with a cardboard attached behind it as artificial, easy-to-remove background.*

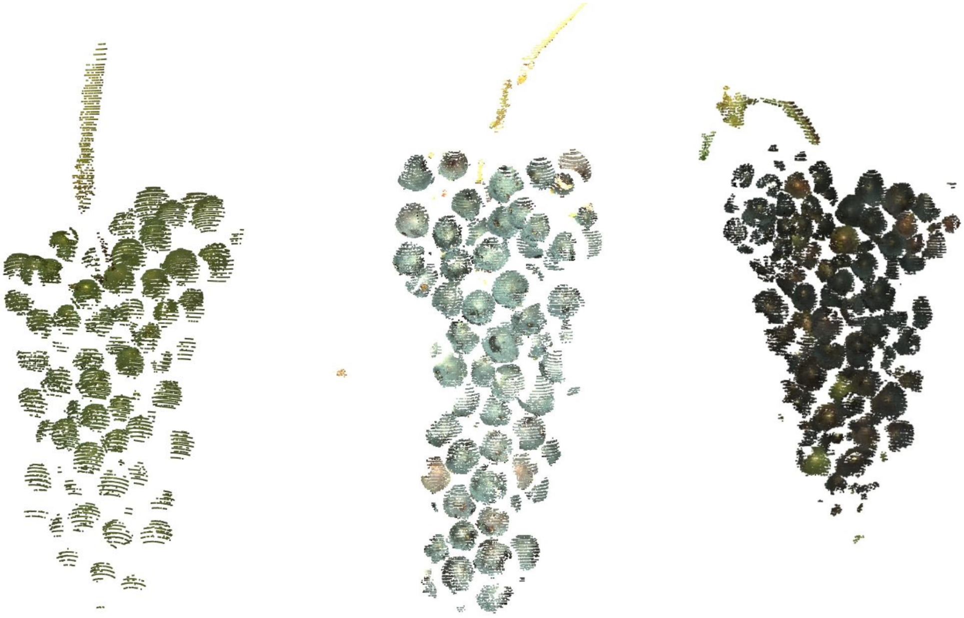
6.1 Data material

Since the scans have to be taken during the harvest time when the staff is highly occupied with routine work, field scans were done for a subset of the *cultivar set* introduced in Section 2.2. This set includes 2 Calardis Blanc, 7 Dornfelder, and 14 Pinot Noir grape bunches, i.e., 23 scans total. Images taken from grape bunches in the field before scanning and screenshots of scans achieved by scanning from the front are shown in Figure 6.2.
The Artec Spider scanner achieves optimal results only if the distance to the object is in a range between 20 and 30 cm (Artec 3D, 2019a). For asymmetric grape bunches, possibly with shoulders or side bunches, this makes the scanning process difficult. Most Dornfelder grape bunches are almost cylindrical, with the grape bunch in Figure 6.2b being slightly more conical than the others. Due to this property, the scanning of this cultivar is comparably easy. Calardis Blanc and Pinot Noir both show irregularities concerning shoulders, side bunches or a distinctly conical shape. In their case, the lower part of the grape bunch is frequently hardly or not at all visible in the pointcloud, as can also be seen in Figures 6.2d and 6.2f. We compare the results to the reference data derived for the complete grape bunch, as the goal is to extrapolate this data from the part of the grape bunch visible in a front scan.

6.2 Reconstruction of front scans

For the reconstruction of grape bunches from front scans we extend an approach we explored first in Peters (2018a). In the following, we describe only the changes to the method presented in Section 4.3.

(a) *Calardis Blanc* ***(b)*** *Dornfelder* ***(c)*** *Pinot Noir*



(d) *Calardis Blanc* ***(e)*** *Dornfelder* ***(f)*** *Pinot Noir*

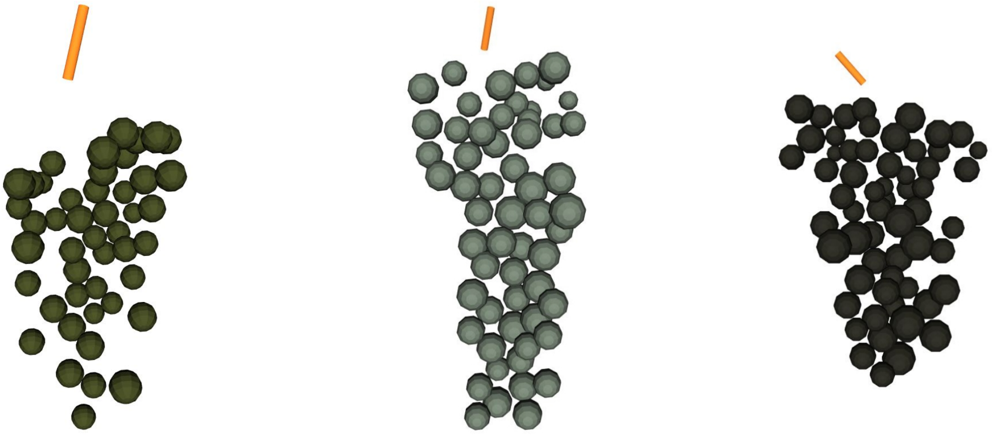
Figure 6.2: *Pointclouds resulting from scanning in the field.*

We refrain from semantic labeling, as almost all points in the pointclouds are part of the surfaces of berries. Therefore, the detection of berries is applied to the whole pointcloud, following the same RANSAC-based procedure introduced in Section 4.3.4. There are several

scans that do not show the peduncle. As we need the peduncle as starting point of the stem skeleton, we approximate it in these cases as the center of mass of the berries in the upper layer. Otherwise, the points covered by berries are extracted from the pointcloud and the peduncle inserted in the remaining part. If the peduncle is visible, but also the stem attaching it to the canopy, the points belonging to the peduncle currently have to be marked manually.
At the border of the scan, only half or an even smaller parts of berries are visible. To detect these berries, berry hypotheses with smaller consensus sets have to be taken into account. Thus, we reduce the threshold on the size of the consensus set to count a detected sphere as valid berry hypothesis $t_i = 50$.
As only a part of the grape bunch is contained in the scan, the number of twigs that is inserted in the initial hypothesis has to be reduced. For this, we adjust the parameter k from the k-means clustering of berry centers to create potential twig endpoints: While equation 5.1 stays the same, we restrict k_{min} and k_{max} to a smaller range: $k_{min} = 5 \leq k \leq k_{max} = 12$.
The rachis is estimated by connecting the end of the peduncle directly with the lowest potential twig endpoint. As the berries are only scanned from the front and, thus, do not give a clear hint about the shape of the grape bunch, the approximation of rachis points as center points of the berries at different levels is removed.
The RJMCMC optimization is only slightly adjusted: due to the shape of the grape bunches, there is a tendency to insert twigs that point upwards, along the side of the grape bunch. To restrict this, we weight the constraints. The angle constraint $G_1^{(1)}$ (Equation 4.7) that restricts the angles between rachis and twigs to about 90° is given the weight $w_h = \frac{3}{12}$, while the other 9 constraints are assigned a weight of $w_e = \frac{1}{12}$.
The parameter σ that steers the strictness of the length constraint on pedicles $G_4^{(3)}$ is set empirically slightly larger to $\sigma = 5\,\text{mm}$.
Otherwise, the method is applied as described before.

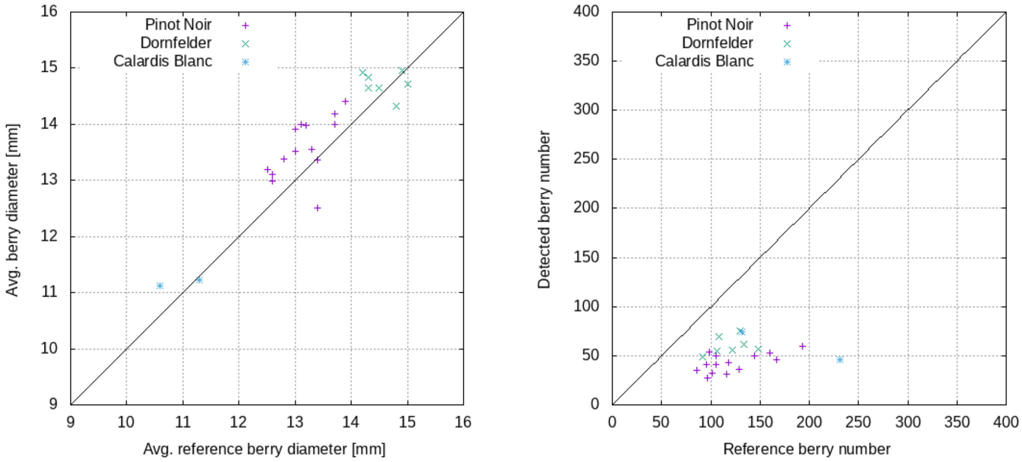
6.3 Evaluation

We evaluate the method analogously to Chapter 5.
Figure 6.3 shows the results of the application of the berry detection approach to the pointclouds in Figure 6.2. The parts that are visible in the pointcloud are well reconstructed, but the effect that berries are missing due to an incomplete scanning process is more significant than for the lab scans. However, the number of berries could not be expected to be stable anyway, as we scan only a part of the grape bunch. For the shown Calardis Blanc grape bunch, only 74 of 132 berries are detected, for Dornfelder 57 of 148 and for Pinot Noir 60 of 193. All results highly underestimate the number of berries.
The average diameter of the berries on the other hand can be derived with sufficient robustness even from the lower number of berries: for Calardis Blanc, the mean percentage error is 0.62 %, for Dornfelder 2.38 % and for Pinot Noir 4.05 %.
The results from the comparison with statistical reference data for the berry detection are



*(**a**) Calardis Blanc* *(**b**) Dornfelder* *(**c**) Pinot Noir*

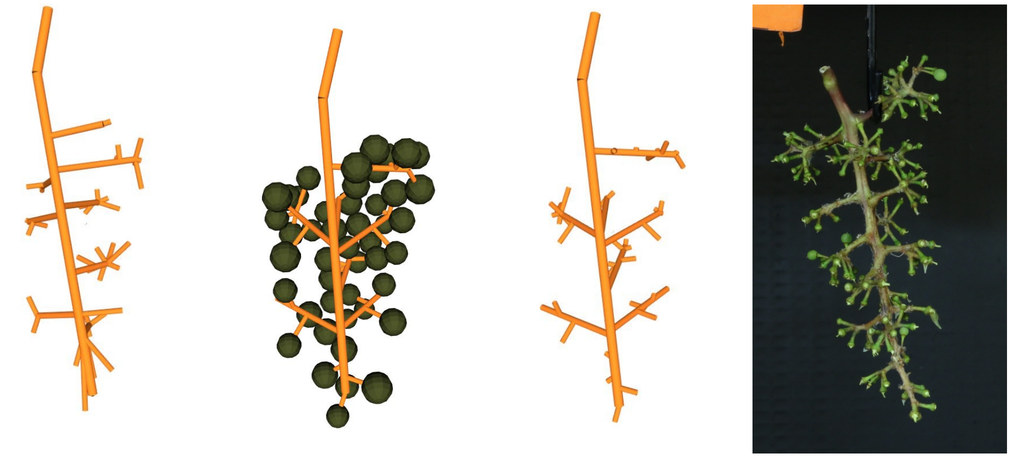
Figure 6.3: *Exemplary results achieved on field scans of grape bunches for the berry detection.*



*(**a**) Average berry diameters* *(**b**) Number of berries*

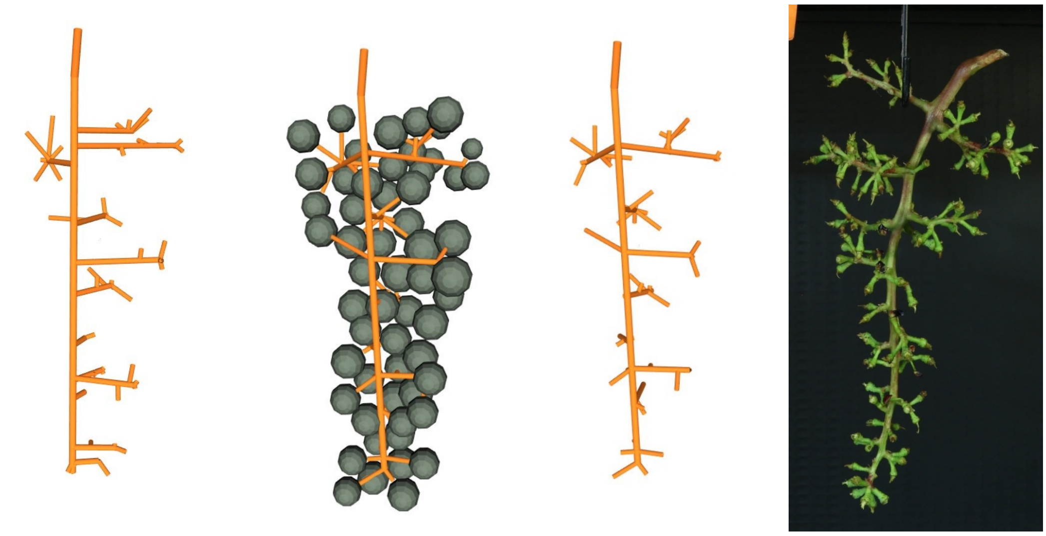
Figure 6.4: *Results achieved on the field scans for the berry detection, covering the average berry diameters (Pearson correlation coefficient 0.91) and the number of berries (Pearson correlation coefficient 0.25). As less than half of the berries are contained in the pointclouds, it is not possible to detect all of them.*

shown in Figure 6.4. Similar to the examples in Figure 6.3 the berry diameters remain stable at almost the same quality with a Pearson correlation coefficient of 0.91. For the berry numbers, the lower scan quality and lower visibility of berries in the data overall lead to a Pearson correlation coefficient of only 0.25. As the percentage of the grape bunch that is scanned is not the same for all grape bunches, no clear correlation can be found.



(a) *Calardis Blanc initial hypothesis* *(b)* *Final reconstruction* *(c)* *Final reconstruction without berries* *(d)* *Stem skeleton*

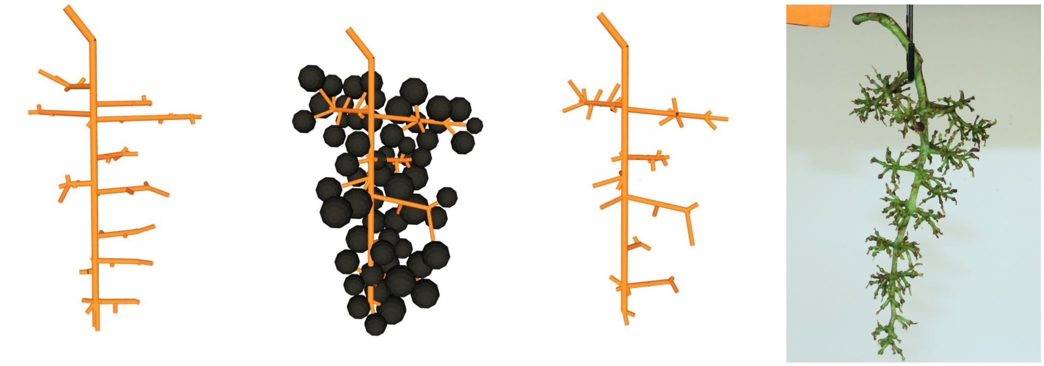
Figure 6.5: *Exemplary results achieved on a field scan of a Calardis Blanc grape bunch for the grape bunch reconstruction.*



(a) *Dornfelder initial hypothesis* *(b)* *Final reconstruction* *(c)* *Final reconstruction without berries* *(d)* *Stem skeleton*

Figure 6.6: *Exemplary results achieved on a field scan of a Dornfelder grape bunch for the grape bunch reconstruction.*

Exemplary results for the grape bunch reconstruction of field scans can be seen in Figures 6.5, 6.6, and 6.7. Again, there is no change between initial and final hypotheses regarding the length of the stem, as there is no jump that adjusts it. The stem length estimated for the



(a) *Pinot Noir initial hypothesis* ***(b)*** *Final reconstruction* ***(c)*** *Final reconstruction without berries* ***(d)*** *Stem skeleton*

Figure 6.7: *Exemplary results achieved on a field scan of a Pinot Noir grape bunch for the grape bunch reconstruction.*

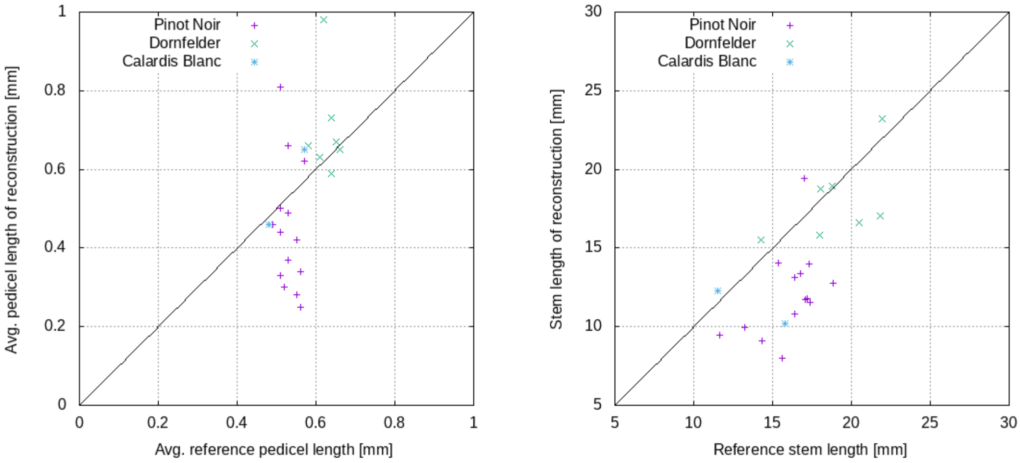
Dornfelder grape bunch shows a mean percentage error of 18.27 % or an underestimation of 3.74 cm. This can be traced back to the heuristics used to insert the rachis that connects the lower part of the peduncle with the lowest candidate for a twig endpoint. As those twig endpoints are derived by clustering berry center points and the berries are only scanned from the front the rachis endpoint is in this case set too far up and too close to the berries, leading to an underestimation of the stem length.

For the Pinot Noir grape bunch shown in the figures, it is clearly visible that the lower part of the grape bunch is missing. This is due to the fact that it cannot be seen in the pointcloud (Figure 6.2f). This has a strong effect on the mean percentage error of the stem length, going up to 32.30 % or an absolute error of 6.08 cm.

While the Calardis Blanc grape bunch shows similar holes in the lower part of the pointcloud, some berries are still sufficiently well represented to be detected. But here, another source of error can be seen: the peduncle is well visible in the pointcloud, but was cut such that the part left of it in the image is shorter. As we calculate the stem as the sum of the lengths of rachis and peduncle, this means that the calculated stem length is too long. The overestimation of the peduncle and a slight undererstimation of the rachis balance each other out on this grape bunch, leading all in all to a mean percantage error of 6.49 %, or an absolute error of 0.75 cm.

The average pedicle length shows the same tendency as for lab scans. The initial hypotheses underestimate it with mean percentage errors of 13.85 % for Calardis Blanc, 13.48 % for Dornfelder, and even 30.43 % for Pinot Noir due to showing too many twigs. The optimization improves those results to 7.52 % for Calardis Blanc, 1.91 % for Dornfelder, and 8.90 % for Pinot Noir.

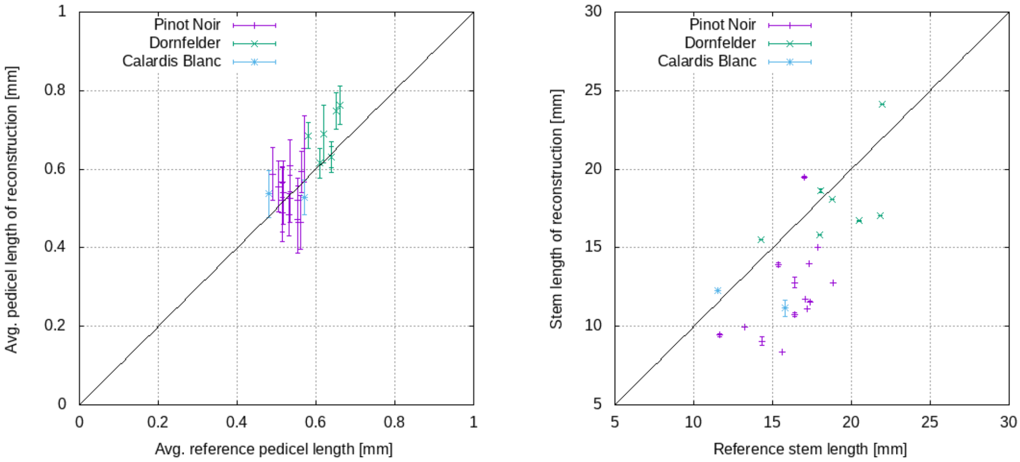
As explained in Section 6.2, we assign a larger weight to the angle constraint between rachis and twigs. As can be seen in Figure 6.5c the Calardis Blanc grape bunch still shows this



(**a**) *Average pedicle lengths*

(**b**) *Stem lengths*

Figure 6.8: *Results achieved on the field scans for average pedicle length (Pearson correlation coefficient 0.47) and stem length (Pearson correlation coefficient 0.67) of the initial hypothesis.*



(**a**) *Average pedicle lengths*

(**b**) *Stem lengths*

Figure 6.9: *Results achieved on field scans for average pedicle length (Pearson correlation coefficient 0.71) and stem length (Pearson correlation coefficient 0.67) of the final hypothesis. Due to the difficult scanning process, the lower part of the grape bunch is often missing for Calardis Blanc and Pinot Noir. Therefore, the rachis cannot be completely reconstructed and the Pearson correlation coefficient is comparably low.*

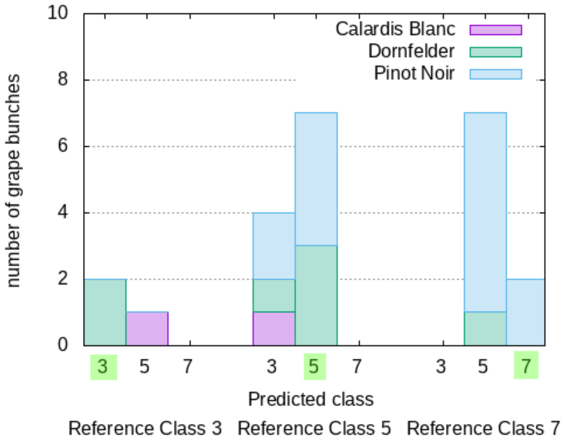
effect. This is due to the shape of the front scan that makes it most efficient to cover berries using such upwards pointing-twigs. While the larger weight on the respective constraint makes this effect more unlikely, it is not completely prevented.

Figure 6.8 shows the results for the average length of pedicles and the length of the stem for the initial hypothesis and Figure 6.9 for the final reconstruction. The results for the average pedicle length are slightly worse than for lab scans (Figure 5.16a) with a Pearson correlation coefficient of 0.47 instead of 0.59. While the results for Calardis Blanc and Dornfelder are still rather close to the reference values, for Pinot Noir, the lengths are underestimated. This is again connected to the initial number of twigs that is too large for Dornfelder grape bunches. The optimization corrects this and achieves a Pearson correlation coefficient of 0.71.

For the stem length, again there are only few changes due to the optimization, as there are no explicit jumps to adjust it. Both, initial hypothesis and final reconstruction yield a Pearson correlation coefficient of 0.67. This is lower than for the lab scans and the plots show that the length is mostly underestimated. This is due to the fact that the field scans often do not show the complete lower part of the grape bunch, thus, inserting the rachis between the peduncle and the lowest twig endpoint candidate does not capture the full length of the grape bunch.

The compactness is computed similar as described in Section 5.4.3. Equation (5.2) is used to derive a first estimation value. The division into classes is adjusted to mirror the new relation between volumes: compactness values between 0 and 2.0 are assigned to class 1, 2.0 to 4.5 to class 3, 4.5 to 6.1 to class 5, 6.1 to 12.0 to class 7, and everything above this value to class 9. As can be seen, the distribution of classes is spread further for front scans, as the convex hull is computed only for the front scan, making the difference between it and the summed up volume of berries and stems smaller. This leads to larger compactness values.

As can be seen in Figure 6.10, again there is no deviation of more than one class.



(a) Final compactness

Figure 6.10: *Results achieved on the field scans for the compactness. There is no difference between initial and final hypotheses. As the compactness classes 1 and 9 are only covered by the variety set that is not used in this experiment, and those classes are also never predicted, we removed them from the plot for clarity.*

6.4 Discussion

For average berry diameters and average pedicle lengths our method achieves results that are comparable to those on lab scans (Pearson correlation coefficients 0.91 to 0.92 and 0.71 to 0.71). Due to their cylindrical shape, the derivation of the stem lengths for Dornfelder grape bunches is also relatively stable, with an average mean percentage error of 11 %.

The derivation of berry numbers and stem lengths are strongly connected to the scanning process. If the asymmetry of the grape bunches makes it difficult to cover the complete grape bunch from top to bottom, the pointcloud is missing the necessary characteristics to provide a robust estimate for the stem length. Additionally, the rachis endpoint is derived from the berries in the lower part of the grape bunch, meaning if too many of them are missing, this location is no longer chosen correctly.

The number of berries cannot be derived correctly if the whole grape bunch is not visible. But it might be possible to achieve a better correlation if the scanned proportion of the grape bunch is kept approximately equal.

Better estimations of the stem lengths for asymmetric grape bunches and the derivation of a method to extrapolate the number of berries both require an easier scanning process. This could be approached by using another scanner with larger working distance, e.g., the Artec Leo (Artec 3D, 2019b). This 3D scanner works with structured light as well, but optimal results can be achieved if the distance to the object is between 0.35 and 1.2 m, clearly improving the working range between 0.2 and 0.3 m of the Artec Spider Scanner. Additionally, scanning with the Artec Leo Scanner does not require a connected laptop, making the process more flexible.

Chapter 7

Conclusions

In viticulture, a loose grape bunch architecture is one of the most import physical barriers that make the establishment of fungal diseases and pests less likely. The developed phenotyping pipeline delivers complete component-based 3D models of the observed grape bunches, deriving their architecture objectively, accurately, and with high-throughput. Due to easy-to-handle sensor technology and an intuitive graphical user interface the pipeline is open to all kinds of users. No interaction is required, but we allow an easy adjustment of parameters if necessary.

The employment of a flexible, constraint-based modelling and reliable semantic labeling of the 3D pointclouds makes it possible to use the method across different cultivars and genotypes. Efficiency and accuracy are improved using a two-phased version of the RJMCMC optimization.

The phenotyping pipeline has been evaluated on scans taken in a lab environment and in the field yielding Pearson correlation coefficients up to 0.9 and an efficiency improvement by the factor of 12 compared to the manual phenotyping.

All in all, we meet the introduced challenges in the following ways:

- We use a model-based approach to deal with the lack of training data.
- As ripe grape bunches are subject to massive self-occlusion, the inner stem skeleton cannot be inferred from the data. Thus, we use an RJMCMC optimization for model selection and parameter estimation. We propose an adjusted version that divides the optimization into phases, with one phase concentrating on the selection of the model and the second on the optimization of the parameters. Our results show that this enhances the reconstruction quality by increasing the acceptance rate for jumps that change the model dimension by up to 75 %.

- To achieve an approach suitable for cross-cultivar application, we define constraints based on geometrical and topological relations between grape bunch organs, resulting in Pearson correlation coefficients between 0.7 and 0.9 on scans taken in a lab environment.
- To robustly reconstruct the visible parts of the grape bunch, a semantic labeling is applied to assign the points of the dense high-precision pointcloud to the surfaces of their respective plant organs. We define the new HFPFH descriptor as a combination of the hue channel of the HSI color space, and the Fast-Point-Feature histograms, resulting in 99 % accuracy with supervised classification and 88 % with unsupervised classification, outperforming the established descriptors PFHRGB and CSHOT. Additionally, we evaluate the SnapNet on pointclouds of grape bunches and of grape bunch stem skeletons, yielding at least 93 % accuracy for each class for the more complicated semantic labeling into surface points of the main stem, twigs, berries, and hook.

It is very important to enable scanning in the field, as this makes the method non-invasive and applicable to the same grape bunch at different points in time during its development, making comparisons possible. Also, the active working time is highly reduced if grape bunches do not have to be harvested and brought inside a lab environment. In this context, we achieve excellent results on symmetrical grape bunches. Grape bunches with strongly asymmetric shapes require the introduction of best practices for scanning.

7.1 Future work

The graphical user interface introduced in this thesis can be improved by giving the user more options to interact with the data. One example would be to introduce a selection tool, allowing the user to select one or several parts of the grape bunch to automatically extract the properties, like lengths, diameters, or angles between the parts.

So far, we refrained from modelling the exact twig curvatures, as they depend on the weight of berries, their distribution on the twig, and the pressure of the berries against each other. Still, this does lead to deviating lengths of the upper twigs in the grape bunch and to a larger standard deviation for the pedicles attached to those twigs. To provide an exact reconstruction of the whole grape bunch, it would be necessary to tackle this problem.

As explained in Section 5.5 the extension to other application fields is possible, as long as the rules imposed on the structure of the target objects can be formulated as constraints. This holds, e.g., for foliated trees, apple trees, or tomato plants.

The successful automated reconstruction of grape bunches, including the stem skeleton, from 3D scans offers new possibilities for larger projects, that are introduced in the following.

Aggregation of physical barriers: Beside the bunch compactness, there are two other physical barriers that protect grape bunches from fungal infections. One is the structure of the canopy, and the other the properties of the berry skin in regards to thickness, the hydrophobic characteristics of the berry cuticle, and its epicuticular wax layer. Especially for the latter, there are approaches to derive its thickness manually (Herzog et al., 2015) or from illumination-separated images using deep learning (Barré et al., 2019). Combining these approaches with the pipeline presented in this thesis results in a 3D reconstruction with attributed berries, including even more information about the grape bunch.

Growth analysis: Another interesting area of application is the reconstruction of grape bunches at earlier development stages. We did first experiments regarding the application of the full reconstruction approach to grape bunches of development stage BBCH 73, with groat-sized berries, in (Peters, 2018b). In this context, larger weight has to be given to the creation of the initial hypothesis. While pedicles are usually still occluded by berries, the rachis and twigs are mostly visible. Thus, they can be reconstructed in more detail. It has to be evaluated whether an optimization is even necessary, or if it is sufficient to derive the structure of the grape bunch from the data.
The application of both the presented approach and a version adjusted for working on earlier development stages in the field gives rise to the possibility to scan the same grape bunch at different points in time during its development and analyse the resulting reconstructions in regards to their differences. Further knowledge about the weather during the growing process or the properties of the soil can be used to evaluate the cause of changes, especially in regards to which abiotic factors benefit a loose grape bunch architecture.
Rose et al. (2016) introduced a method to extract the berries from grape bunches in pointclouds generated using multi-view stereo vision with the PHENObot. As their method takes pictures only from the side, the resulting extracted pointclouds are comparable to the front scans described in Chapter 6. The phenotyping pipeline would, thus, be applicable to those pointclouds as well. Using the PHENObot, it would be comparably easy to cover large parts of the grapevine field even several times during the growth period.

Constraint learning: So far, expert knowledge is required for the definition of the constraints. On the one hand, this offers the possibility to the users to directly influence the approach and take control in its application. On the other hand, it would be interesting to introduce an automatic derivation of constraints, as mentioned in Section 5.5. One possibility for this is the automatic extraction and analysis of frequently occurring substructures in the grape bunch reconstructions. Constraints controlling the occurrence of such substructures can be defined automatically. If applied to changing populations or a growing number of genotypes, self-learning could be used by adding additional substructures as constraints whenever their occurrence exceeds a given threshold, or removing old ones if they no longer appear.

Appendix A

Constraint relations

During the introduction of the constraints in Section 4.3.2 we use several relations that are computed on the grape organs. Those will be defined formally in the following.

Similarity constraint (Equation (4.8)**):** $s(x)$ is a relation that returns the startpoint of the cylinder representing a grape organ x, with the startpoint being the point that is closer to the peduncle following the topology. E.g., for all pedicles the startpoint is the point attached to the twig and the endpoint the one attached to a berry. Additionally, $o(x)$ describes a relations that computes the orientation of the cylinder corresponding to grape organ x in the topological direction towards the berries.

$$\text{count\_similar\_elements}(x_i, x_j) = \begin{cases} 1 & \text{if } \|s(x_i) - s(x_j)\| < 10\,\text{mm and angle}(o(x_i), o(x_j)) < 30^\circ \\ 0 & \textit{otherwise} \end{cases} \tag{A.1}$$

Overlap constraint (Equation (4.9)**):** The function $dist(x, y)$ computes the smallest Euclidean distance between two grape organs x and y, each interpreted as line between their respective start and endpoints. The relation $r(x)$ retrieves the radius of a grape organ x.

$$\text{count\_overlapping\_parts}(x_i, x_j) = \begin{cases} 1 & \text{if } dist(x_i, x_j) < r(x_i) + r(x_j) \\ 0 & \textit{otherwise} \end{cases} \tag{A.2}$$

Decreasing length constraint (Equation (4.14)**):**

$$\text{length\_shorter\_elements}(x_i, x_j) = \begin{cases} \|x_j\| - \|x_i\| & \text{if } x_i \text{ lies above } x_j \text{ and } \|x_i\| < \|x_j\| \\ 0 & \textit{otherwise} \end{cases} \quad \text{(A.3)}$$

Twig endpoint constraint (Equation (4.15)**):** The relation outside(x_i) returns true if x_i lies outside the convex hull calculated around the berry centers, and false otherwise.

$$\text{count\_if\_outside\_convex\_hull}(x_i) = \begin{cases} 1 & \text{if outside}(x_i) = \text{true} \\ 0 & \textit{otherwise} \end{cases} \quad \text{(A.4)}$$

Glossary

3D reconstruction is a three-dimensional estimation of the target object represented with geometric primitives.

Angle constraint restricts the angles between the respective grape bunch organs to a fixed range.

BBCH scale is a scale to measure the development stages of grape bunches. BBCH 87-89 corresponds to grape bunches with fully developed berries, ready for harvesting.

Berry is the fruit of a grape bunch; the berries serve as endpoints for the pedicles during the reconstruction of the stem skeleton.

Complete grape bunch describes the grape bunch including berries and inner stem skeleton.

Connection constraint enforces a connection between the respective grape bunch organs.

Constraints are used to represent geometrical and topological information to restrict the hypotheses space of the RJMCMC optimization.

Decreasing length constraint punishes if lengths of grape bunch organs do not decrease in the specified way.

Deviation constraint restricts the standard deviation of the respective grape bunch organs.

Fast-Point-Feature Histograms are histograms representing shape information for each point based on the geometrical relations between the point and its neighbours.

Geometrical constraints encode information about the dimensions and geometrical relations of the grape bunch organs.

Geometrical jumps are jumps that optimize parameters of the hypothesis.

Hue-Fast-Point-Feature Histograms are histograms combining the shape information included in FPFHs with the hue channel from the HSI color model.

Jumps encode how a hypothesis can be adjusted during the RJMCMC optimization.

k-means is an unsupervised clustering approach that can be used in combination with a descriptor.

Length constraint restricts the lengths of the grape bunch organs.

Overlap constraint punishes overlapping grape bunch organs.

Pedicle is a stalk connecting a berry to a twig or subtwig.

Peduncle is the former connection between the rachis of the grape bunch and the canopy; serves as startpoint for the reconstruction of the stem skeleton.

Potential connection constraint allows a connection between the respective grape bunch organs.

Rachis is the main stem of the grape bunch below the peduncle.

Random Sample Consensus is a method that can be used to fit geometric primitives into pointclouds.

Reversible-Jump-Markov-Chain-Monte-Carlo is an optimization algorithm that includes model selection and parameter optimization.

Similarity constraint punishes the occurrence of very similar elements.

SnapNet is a multi-view deep learning approach that is used for the semantic labeling of pointclouds.

Stem skeleton is the collection of stems and stalks inside the grape bunch, including peduncle, rachis, twigs, subtwigs, and pedicles.

Subtwig is a twig of second order, i.e., a stalk connecting a twig with pedicles.

Support Vector Machine is a supervised learning approach that can be used in combination with a descriptor.

Topological constraints encode information about the topology of the grape bunch stem skeleton.

Topological jumps are jumps that change the topology of the hypothesis, thus, changing the model dimension.

Twig connects the rachis to the pedicles and/or subtwigs.

Twig endpoint constraint punishes twig endpoints lying outside the convex hull spanned by the berries.

Acronyms

BBCH	Biologische Bundesanstalt, Bundesssortenamt and Chemische Industrie.
CNN	Convolutional Neural Network.
FPFH	Fast-Point-Feature Histogram.
HFPFH	Hue-Fast-Point-Feature Histogram.
RANSAC	Random Sample Consensus.
RJMCMC	Reversible-Jump-Markov-Chain-Monte-Carlo.
SVM	Support Vector Machine.

List of Figures

List of Tables

Bibliography

Fabian Aiteanu and Reinhard Klein (2014). Hybrid Tree Reconstruction From Inhomogeneous Point Clouds. *The Visual Computer*, 30(6):763–771, DOI: `10.1007/s00371-014-0977-7`.

Arturo Aquino, Ignacio Barrio, Maria-Paz Diago, Borja Millan, and Javier Tardaguila (2018a). vitisBerry: An Android-smartphone application to early evaluate the number of grapevine berries by means of image analysis. *Computers and Electronics in Agriculture*, 148:19 – 28, DOI: `10.1016/j.compag.2018.02.021`.

Arturo Aquino, Maria P. Diago, Borja Millán, and Javier Tardáguila (2017). A new methodology for estimating the grapevine-berry number per cluster using image analysis. *Biosystems Engineering*, 156:80 – 95, DOI: `10.1016/j.biosystemseng.2016.12.011`.

Arturo Aquino, Borja Millan, Maria-Paz Diago, and Javier Tardaguila (2018b). Automated early yield prediction in vineyards from on-the-go image acquisition. *Computers and Electronics in Agriculture*, 144:26 – 36, DOI: `10.1016/j.compag.2017.11.026`.

Arturo Aquino, Borja Millan, Daniel Gaston, María-Paz Diago, and Javier Tardaguila (2015a). vitisFlower®: Development and Testing of a Novel Android-Smartphone Application for Assessing the Number of Grapevine Flowers per Inflorescence Using Artificial Vision Techniques. *Sensors*, 15(9):21204–21218, DOI: `10.3390/s150921204`.

Arturo Aquino, Borja Millan, Salvador Gutiérrez, and Javier Tardáguila (2015b). Grapevine flower estimation by applying artificial vision techniques on images with uncontrolled scene and multi-model analysis. *Computers and Electronics in Agriculture*, 119:92 – 104, DOI: `10.1016/j.compag.2015.10.009`.

Artec 3D (2019a). `https://www.artec3d.com/3d-scanner/artec-spider` [Accessed: 10.07.2019].

Artec 3D (2019b). `https://www.artec3d.com/portable-3d-scanners/artec-leo` [Accessed: 10.07.2019].

David Arthur and Sergei Vassilvitskii (2007). K-means++: The Advantages of Careful Seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, DOI: `10.1109/TPAMI.2016.2644615`.

Fabian Barmscheidt (2018). On multi-view strategies for CNN-based semantic labeling of 3D point clouds. Master's thesis, University of Bonn.

Pierre Barré, Katja Herzog, Rebecca Höfle, Matthias B. Hullin, Reinhard Töpfer, and Volker Steinhage (2019). Automated phenotyping of epicuticular waxes of grapevine berries using light separation and convolutional neural networks. *Computers and Electronics in Agriculture*, 156:263 – 274, DOI: `10.1016/j.compag.2018.11.012`.

Riaz Bhunnoo (2018). Feeding the world: a global challenge. *New Food*, 1, `https://www.newfoodmagazine.com/article/64808/food-security-riaz-bhunnoo/`.

Alexandre Boulch, Joris Guerry, Bertrand Le Saux, and Nicolas Audebert (2018). SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Computers & Graphics*, 71:189 – 198, DOI: `10.1016/j.cag.2017.11.010`.

Kang Chen, Yu-Kun Lai, Yu-Xin Wu, Ralph Martin, and Shi-Min Hu (2014). Automatic Semantic Modeling of Indoor Scenes from Low-quality RGB-D Data Using Contextual Information. *ACM Transaction on Graphics*, 33(6):208:1–208:12, DOI: `10.1145/2661229.2661239`.

Stephen Chenney and D. A. Forsyth (2000). Sampling Plausible Solutions to Multi-body Constraint Problems. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 219–228. DOI: `10.1145/344779.344882`.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger (2016). 3d u-net: Learning dense volumetric segmentation from sparse annotation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 424–432. Springer International Publishing, DOI: `10.1007/978-3-319-46723-8_49`.

S. Cubero, M.P. Diago, J. Blasco, J. Tardaguila, J.M. Prats-Montalbán, J. Ibáñez, J. Tello, and N. Aleixos (2015). A new method for assessment of bunch compactness using automated image analysis. *Australian Journal of Grape and Wine Research*, 21(1):101–109, DOI: `10.1111/ajgw.12118`.

Sergio Cubero, María Paz Diago, José Blasco, Javier Tardáguila, Borja Millán, and Nuria Aleixos (2014). A new method for pedicel/peduncle detection and size assessment of grapevine berries and other fruits by image analysis. *Biosystems Engineering*, 117:62–72, DOI: `10.1016/j.biosystemseng.2013.06.007`.

Luc De Raedt, Tias Guns, and Siegfried Nijssen (2010). Constraint Programming for Data Mining and Machine Learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, pages 1671–1675.

Youness Dehbi, Fabian Hadiji, Gerhard Gröger, Kristian Kersting, and Lutz Plümer (2016). Statistical Relational Learning of Grammar Rules for 3D Building Reconstruction. *Transactions in GIS*, 21(1):134–150, DOI: `10.1111/tgis.12200`.

Maria P Diago, Andres Sanz-Garcia, Borja Millan, Jose Blasco, and Javier Tardaguila (2014). Assessment of flower number per inflorescence in grapevine by image analysis under field conditions. *Journal of the Science of Food and Agriculture*, 94(10):1981–1987, DOI: `10.1002/jsfa.6512`.

Maria P Diago, Javier Tardaguila, Nuria Aleixos, Borja Millan, Jose M Prats-Montalban, Sergio Cubero, and Jose Blasco (2015). Assessment of cluster yield components by image analysis. *Journal of the Science of Food and Agriculture*, 95(6):1274–1282, DOI: `10.1002/jsfa.6819`.

Martin A. Fischler and Robert C. Bolles (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, DOI: `10.1145/358669.358692`.

D. Font, T. Pallejà, M. Tresanchez, M. Teixidó, D. Martinez, J. Moreno, and J. Palacín (2014). Counting red grapes in vineyards by detecting specular spherical reflection peaks in RGB images obtained at night with artificial illumination. *Computers and Electronics in Agriculture*, 108:105–111, DOI: `10.1016/j.compag.2014.07.006`.

Robert T. Furbank and Martin Tester (2011). Phenomics – technologies to relieve the phenotyping bottleneck. *Trends in Plant Science*, 16(2):635–644, DOI: `10.1016/j.tplants.2011.09.005`.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587. DOI: `10.1109/CVPR.2014.81`.

Peter J. Green (2003). *Trans-dimensional Markov chain Monte Carlo*, chapter 6, pages 179 – 198. Oxford University Press, United Kingdom.

Jonatan Grimm, Katja Herzog, Florian Rist, Anna Kicherer, Reinhard Toepfer, and Volker Steinhage (2019). An adaptable approach to automated visual detection of plant organs with applications in grapevine breeding. *Biosystems Engineering*, 183:170–183, DOI: `10.1016/j.biosystemseng.2019.04.018`.

Joris Guerry, Alexandre Boulch, Bertrand Le Saux, Julien Moras, Aurélien Plyer, and David Filliat (2017). SnapNet-R: Consistent 3D Multi-view Semantic Labeling for Robotics. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 669–678. DOI: `10.1109/ICCVW.2017.85`.

Yulan Guo, Ferdous Sohel, Mohammed Bennamoun, Min Lu, and Jianwei Wan (2013). Rotational projection statistics for 3d local surface description and object recognition. *International Journal of Computer Vision*, 105(1):63–86, ISSN: `1573-1405`, DOI: `10.1007/s11263-013-0627-y`.

Anne Hempel (2019). CNN basierte Rekonstruktion eines Weintraubenstielgerüstes aus einer 3D Punktwolke. Bachelor's thesis, University of Bonn.

Katja Herzog, Rolf Wind, and Reinhard Töpfer (2015). Impedance of the Grape Berry Cuticle as a Novel Phenotypic Trait to Estimate Resistance to Botrytis Cinerea. *Sensors*, 15(6):12498–12512, DOI: `10.3390/s150612498`.

Pingbo Hu, Bisheng Yang, Zhen Dong, Pengfei Yuan, Ronggang Huang, Hongchao Fan, and Xuan Sun (2018). Towards Reconstructing 3D Buildings from ALS Data Based on Gestalt Laws. *Remote Sensing*, 10(7), DOI: `10.3390/rs10071127`.

Hai Huang, Claus Brenner, and Monika Sester (2013). A generative statistical approach to automatic 3d building roof reconstruction from laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 79:29 – 43, DOI: `10.1016/j.isprsjprs.2013.02.004`.

E. Ivorra, A.J. Sánchez, J.G. Camarasa, M.P. Diago, and J. Tardaguila (2015). Assessment of grape cluster yield components based on 3D descriptors using stereo vision. *Food Control*, 50:273–282, DOI: `10.1016/j.foodcont.2014.09.004`.

Bernhard Japes, Jennifer Mack, Florian Rist, Katja Herzog, Reinhard Töpfer, and Volker Steinhage (2018). Multi-View Semantic Labeling of 3D Point Clouds for Automated Plant Phenotyping. *arXiv preprint*, `http://arxiv.org/abs/1805.03994`.

Thosten Joachims (1999). Making large-Scale SVM Learning Practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 11, pages 169–184. MIT Press, Cambridge, MA, DOI: `10.17877/DE290R-5098`.

Andrew E. Johnson and Martial Hebert (1999). Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, DOI: `10.1109/34.765655`.

Anna Kicherer, Katja Herzog, Michael Pflanz, Markus Wieland, Philipp Rüger, Steffen Kecke, Heiner Kuhlmann, and Reinhard Töpfer (2015). An Automated Field Phenotyping Pipeline for Application in Grapevine Research. *Sensors*, 15(3):4823–4836, DOI: `10.3390/s150304823`.

Anna Kicherer, Ribana Roscher, Katja Herzog, Silvio Šimon, Wolfgang Förstner, and Reinhard Toepfer (2013). BAT (Berry Analysis Tool): A high-throughput image interpretation tool to acquire the number, diameter, and volume of grapevine berries. *Vitis Grapevine J. Res.*, 52:129–135.

Young Min Kim, Niloy J. Mitra, Dong-Ming Yan, and Leonidas Guibas (2012). Acquiring 3D Indoor Environments with Variability and Repetition. *ACM Transactions on Graphics*, 31(6):138:1–138:11, DOI: `10.1145/2366145.2366157`.

Maria Klodt, Katja Herzog, Reinhard Töpfer, and Daniel Cremers (2015). Field phenotyping of grapevine growth using dense stereo reconstruction. *BMC Bioinformatics*, 16(1):1–11, DOI: `10.1186/s12859-015-0560-x`.

Qingde Li and J. G. Griffiths (2004). Least squares ellipsoid specific fitting. In *Proceedings on Geometric Modeling and Processing*, pages 335–340. DOI: `10.1109/GMAP.2004.1290055`.

Scarlett Liu, Steve Cossell, Julie Tang, Gregory Dunn, and Mark Whitty (2017). A computer vision system for early stage grape yield estimation based on shoot detection. *Computers and Electronics in Agriculture*, 137:88 – 101, DOI: `10.1016/j.compag.2017.03.013`.

Scarlett Liu, Xuesong Li, Hongkun Wu, Bolai Xin, Julie Tang, Paul R. Petrie, and Mark Whitty (2018). A robust automated flower estimation system for grape vines. *Biosystems Engineering*, 172:110 – 123, DOI: `10.1016/j.biosystemseng.2018.05.009`.

Scarlett Liu, Samuel Marden, and Mark Whitty (2013). Towards automated yield estimation in viticulture. *Proceedings of the Australasian Conference on Robotics and Automation, Sydney, Australia*, 24.

Scarlett Liu, Mark Whitty, and Steve Cossell (2015). A Lightweight Method for Grape Berry Counting based on Automated 3D Bunch Reconstruction from a Single Image. In *International Conference on Robotics and Automation (IEEE), Workshop on Robotics in Agriculture*.

Yotam Livny, Feilong Yan, Matt Olson, Baoquan Chen, Hao Zhang, and Jihad El-Sana (2010). Automatic Reconstruction of Tree Skeletal Structures from Point Clouds. *ACM Transactions on Graphics*, 29(6), DOI: `10.1145/1882261.1866177`.

David H. Lorenz, Klaus Werner Eichhorn, Hermann Bleiholder, Rudolf Klose, Rudolf Meier, and Ed Weber (1994). Phänologische Entwicklungsstadien der Weinrebe (Vitis vinifera L. ssp. vinifera). Codierung und Beschreibung nach der erweiterten BBCH-Skala. *Wein-Wissenschaft, Vitic. Enol. Sci*, 49:66–70.

Jennifer Mack, Christian Lenz, Johannes Teutrine, and Volker Steinhage (2017a). High-precision 3D detection and reconstruction of grapes from laser range data for efficient phenotyping based on supervised learning. *Computers and Electronics in Agriculture*, 135:300–311, DOI: `10.1016/j.compag.2017.02.017`.

Jennifer Mack, Anatina Trakowski, Florian Rist, Katja Herzog, Reinhard Töpfer, and Volker Steinhage (2017b). Experimental Evaluation of the Performance of Local Shape Descriptors for the Classification of 3D Data in Precision Farming. *Journal of Computer and Communications, Special Edition on Computer Vision*, 5(12), DOI: `10.4236/jcc.2017.512001`.

Jennifer Mack, Frank Schindler, Florian Rist, Katja Herzog, Reinhard Töpfer, and Volker Steinhage (2018). Semantic labeling and reconstruction of grape bunches from 3D range data using a new RGB-D feature descriptor. *Computers and Electronics in Agriculture*, 155:96–102, DOI: `10.1016/j.compag.2018.10.011`.

Jennifer Mack, Florian Rist, Katja Herzog, Reinhard Töpfer, and Volker Steinhage (2019). Making the invisible visible with constraint-based 3D reconstruction in infinitely large model spaces. *Submitted to Computers and Electronics in Agriculture*.

Francesco Marinello, Andrea Pezzuolo, Donato Cillis, and Luigi Sartori (2016). Kinect 3d reconstruction for quantification of grape bunch volume and mass. In *15th International Scientific Conference on Engineering for Rural Development*.

Zoltan Csaba Marton, Radu Bogdan Rusu, and Michael Beetz (2009). On Fast Surface Reconstruction Methods for Large and Noisy Datasets. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3218–3223. DOI: `10.1109/ROBOT.2009.5152628`.

Daniel Maturana and Sebastian Scherer (2015). 3D Convolutional Neural Networks for Landing Zone Detection from LiDAR. In *International Conference on Robotics and Automation*, pages 3471–3478, Seattle, WA, USA. DOI: `10.1109/ICRA.2015.7139679`.

Annalisa Milella, Roberto Marani, Antonio Petitti, and Giulio Reina (2019). In-field high throughput grapevine phenotyping with a consumer-grade depth camera. *Computers and Electronics in Agriculture*, 156:293 – 306, DOI: `10.1016/j.compag.2018.11.026`.

Borja Millan, Santiago Velasco-Forero, Arturo Aquino, and Javier Tardaguila (2018). On-the-Go Grapevine Yield Estimation Using Image Analysis and Boolean Model. *Journal of Sensors*, 2017, DOI: `10.1155/2018/9634752`.

Rainer Muthmann (2007). *The use of plant protection products in the European Union*. Statistical books. eurostat.

Erickson R. Nascimento, Gabriel L. Oliveira, Mario F. M. Campos, Antonio W. Vieira, and William Robson Schwartz (2012). BRAND: A robust appearance and depth descriptor for RGB-D images. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1720–1726. DOI: `10.1109/IROS.2012.6385693`.

Stephen Nuske, Kyle Wilshusen, Supreeth Achar, Luke Yoder, Srinivasa Narasimhan, and Sanjiv Singh (2014). Automated Visual Yield Estimation in Vineyards. *Journal of Field Robotics*, 31(5):837–860, DOI: `10.1002/rob.21541`.

Sebastian Ochmann, Richard Vock, Raoul Wessel, and Reinhard Klein (2016). Automatic reconstruction of parametric building models from indoor point clouds. *Computers & Graphics*, 54:94 – 103, DOI: `10.1016/j.cag.2015.07.008`. Special Issue on CAD/Graphics 2015.

Organisation Internationale de la Vigne et du Vin (2009). Descriptor List for Grape Varieties and Vitis Species, second ed. `http://www.oiv.int/public/medias/2274/code-2e-edition-finale.pdf` [Accessed: 31.07.2019].

Stefan Paulus, Jan Dupuis, Anne-Katrin Mahlein, and Heiner Kuhlmann (2013). Surface feature based classification of plant organs from 3D laserscanned point clouds for plant phenotyping. *BMC Bioinformatics*, 14(1), DOI: `10.1186/1471-2105-14-238`.

Dirk Peters (2018a). 3D reconstruction of wine grapes from front scans using RJMCMC optimization. Lab Report, University of Bonn.

Dirk Peters (2018b). On the automated annotation and reconstruction of grape bunches from 3D range data using a pre-trained CNN. Master's thesis, University of Bonn.

Chakkrit Preuksakarn, Frédéric Boudon, Pascal Ferraro, Jean-Baptiste Durand, Ekko Nikinmaa, and Christophe Godin (2010). Reconstructing plant architecture from 3d laser scanner data. In *6th International Workshop on Functional-Structural Plant Models*, pages 16–18.

Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas (2017a). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85. DOI: `10.1109/CVPR.2017.16`.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas (2017b). PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *International Conference on Neural Information Processing Systems (NIPS)*, pages 5099–5108. Curran Associates, Inc.

T. Rabbani, F. Van Den Heuvel, and G. Vosselman (2006). Segmentation of point clouds using smoothness constraint. In *ISPRS Commission V Symposium 'Image Engineering and Vision Metrology'*.

Hongyu Ren, Russell Stewart, Jiaming Song, Volodymyr Kuleshov, and Stefano Ermon (2018). Adversarial Constraint Learning for Structured Prediction. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 2637–2643. DOI: `10.24963/ijcai.2018/366`.

Florian Rist, Katja Herzog, Jenny Mack, Robert Richter, Volker Steinhage, and Reinhard Töpfer (2018). High-Precision Phenotyping of Grape Bunch Architecture Using Fast 3D Sensor and Automation. *Sensors*, 18(3), DOI: `10.3390/s18030763`.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer International Publishing, DOI: `10.1007/978-3-319-24574-4_28`.

Ribana Roscher, Katja Herzog, Annemarie Kunkel, Anna Kicherer, Reinhard Töpfer, and Wolfgang Förstner (2014). Automated image analysis framework for high-throughput determination of grapevine berry sizes using conditional random fields. *Computers and Electronics in Agriculture*, 100:148–158, DOI: `10.1016/j.compag.2013.11.008`.

Johann Christian Rose, Anna Kicherer, Markus Wieland, Lasse Klingbeil, Reinhard Töpfer, and Heiner Kuhlmann (2016). Towards Automated Large-Scale 3D Phenotyping of Vineyards under Field Conditions. *Sensors*, 16(12), DOI: `10.3390/s16122136`.

Robert Rudolph, Katja Herzog, Reinhard Töpfer, and Volker Steinhage (2018). Efficient identification, localization and quantification of grapevine inflorescences in unprepared field images using Fully Convolutional Networks. *arXiv preprint*, abs/1807.03770, `http://arxiv.org/abs/1807.03770`.

Radu Bogdan Rusu, Nico Blodow, and Michael Beetz (2009). Fast Point Feature Histograms (FPFH) for 3D registration. In *IEEE International Conference on Robotics and Automation*, pages 3212–3217. DOI: `10.1109/ROBOT.2009.5152473`.

Radu Bogdan Rusu and Steve Cousins (2011). 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation*, pages 1–4. DOI: `10.1109/ICRA.2011.5980567`.

Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, and Michael Beetz (2008). Learning Informative Point Classes for the Acquisition of Object Model Maps. In *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 643–650. DOI: `10.1109/ICARCV.2008.4795593`.

Ruwen Schnabel, Roland Wahl, and Reinhard Klein (2007). Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, 26(2):214–226, DOI: `10.1111/j.1467-8659.2007.01016.x`.

Florian Schöler and Volker Steinhage (2015). Automated 3D reconstruction of grape cluster architecture from sensor data for efficient phenotyping. *Computers and Electronics in Agriculture*, 114:163 – 177, DOI: `10.1016/j.compag.2015.04.001`.

Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller (2015). Multi-view Convolutional Neural Networks for 3D Shape Recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 945–953. DOI: `10.1109/ICCV.2015.114`.

Javier Tello, Sergio Cubero, José Blasco, Javier Tardaguila, Nuria Aleixos, and Javier Ibáñez (2016). Application of 2D and 3D image technologies to characterise morphological attributes of grapevine clusters. *Journal of the Science of Food and Agriculture*, 96(13):4575–4583, DOI: `10.1002/jsfa.7675`.

J. Tello and J. Ibáñez (2017). What do we know about grapevine bunch compactness? a state-of-the-art review. *Australian Journal of Grape and Wine Research*, 24(1):6–23, DOI: `10.1111/ajgw.12310`.

Paul Temple, José A. Galindo, Mathieu Acher, and Jean-Marc Jézéquel (2016). Using Machine Learning to Infer Constraints for Product Lines. In *Proceedings of the 20th International Systems and Software Product Line Conference*, SPLC '16, pages 209–218. DOI: `10.1145/2934466.2934472`.

Federico Tombari, Samuele Salti, and Luigi Di Stefano (2010). Unique Signatures of Histograms for Local Surface Description. In *European Conference on Computer Vision*, pages 356–369. DOI: `10.1007/978-3-642-15558-1_26`.

Federico Tombari, Samuele Salti, and Luigi Di Stefano (2011). A combined texture-shape descriptor for enhanced 3D feature matching. In *18th IEEE International Conference on Image Processing*, pages 809–812. DOI: `10.1109/ICIP.2011.6116679`.

Reinhard Töpfer, Ludger Hausmann, Margit Harst, Erika Maul, Eva Zyprian, and Rudolf Eibach (2011). *Fruit, Vegetable and Cereal Science and Bitoechnology*, volume 5, chapter New Horizons for Grapevine Breeding, pages 79–100. Global Science Books, Ltd.

Vladimir Vapnik (1995). *The Nature of Statistical Learning Theory*. Springer Verlag, New York, NY, USA.

Mirwaes Wahabzada, Stefan Paulus, Kristian Kersting, and Anne-Katrin Mahlein (2015). Automated interpretation of 3D laserscanned point clouds for plant organ segmentation. *BMC Bioinformatics*, 16:248–258, DOI: `10.1186/s12859-015-0665-2`.

Bin Wu, Bailang Yu, Qiusheng Wu, Shenjun Yao, Feng Zhao, Weiqing Mao, and Jianping Wu (2017a). A Graph-Based Approach for 3D Building Model Reconstruction from Airborne LiDAR Point Clouds. *Remote Sensing*, 9(1), DOI: `10.3390/rs9010092`.

Kanzhi Wu, Xiaoyang Li, Ravindra Ranasinghe, Gamini Dissanayake, and Yong Liu (2017b). RISAS: A novel rotation, illumination, scale invariant appearance and shape feature. In *IEEE International Conference on Robotics and Automation*, pages 4008–4015. DOI: `10.1109/ICRA.2017.7989461`.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao (2015). 3D ShapeNets: A deep representation for volumetric shapes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920. DOI: `10.1109/CVPR.2015.7298801`.

Longjiao Xin, Lihong Xu, Dawei Li, and Daichang Fu (2014). The 3D Reconstruction of Greenhouse Tomato Plant Based on Real Organ Samples and Parametric L-system. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 9159. DOI: `10.1117/12.2064002`.

Jiaqi Yang, Zhiguo Cao, and Qian Zhang (2016a). A fast and robust local descriptor for 3D point cloud registration. *Information Sciences*, 346–347:163–179, ISSN: `0020-0255`, DOI: `10.1016/j.ins.2016.01.095`.

Jiaqi Yang, Qian Zhang, Ke Xian, Yang Xiao, and Zhiguo Cao (2016b). Rotational contour signatures for robust local surface description. In *IEEE International Conference on Image Processing*, pages 3598–3602. DOI: `10.1109/ICIP.2016.7533030`.

Yi-Ting Yeh, Lingfeng Yang, Matthew Watson, Noah D. Goodman, and Pat Hanrahan (2012). Synthesizing Open Worlds with Constraints Using Locally Annealed Reversible Jump MCMC. *ACM Transaction on Graphics*, 31(4):56:1–56:11, DOI: `10.1145/2185520.2185552`.