Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C | Dissertationen | Heft Nr. 812

Przemyslaw P. Polewski

Reconstruction of standing and fallen single dead trees in forested areas from LiDAR data and aerial imagery

München 2018

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325 | ISBN 978-3-7696-5224-6

Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C | Dissertationen | Heft Nr. 812

Reconstruction of standing and fallen single dead trees in forested areas from LiDAR data and aerial imagery

Von der Ingenieurfakultät Bau Geo Umwelt

der Technischen Universität München

zur Erlangung des Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

Vorgelegt von

M.Sc. Przemyslaw P. Polewski

Geboren am 03.02.1984 in Poznan

München 2018

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325 ISBN 978-3-7696-5224-6

Adresse der DGK:

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften (DGK)

Alfons-Goppel-Straße 11 ● D – 80 539 München

Telefon +49 – 331 – 288 1685 ● Telefax +49 – 331 – 288 1759

E-Mail post@dgk.badw.de ● http://www.dgk.badw.de

Prüfungskommission:

Vorsitzender: Prof. Dr.-Ing. habil. Thomas Wunderlich

Referent: Prof. Dr.-Ing. Uwe Stilla

Korreferenten: Prof. Dr.-Ing. Helmut Mayer (Universität der Bundeswehr München)
Prof. Dr.-Ing. Peter Krzystek (Hochschule München)

Tag der mündlichen Prüfung: 06.12.2017

ISSN 0065-5325 ISBN 978-3-7696-5224-6

Abstract

Dead wood constitutes an important component of forest ecosystems due to its various roles: preserving biodiversity, ensuring forest stand succession, as well as serving as a sink for sequestering greenhouse gas emissions. Therefore, the mapping of dead wood is an important issue for scientists studying forest environments. Currently, the predominant approach relies on time-consuming field inventories. On the other hand, during the last 15 years remote sensing techniques for surveying forested areas have become widespread. Particularly the advancement of laser scanning technology allows for an unprecedented level of detail in mapping the understory layer. The principal goal of this thesis is to develop methods for mapping various forms of dead wood under challenging scenarios, based on 3D laser scanning data complemented by aerial imagery, whose infrared spectral band is known to facilitate detection of dead vegetation.

First, the detection of fallen trees from point clouds was considered. A general framework is proposed for delineating elongated structures. Initially, equally-sized object *parts* (primitives) that are likely to form objects together are found through contextual classification using a conditional random field (CRF). Then, the primitives are merged into objects through a spectral clustering algorithm. The components of the CRF and the parameters of spectral clustering are learned from training data. This framework is then instantiated for detecting fallen stems in ALS point clouds. Regarding TLS data, an approach for cylinder reconstruction based on voting in parameter accumulation space is proposed. By utilizing a kernel density estimator to model the parameter space, the accumulation cell size is determined automatically. The method is then applied to TLS point clouds, where the cylindrical shapes of fallen stems are well-preserved.

In case of standing dead trees, three scenarios were studied. For detecting dead tree crowns only using aerial CIR imagery, a level set segmentation method with shape and intensity priors is adapted from a medical imaging application. Standing dead trunks without crowns were detected from ALS point clouds using a newly introduced 3D shape descriptor, the Free Shape Context (FSC), which generalizes cylindrical Shape Contexts by allowing a more heterogeneous structure. The combination of ALS data and aerial imagery was also considered for dead tree detection. This is done by segmenting the point cloud and projecting the clusters onto the image plane to obtain bounding polygons for each tree. Spectral features are derived from image pixels.

The problems of standing dead tree detection were studied in the context of semi-supervised and active learning, two techniques for accelerating the training of supervised models. The semi-supervised Shannon entropy-regularized logistic model is generalized to handle Renyi entropies. Also, a method for combining the two learning paradigms on the grounds of minimizing the entropy of class posterior probabilities is introduced, extending Expected Error Reduction.

Experimental results reveal that under a moderate overstory cover, fallen stems can be recovered quite reliably from ALS data. A considerable gain in detection quality can be associated with the CRF-based contextual classification, whereas the learned components of both the CRF and the spectral clustering showed good generalization abilities. The proposed cylinder detection method for TLS data led to an increased stem reconstruction quality compared to various sample consensus (SAC) baselines by up to 10 percentage points (pp). Overall, both the TLS and ALS processing pipelines achieved detection completeness over 75% with an error rate below 20%. The utility of the near infrared band for dead vegetation detection was confirmed, although the method utilizing only 2D imagery sometimes confused dead trees with open ground patches due to similar hues. This was remedied by including 3D information from the point cloud in the fused data scenario. The FSC descriptor was found to yield a good balance between feature space size and discriminative properties. Finally, the results show that a benefit from combining active and semisupervised learning of up to 10 pp is attainable, mainly for small training sets.

Zusammenfassung

Totholz ist ein wichtiges Element forstlicher Ökosysteme wegen seiner Rollen in der Erhaltung der Artenvielfalt, der Nachhaltigkeitssicherung von Baumbeständen sowie als Kohlenstoffspeicher für Treibhausgasemissionen. Die Kartierung von Totholz stellt daher im Bereich der waldökologischen Forschung ein hochaktuelles und interessantes Thema dar. Derzeit werden solche Informationen meistens durch aufwändige Inventuren bzw. manuelle Messkampagnen gewonnen. Andererseits haben sich Fernerkundungsmethoden bei der Charakterisierung von Waldstrukturen, vor allem das Laserscanning, in den letzten 15 Jahren sehr verbreitet. Das Hauptziel dieser Arbeit ist, Methoden für die automatische Kartierung verschiedener Formen von Totholz zu entwickeln, die auf Daten terrestrischer (TLS) und flugzeuggetragener (ALS) Laserscanner sowie multispektralen CIR Luftbildern basieren.

Im ersten Teil der Arbeit wird die Segmentierung von liegenden Stämmen aus Laserscanner-Punktwolken behandelt, wobei ein Framework zur Erkennung von länglichen Strukturen vorgeschlagen wird. Zuerst werden Objektteile, die mit hoher Wahrscheinlichkeit zusammen ein Objekt bilden, durch kontextuelle Klassifikation mittels Conditional Random Fields (CRF) gefunden. Danach werden die Teile anhand einer spektraler Clusteranalyse in Objekte zusammengefügt. Die Parameter des Clusterverfahrens sowie der CRFs werden aus Trainingsdaten gelernt. Das Framework wird anschließend für die konkrete Aufgabe der Erkennung von liegenden Stämmen aus ALS-Daten instanziiert. Für TLS-Daten wird ein Verfahren zur Rekonstruktion von Zylinderformen aus dichten Punktwolken eingeführt, das auf der Modellierung des Parameterraums mittels eines Kerndichteschätzers beruht. Das Verfahren wird auf TLS-Punktwolken angewendet, in denen die Zylinderform der gefallenen Stämme gut repräsentiert wird.

Im Weiteren werden drei Szenarien zur Erkennung von stehenden toten Bäumen untersucht. Im ersten Szenario werden lediglich CIR Luftbilder als Eingangsdaten benutzt und mit einem Level-Set-Segmentierungsverfahren mit Form- und Intensitäts-Apriori-Wissen verarbeitet. In einem zweiten Verfahren wird für die Erkennung stehender Stämme ohne Krone aus ALS-Punktwolken der Freie Form-Kontext als ein neuer 3D-Formdeskriptor eingeführt, der eine Verallgemeinerung von zylindrischen Form-Kontexten mit heterogener Struktur darstellt. Schließlich wird die Totholzerkennung mit ALS-Daten und Luftbildern behandelt. Die Punktwolke wird in Einzelbäume segmentiert, deren Umringe auf die Bildebene projiziert werden und somit eine Umgebung zur Ableitung spektraler Baummerkmale aus Pixelwerten bilden.

Die Methoden zur Erkennung von stehendem Totholz werden mittels aktivem und teilüberwachtem Lernen untersucht. Das logistische, mit Shannon-Entropie regularisierte Regressionsmodell wird zur Familie von Renyi-Entropien verallgemeinert. Des Weiteren wird eine Methode zur Verbindung des aktiven und teilüberwachten Lernens innerhalb eines Verfahrens basierend auf der Entropieminimierung von a posteriori Klassenwahrscheinlichkeiten als Erweiterung des Expected Error Reduction Algorithmus eingeführt.

Die experimentellen Ergebnisse zeigen, dass liegende Stämme bei moderater Abdeckung durch das Kronendach mit guter Genauigkeit aus ALS-Daten rekonstruiert werden können. Die Anwendung von CRFs führte zu einer beträchtlichen Qualitätssteigerung der Ergebnisse, während die gelernten Teile des CRFs und der spektralen Clusteranalyse gutes Verallgemeinerungsvermögen aufwiesen. Das eingeführte Verfahren zur Zylinderrekonstruktion zeigte bei TLS-Daten eine Verbesserung von bis zu 10 Prozentpunkten (pp) gegenüber Sample-Consensus-Methoden. Für beide Datenquellen (ALS und TLS) konnte eine Erkennungs-Sensitivität von über 75% bei einer Fehlerrate unter 20% erreicht werden. Die Nützlichkeit des infraroten Bildkanals bei Erkennung toter Vegetation wurde bestätigt, obwohl tote Bäume mit Bereichen offenen Bodens aufgrund ähnlicher Pixelfarben verwechselt werden können. Dieses Problem wurde mit der Einführung von 3D-Informationen im Datenfusionsszenario verringert. Der Freie Form-Kontext bietet ein Balance zwischen der Größe des Merkmalsraums und Unterscheidungsvermögen. Die Ergebnisse weisen auch darauf hin, dass das Kombinieren des aktiven und teilüberwachten Lernens für kleine Datensätze einen Mehrwert von bis zu 10 pp erbringt.

Contents

List of Abbreviations

Abbreviation	Description	Page
ALS	Aerial Laser Scanning	15
CHM	Canopy Height Model	21
CIR	Color infrared	20
CRF	Conditional Random Field	26
DTM	Digital Terrain Model	19
GNSS	Global Navigation Satellite System	15
iid	independent and identically distributed	34
IMU	Inertial Measurement Unit	15
KDE	Kernel Density Estimator	39
MLE	Maximum Likelihood Estimator	34
NIR	Near Infrared	91
PDF	Probability Density Function	39
SAC	Sample Consensus	43
TLS	Terrestrial Laser Scanning	15

List of Figures

List of Tables

1 Introduction

Dead wood is an important component of any forest ecosystem. It is estimated that up to one third of all plant and animal species living in temperate forests depend on dead wood for their survival [Stokland et al., 2012; Freedman et al., 1996]. Considering the fact that forests are one of Earth's most diverse and rich environments in terms of the inhabitant species, it is apparent that dead wood has a prominent role in the preservation of wildlife biodiversity. However, it also fulfills several other important roles. The presence of decaying woody debris was found to be positively correlated with tree regeneration, which highlights its role in forest stand succession and sustainability [Weaver et al., 2009]. This is related to the participation of downed wood in forest nutrient cycles, due to releasing carbon and nutrient elements (nitrogen, phosphorus) into the soil [Harmon et al., 1986; Finér et al., 2003]. Also, it possesses a geomorphic function, which can be categorized into effects on landforms as well as transport and storage of soil and sediment [Harmon et al., 1986]. Last but not least, the matter of forest carbon stocks should be considered. It was estimated that up to 11% of all greenhouse gas emissions in the USA are sequestered in forests or forest products per year [Woodall et al., 2008]. Also, estimates put the percentage of total forest carbon stocks contained within dead woody material at 14%, which shows that dead wood must be taken into account as a significant factor when approximating the total sequestered carbon pool. For these reasons, obtaining accurate information about the distribution of dead trees in forests is important for entities and institutions concerned with comprehensive monitoring of forested areas. There are three target consumer groups of this information: (i) environmental scientists studying stand succession, nutrient cycles and wildlife, (ii) private forest owners wishing to make their contribution to protecting forest biodiversity, and (iii) governmental and inter-governmental institutions which are legally obligated to monitor habitats due to national law or international treaties. Regarding the latter category, the European Union (EU) maintains a list of protected areas which provide habitats to rare or threatened plant and animal species, with the aim to preserve biodiversity - the Natura2000 program [Sundseth & Creed, 2008]. Since a total of 18% of the EU's land area is designated as Natura2000 relevant, monitoring such a vast territory would not be possible when relying only on field surveys and manual methods.

Aside from the ecological aspect, monitoring dead trees is also relevant from a disaster management standpoint. A vehement storm can devastate parts of entire forest stands, resulting in large swaths of windthrown tree-covered land. Such a vast quantity of dead woody debris can be the starting point of an insect infestation, which poses a serious threat to the remaining living trees [Lidskog & Sjödin, 2016]. Moreover, the fallen trees in such disaster areas are often clustered together so densely that is is dangerous for human surveying personnel to conduct an on-site inventory, further highlighting the need for remote sensing methods. Also, in case of an ongoing infestation, the mapping of standing dead trees can provide information about the spreading patterns of the disease [Lausch et al., 2013].

The current prevalent method for acquiring information about dead wood in forested areas relies on manual field measurements, which are usually associated with a high cost and provide

only limited spatial coverage. On the other hand, remote sensing techniques based on laser scanning as well as aerial imagery could help overcome both of these limitations, allowing for large-scale mapping of dead wood in forests and reducing the unit cost of measurements.

Aerial Laser Scanning (ALS) is a surveying technique which appeared in the 1980s for the purpose of large-scale terrain mapping. Although first attempts may be traced back to the 1970s, it was not until the following decade that ALS experienced a significant boost in accuracy owing to the integration of Global Navigation Satellite System (GNSS) and Inertial Measurement Unit (IMU) data with the laser scanner [Ackermann, 1999]. This laid the foundation for the introduction of commercial scanners, and stimulated the growth of the emerging ecosystem of ALS-related services, applications and solutions. Over the last two decades, the remote sensing community has witnessed a rapid development of ALS technology [Krzystek & Polewski, 2017]. While the commercial devices released in the late 1990s featured scanning rates around 10 kHz [Baltsavias, 1999], currently pulse rates of 400 kHz are not uncommon [Riegl GmbH, 2017]. This is reflected by a dramatic increase in the density of the acquired point cloud, from 1 point per 20 m^2 to over 30 points per m^2 for modern systems, for a flight altitude of 1000 m. An important milestone of ALS development was the advent of full-waveform (FWF) systems, which in contrast to previous discrete return systems are capable of recording the entire waveform reflected by the measured objects, not just the reflected point coordinates. This makes it possible to derive radiometric properties (e.g. intensity) of scattering objects by deconvolving the received waveform from the sensor's output signal and fitting a model describing the single reflections [Jutzi & Stilla, 2006; Wagner et al., 2006]. Recently, two related techniques started gaining momentum in practical applications of ALS: single-photon LiDAR [Swatantran et al., 2016] as well as Geiger-mode LiDAR [Stoker et al., 2016]. Both rely on extremely accurate receiver arrays capable of detecting individual photons, which reduces the necessary power for outgoing pulses and allows for greater flight altitudes, increasing data collection speeds. Although these techniques have delivered promising results in preliminary studies, the extreme sensitivity of the applied sensors comes at a cost of increased susceptibility to solar noise.

Precision forestry benefited greatly from laser scanning technology, due to the possibility of obtaining precise information about large areas quickly, without the need to conduct expensive and time-consuming field inventories. This can be attributed to the capability of penetrating the canopy cover as opposed to passive data collection systems, resulting in 3D structural information. Starting from early applications of mapping vegetation height profiles with ALS in the 1980s [Nelson et al., 1984], the field has experienced steady growth, extending to new forest types and estimating new types of vegetation parameters [Monnet, 2012]. Around the early 2000s, the research pace accelerated further, making laser scanning of forests one of the driving forces of laser scanning development in general [Hyyppä et al., 2012]. Initially, due to the relatively low density of scanned point clouds, research efforts concentrated on so-called *area-based methods*, which aimed at providing aggregate properties of entire forest stands such as leaf area index (LAI) [Morsdorf et al., 2006], forest land cover characteristics [Antonarakis et al., 2008], mean tree height [Naesset, 1997], 3D vegetation structure [Lindberg et al., 2012b] and many others [Hyyppä, 2011]. Eventually, significant advancements in LiDAR hardware made it possible to obtain point clouds detailed enough to distinguish individual trees, which provided a strong impulse for the development of *single tree-based* methods for segmentation [Lindberg & Holmgren, 2017], species classification [Fassnacht et al., 2016] and 3D parameter extraction. Today, both area-based and single-tree based approaches form a crucial part of the ALS research landscape. It is important to note that aerial laser scanning in forests is not a purely academic topic. Many of the developed methods have reached a maturity level permitting operational deployment in large-scale inventories. Indeed, several countries have set up national programs for mapping the entire territory

by laser scanning [Monnet, 2012], while in Nordic countries forest stand-level inventory based on ALS data is now routine [Hyyppä et al., 2012].

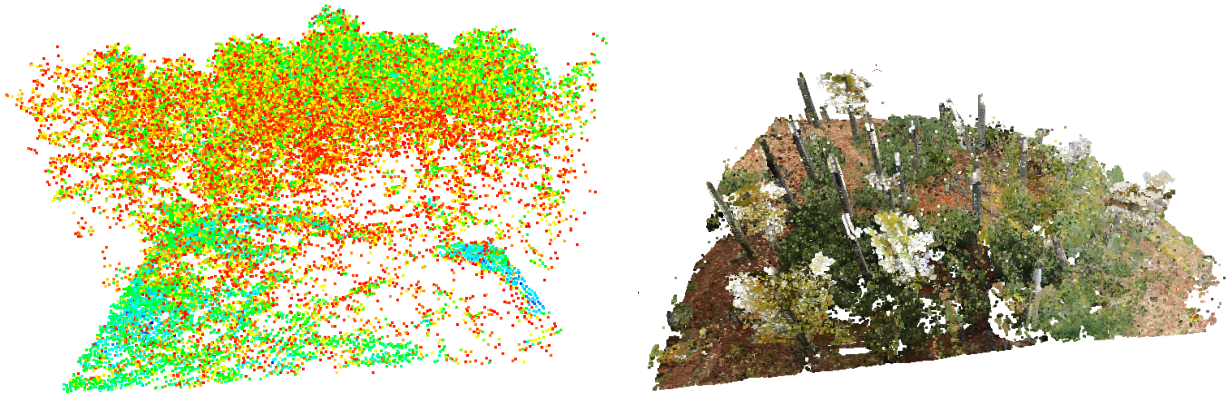
Although ALS mapping has proven extremely useful for forest applications, it is not without certain weaknesses. First, under conditions of dense tree canopy cover, the laser beam is strongly attenuated in the upper forest layers, resulting in poor to moderate laser reflection density near the ground. Also, the point density resulting from operational flight heights that ensure a big enough area coverage is usually insufficient to capture fine details like tree stems or the branch structure. Terrestrial laser scanning (TLS) addresses these problems, trading spatial coverage for a dramatically increased scanning resolution. The dense point clouds acquired from a terrestrial perspective are a natural complement to ALS data. During the last 15 years, the application of TLS technology has changed the way forest inventories are conducted [Liang et al., 2016], enabling the calculation of new types of information about forest stands. This includes mapping tree stems by determining their height and diameter [Liang & Hyyppä, 2013] or precise cross section shape [Wang et al., 2017] as well as the reconstruction of the entire branch structure [Raumonen et al., 2013]. Because of the high point density, TLS is an invaluable tool for the task of obtaining comprehensive forest stand information.

Another important source of remote sensing data with a long-standing tradition in forestry applications is aerial imaging with passive sensors. Initial attempts at aerial mapping of forests date back to the period World War I, prompted by increased availability of aircraft and improvements in hardware [Standish, 1945]. For decades, the use of aerial imagery was restricted to visual interpretation by a human operator, which inherently limited the spatial coverage. In the mid 1980s, research towards automatic delineation of individual tree crowns from passive optical imagery started gaining momentum [Zhen et al., 2016]. Today, a multitude of approaches exist, and this is still an active research topic [Ke & Quackenbush, 2011]. One of the major disadvantages of passive aerial imagery in forest applications is the inability to penetrate the canopy cover, which means that only the top layer of the forest can be mapped. This limits potential applications to studies of the understory when the canopy layer is dense. On the other hand, instruments which extend the range of captured radiation beyond the visible spectrum, such as multi- and hyperspectral sensors, offer a unique opportunity for characterizing material types based on their spectral signatures [Xie et al., 2008]. The distinction between the two sensor types is made based on the number and width of spectral bands. Multispectral sensors capture up to 10 relatively wide (50-500 nm) bands, typically the 3 visible ranges (RGB) and one or more near, short-wave or thermal infrared channels, whereas hyperspectral instruments can record hundreds of narrow (10-20 nm) bands within the infrared, visible and ultraviolet ranges. While this great level of detail associated with hyperspectral imagery can benefit object classification tasks, it also poses some specific challenges [Bioucas-Dias et al., 2012]. Due to the lower spatial resolution of these sensors, multiple unrelated materials may be captured within a single image pixel, necessitating a method for *spectral unmixing*. Moreover, the high dimensionality of the raw spectral data dramatically increases the requirements for training samples, causing traditional machine learning approaches to break down in the face of a sparsely populated feature space. Therefore, developing efficient methods for hyperspectral data processing remains an active research topic. Regarding the use of multispectral imagery in forestry, particularly the near infrared bands are important for tree health monitoring applications, since the reflection patterns of chlorophyll pigments within this spectral range can be used to detect and quantify a decline in chlorophyll content [Jensen, 2006].

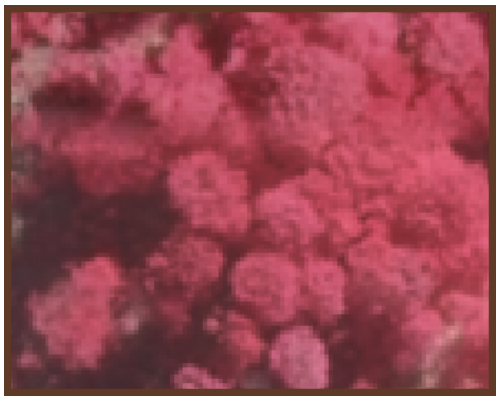
A considerable limitation of both laser scanning and passive sensors is their dependence on weather conditions. Increased presence of water due to fog, rain or mist can significantly alter the atmospheric transmittance of the laser beam [Wilkes et al., 2017], whereas passive sensors require sufficient reflected natural light, excluding data acquisition in darkness. To mitigate

this disadvantage, some research effort has been dedicated to studying synthetic aperture radar systems (SAR) for forest mapping, due to their relative independence of weather and lighting conditions. Until recently, research has focused on large-scale (stand level) applications such as biomass [Sandberg et al., 2011] or canopy height [Balzter et al., 2007] estimation. With the advancement of SAR interferometry sensors and techniques, generation of a 3D point cloud with a resolution high enough for single tree segmentation became possible. This topic was studied by Schmitt & Stilla [2014] as well as Stilla et al. [2014]. Recently, Magnard et al. [2016] compared the quality of single tree segmentation based on point clouds obtained from SAR, laser scanning and aerial photogrammetry. They found that in case of SAR, most of the scattering occurs near the top of the canopy, with only limited penetration into deeper vegetation layers. Consequently, it seems that SAR interferometry techniques are currently not an alternative for laser scanning as a tool for mapping the forest understory (e.g. fallen trees).

Since the beginning of the century there has been a perceivable trend of combining data from multiple types of sensors with the purpose of obtaining a more complete and multifaceted description of the target forested area. Notably fusion of multispectral and ALS data has received much attention [Man et al., 2014]. Figure 1.1 illustrates the three aforementioned types of remote sensing data (ALS, TLS, multispectral imagery) which were captured for the same forest plot.



(a) (b)



(c)

Figure 1.1: A forest plot captured with three sensors: ALS (a), TLS (b), and multispectral with near infrared (c). The colors in (a) represent the laser point intensity from low (red) to high (blue), whereas the TLS point cloud (b) is colored with RGB values captured by an optical camera. (c) shows a false-color CIR image with channels NIR, red, green. The understory is poorly represented in both (a) and (c) due to canopy cover, whereas (b) does not capture tree crowns.

1.1 Motivation

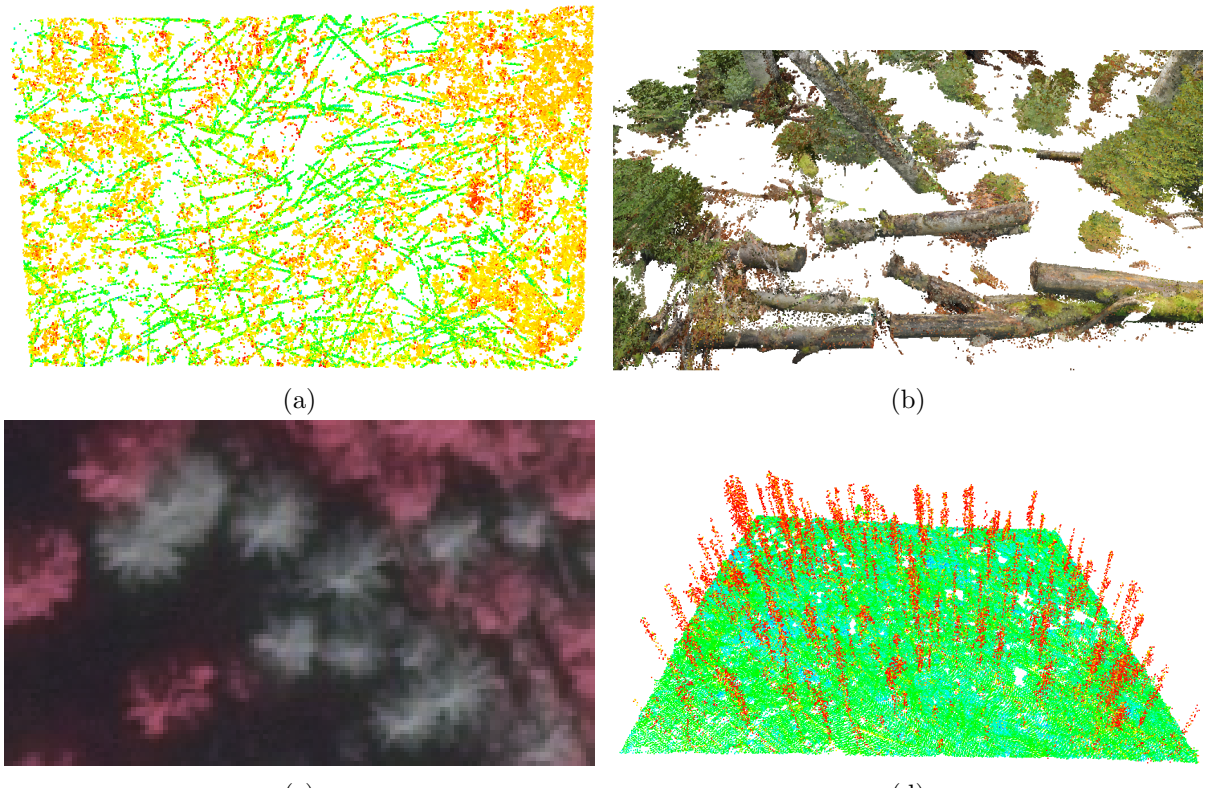
Although methods for deriving a myriad of forest parameters and characteristics from laser scanning and aerial imagery data have been developed, it seems that up to now the aspect of dead wood mapping has remained relatively unexplored. Given the success of remote sensing methods in forestry applications, and considering that most dead wood mapping campaigns are carried out by manual surveying, it is clear that remote sensing techniques offer a unique opportunity for automatizing this laborious task. This makes it possible for the first time to obtain complete information about the distribution of dead wood in forested areas occupying hundreds or even thousands of square kilometers.

From a pattern recognition standpoint, the task of single dead tree detection is quite inhomogeneous, with a degree of difficulty ranging from moderate to high. In case of fallen stems, the problem is complicated by the presence of near-ground objects such as shrubs/ground vegetation, tree stumps and digital terrain model (DTM) artifacts. Many fallen trunks may be clustered together in close proximity, forming non-trivial 3D structures (Fig. 1.2a). Furthermore, it is not uncommon for the fallen stems to fracture into non-collinear segments. For sparse (ALS) point clouds, the point density is too low to capture the cylindrical shape of the stems, eliminating an important visual cue. Instead, the detection process must rely on finding elongated linear structures formed by adjacent laser points. The cylindrical shape is better preserved in TLS point clouds (Fig. 1.2b), however the terrestrial perspective also gives rise to certain challenges: occlusion of the target objects by standing trees and shrubs, incompleteness of captured objects due to scanning positions, as well as varying point density associated with distance to the scanner. Owing to the aforementioned difficulties, the extraction of individual fallen stems from laser data is a non-trivial task requiring complex methodologies. In comparison, detecting standing dead trees, also referred to as snags, seems somewhat simpler, due to the larger distances between objects (reduced overlap) and their verticality resulting from the phenomenon of gravitropism (trees growing in the opposite direction of gravitational pull). Additionally, dead trees still possessing a crown produce a distinctive reflectance spectrum in the near infrared radiation range, which enables their detection from aerially mounted passive multispectral sensors (Fig. 1.2c). On the other hand, accurate delineation of single dead crowns solely from aerial images remains a challenge in the presence of large groupings of dead trees, as well as because of potential similarities in reflectance values shared with other materials, e.g. open ground or soil. The incorporation of 3D data from point clouds could alleviate some of these problems. In contrast, standing dead stems which have lost their crowns are hardly recognizable within aerial imagery acquired from the nadir view due to their small cross-section area, bordering on the sensor's ground resolution. Therefore, full 3D information provided by point clouds is advantageous for this scenario. The dead trunks exhibit some variability in appearance due to residual branches and the potential presence of ground vegetation (Fig. 1.2d). A further difficulty is their resemblance to more slender trees.

In real-world remote sensing applications, oftentimes the amount of available raw data is vast. On the other hand, the machine learning community is concerned with developing highly scalable methods for data interpretation, therefore it is natural that machine learning techniques are becoming increasingly prevalent as parts of remote sensing workflows [Tuia et al., 2014; Camps-Valls et al., 2016; Waske et al., 2009]. In the context of vast datasets and classification tasks, it becomes prohibitively expensive to acquire class labels for all available data. Two paradigms have evolved to remedy this problem. First, *active learning* aims at interactively selecting a small subset of labeled training examples from the data pool which contain the most representative information about the entire pool, by sending queries to an external labeling agent ('oracle'). Second, *semi-supervised* learning assumes that only a small percentage of the data are associated with labels,

and instead attempts to refine the model learned from the labeled data with the information contained in the unlabeled part. Since both paradigms have been successfully applied to remote sensing problems [Camps-Valls, 2009; Tuia et al., 2014] and at the same time they are based on different, complementary principles, it is a highly interesting research question to investigate the synergy between them when applied together as part of a unified framework.

Another prominent class of machine learning methods which gradually found applications within remote sensing are *probabilistic graphical models*, a tool for compactly describing complex, structured probability distributions [Koller & Friedman, 2009]. Graphical models have the ability to express the structure of conditional independence between the modeled variables, which allows the introduction of *context* into classification tasks. Other than the usual classification procedure, where class labels are assigned independently, in case of graphical models the labels of neighboring objects influence each other, making it necessary to calculate a *joint* labeling of all objects simultaneously. In light of the considerable success of probabilistic graphical models for a wide array of computer vision and remote sensing problems, it is hypothesized that contextual classification methods could also be useful for extracting linear structures from point clouds by modeling their spatial interactions.



(a) (b) (c) (d)

Figure 1.2: Various forms of dead wood captured in 3D point clouds and aerial imagery. (a) ALS point cloud with visible fallen stems (green), color by laser reflection intensity. (b) TLS point cloud with fallen stems, color by RGB camera. (c) Color infrared (CIR) aerial photo with dead tree crowns (gray). (d) ALS point cloud of area with standing dead trunks (red), color by laser reflection intensity. In (a),(b) the DTM was removed for better visualization.

1.2 State of the art

In this section, the results of prior work is summarized which has been done on the detection of both fallen and standing dead trees using the three relevant types of sensors (TLS, ALS, multispectral imagery). Also, an overview of methodological results is provided from the fields of machine learning, computer vision and remote sensing which are related to the development of the presented methods.

1.2.1 Dead tree detection

ALS point clouds

Up to now, most research efforts have concentrated on the area-based estimation of the proportion [Vehmas et al., 2011] and volume [Pesonen et al., 2008] of downed stems using regression methods (e.g. ordinary linear regression or random forest [Breiman, 2001]). The independent variables are usually derived from laser point statistics such as height distribution, percentage of ground/vegetation returns, or topographic metrics [Martinuzzi et al., 2009]. On the other hand, the problem of detecting individual fallen trees from ALS point clouds has not been extensively investigated.

Among the first methodological studies was the work of Blanchard et al. [2011]. This approach relies on rasterizing the point cloud with respect to different metrics which quantify elevation and scene information. A vector thematic layer is created and polygons are manually masked in order to retain polygons that best match the shape of a downed stem. Refinement of this layer is achieved by setting various thresholds determined through visual interpretation. Finally, a multiresolution, bottom-up pairwise region merging algorithm is applied on the raster and vector layers to produce an object-level segmentation. Each object is then classified into one of three categories: downed logs, canopy cover and ground using a total of 113 individual refinement and classification rules. The authors report a classification accuracy of 73% and note that their approach has some difficulty in situations like proximity of the stems to ground vegetation as well as large clusters of logs. Also, the task of constructing the classification rules and setting the various thresholds based on visual inspection may take days of the analyst's time and its result is not easily transferable to new plots. Although the evaluation of the results was carried out in an automatic fashion based on spatial overlap of polygons, the fallen tree reference data itself does not originate from field works - instead, it was derived from the input LiDAR point cloud in combination with orthophotographs.

Muecke et al. [2013] also perform the classification on a vectorized layer derived from the binarized point cloud filtered based on distance to the DTM. Additionally, to remove ground vegetation and shrubs, a pulse echo width filter is applied. The authors show that it is possible to reliably detect stems which distinguish themselves well upon the DTM. However, they note that applying the pulse echo width filter together with the vectorized object shape features is not always enough to separate stems from densely intertwined ground vegetation and piles of twigs. The classification relies on the value of the area-perimeter ratio of the polygons lying within a specified range which corresponds to the shape of elongated objects. This is a potential limitation of their method, since for multiple overlapping stems lying in close proximity, the vectorized polygon may have a shape that is not consistent with such an area-perimeter ratio. The evaluation of the results is performed manually by a human expert, which introduces a degree of subjectivity.

The next study, due to Lindberg et al. [2013], is based on analyzing height homogeneity of points on raster cells with two resolutions. A line template matching algorithm is proposed, where

the templates have a rectangular shape with fixed width and height. A voting scheme is then employed, wherein each raster cell votes for line parameters mostly supported by the points inside it. An important methodological advancement is the attempt to model not only the candidate stem, but also its immediate surroundings in order to distinguish between actual fallen trees and ground vegetation or other non-relevant objects. A potential drawback of this approach is that the entire modeling of shape is done explicitly using quite simple features and based on a multitude of user-defined thresholds. The authors note themselves that the choice of parameters was geared towards the characteristics of the study area. Also, this method only detects tree segments of equal length and there is no attempt to merge individual segments. This precludes successful detection of trees whose parts are not ideally linear. Finally, an automatic validation of the obtained results was made difficult by positioning errors in the field measurements, which resulted in a rather lenient criterion for matching the detected and reference trees (10 m horizontal distance and 30 degrees deviation in planimetric direction).

In a recent study, Nyström et al. [2014] also apply line template matching. In contrast to [Lindberg et al., 2013], they first derive an object height model (OHM) based on the active contour surface algorithm [Elmqvist, 2002]. This OHM layer is defined on a raster grid with a resolution of 10 cm and represents the DTM-normalized heights of objects close to the ground. This can be seen as an alternative to the moving least squares interpolation used in [Muecke et al., 2013]. The matching is done on the OHM raster cells with rectangular templates of fixed length and 4 width levels. The templates having the highest correlation with the OHM are iteratively picked as stem candidates. Each candidate undergoes a classification step using linear discriminant analysis manually trained on a set of negative and positive examples based on a set of simple point height statistics within the template. The purpose of this step is to distinguish trees from other linear objects, e.g. edges, ditches and roads. The fixed length of the templates and the lack of any merging step make it impossible for this approach to detect trees of different length and of a more complex shape. Finally, similarly to the previous study, the evaluation method for linking reference and detected trees seems too permissive, as it allows a maximum distance of 12 m and up to 30 degrees of directional deviation.

Regarding the detection of snags from LiDAR point clouds, there have been relatively few contributions which focus on the extraction of individual standing dead trees. Korpela et al. [2010] investigated the utility of full waveform LiDAR reflection intensity statistics for the purpose of discriminating between tree species as well as between dead and living trees. They found that dead trees exhibited an intensity lower by about 40-60% compared to their living counterparts. However, in this study no attempt was made to delineate single trees within the point clouds. Instead, the LiDAR features were extracted around reference tree positions which were known apriori. One of the first studies to present a full methodology designed specifically for the delineation of single snags is due to Yao et al. [2012]. A two-step approach was proposed: first, single trees were segmented using the Normalized Cut algorithm, and then each found tree was classified using a support vector machine as either dead or living based on a collection of radiometric (laser reflection intensity and pulse width statistics) as well as geometric (crown shape, point height distribution) predictors. Classification accuracies of 0.71-0.73 were obtained depending on the presence/absence of foliage. Casas et al. [2016] follow a similar strategy, first applying watershed segmentation to the canopy height model (CHM) to yield single trees, which are subsequently classified using a Gaussian Process method, based on structural and intensity features. The reported accuracy is 0.92, however it should be noted that the experiment was carried out in a forest which experienced a megafire. This means that the damage to the trees was more severe compared to dying off due to an insect infestation or drought, which could have resulted in more extreme differences in LiDAR reflection intensities with respect to living vegetation. In [Wing et al., 2015], the order of the two aforementioned steps is reversed: initially the LiDAR points are

labeled as 'snag' or 'alive' using reflection intensity and geometric features derived from spherical and cylindrical neighborhoods. The points classified as belonging to snags then enter a CHM calculation and smoothing step, followed by local maxima detection to yield the individual snag positions. A drawback of this method is that the point classification step is based on a set of empirically defined rules with a number of thresholds which have to be manually tuned. Also, the reported detection accuracies were not satisfactory, reaching 0.59 despite a rather lenient positional error threshold of up to 4.5 m for matching of reference and detected trees.

Multispectral aerial imagery

The use of infrared bands for assessing vegetation health based on aerial and satellite imagery is quite widespread (e.g. [Wang et al., 2007], [Vogelmann, 1990]). Both the intensity of the infrared channels and various indices derived from it, e.g. NDVI [Tucker, 1979], are useful for distinguishing between dead wood and living vegetation due to the reflectance differences in these spectral bands [Jensen, 2006]. Several authors have developed manual or semi-automatic individual snag detection methods which make use of this fact ([Bütler & Schlaepfer, 2004], [Heurich et al., 2010], [Pasher & King, 2009]). The disadvantage of these approaches is the necessity to involve a human expert in the processing, which may limit their applicability to large areas. Note that because of the nature of passive sensors, only the top forest layer information can be captured, therefore, it is not possible to detect fallen trees in stands with even a moderate canopy cover (due to shadows, occlusion). Some studies describe automatic methods for estimating the total dead tree biomass from multispectral imagery [Eskelson et al., 2012], as well as the classification of snags at image raster level [Kelly et al., 2004; Nielsen et al., 2014]. As in the previously discussed case of ALS data, studies on individual snag delineation are few. Lee & Cho [2006] apply a local pixel intensity maxima filtering algorithm to isolate individual tree crowns and investigate the use of 4 spectral (RGB + near infrared) bands for discriminating between infected and healthy trees. They note that the quality of the crown detection is sensitive to the correct choice of the window size parameter for filtering. More recently, Bhattarai et al. [2012] adopt a similar strategy for classifying Sirex-infested pines from 8-band WorldView-2 imagery. They first perform delineation of individual tree crowns using the algorithm by Ke et al. [2010b], which involves active contour segmentation and subsequent refinement using template matching and heuristic rules. Each tree crown is then classified as either infested or healthy based on features derived from the 8 spectral bands. The overall classification accuracy at the object level attains 0.81, however the authors observe that the delineation algorithm failed to find a significant portion of the smaller crowns and point out that this part of their method has improvement potential.

TLS point clouds

The author is not aware of any published complete studies regarding attempts to detect and reconstruct individual dead trees from TLS point clouds in a fully automatic manner, however a *semi-automatic* method requiring the interaction of a human operator was presented in [Grigillo et al., 2015]. The closest related application is the segmentation of standing tree stems in TLS point clouds. Lindberg et al. [2012a] project candidate stem points onto a 2D plane and apply the Hough transform to find circles corresponding to potential stems. Schilling et al. [2011] employ a similar strategy. In [Wang et al., 2016], the authors first perform RANSAC based circle fitting of projected stem points, followed by RANSAC cylinder fitting in 3D and subsequent growing of the initial result. Olofsson et al. [2014] voxelize the point cloud and analyze the occupancy of different height layers to determine stem positions, followed by a RANSAC procedure for radius determination. A common feature of these studies is to utilize only points lying over 1 m over the ground, which eliminates most of the ground vegetation. In [Liang et al., 2012], the

stem points are found using classification based on features of the local covariance matrix. A similar pre-processing step is utilized to filter out ground points, shrubs and foliage. The authors then perform cylinder fitting with an M-estimator based on Tukey's biweight influence function. Overall, it appears that the detection and modeling of standing stems is a somewhat easier task compared to fallen tree mapping due to the availability of good prior knowledge of the cylinder's orientation (world Z axis), the ability to remove the influence of ground vegetation through DTM filtering, and the fact that neighboring tree stems are usually well separated by empty space.

1.2.2 3D cylinder detection

The task of detecting stems in TLS data can be regarded as a specialization of the more general problem of cylinder detection from dense point clouds. A number of methods have been developed for this purpose [Lari & Habib, 2014; Díaz-Vilariño et al., 2015]. These can be broadly categorized into several groups. First, direct or spatial-domain methods try to extract cylinders by region growing [Rabbani et al., 2006], starting from a seed point and successively adding new points to the candidate cylinder cluster by analyzing the current points' 3D neighborhood. Once the entire point cluster is obtained, the cylinder parameters (orientation, position etc.) can be calculated by fitting a cylinder to the selected points (e.g. using a least-squares procedure). A disadvantage of these methods is their dependence on the seed, and sensitivity to noise in the point cloud. Another broad category encompasses parameter space-based approaches. Here, instead of first finding the points supporting the cylinder hypothesis and then fitting the shape, the parameter space of the cylinder is directly explored. Cylinder hypotheses in the form of sets of concrete parameter values are generated and ranked, and the highest-ranking cylinder is returned. A prominent subclass of the parameter space category is the family of sample consensus (SAC) methods [Schnabel et al., 2007], where the hypotheses are generated based on minimal samples from the point cloud necessary to determine the parameters, and ranked using the number of inlier points or the distance between the inliers and the cylinder model. For cylinders, the number of parameters which must be determined ranges from 3 to 9, depending on the specific parameterization as well as parallelism constraints to a plane or line [Beder & Förstner, 2006]. For unconstrained cylinders in very dense point clouds, the number of samples which must be examined to ensure a reasonable probability of finding the optimal hypothesis may become prohibitively large. Accumulation-based methods form the second important subgroup of parameter-driven approaches. The hypotheses are no longer randomly generated, instead each input point casts a vote in the parameter space for one or more hypothesis it supports. The optimal hypothesis is found by locating the maximum in the accumulated space of votes. The Hough transform is perhaps the most well-known accumulation based method. Although it was originally designed for 2D raster data and circle detection, it has been since generalized to other shapes and data types. Notably, Rabbani & van den Heuvel [2005] presented a two-stage Hough transform for detecting cylinders from point clouds. The prohibitively large 5D parameter space of the cylinder was partitioned into two smaller subspaces: (i) a 2D space corresponding to the cylinder axis orientation, and (ii) a 3D space defining the axis position and the cylinder radius. By estimating the two groups parameters in sequence, the entire parameterization can be recovered. Although the Hough transform is very useful in practice, it also has some disadvantages. First, the computational complexity may be extremely high due to exhaustive generation of samples from different parameterizations of the target shape during the voting part. Also, the search for the optimal hypothesis is sensitive to the quantization (i.e. bin width) of the accumulation space.

1.2.3 Active and semi-supervised learning

Active learning

Due to the vast amounts of raw data available from a multitude of sensor types, the remote sensing community has shown an increasing interest in active learning methodologies as a means to optimize the use of the (usually limited) resources available for labeling data in various practical applications. Up to now, active learning has been successfully incorporated into classification workflows for data acquired using SAR [Babaee et al., 2015], multispectral [Tuia et al., 2009] and hyperspectral [Rajan et al., 2008] sensors. Several authors report methods tailored to specific characteristics of remote sensing data. In [Pasolli et al., 2014], spectral and spatial information is integrated in a sample selection framework based on SVMs. Persello et al. [2014] construct a cost-sensitive active learning framework, where the utility of a sample is calculated based on both its informative value and the cost of labeling it. Zhang et al. [2015] propose a multi-kernel version of the query-by-committee method, where each kernel is associated with a different data source (hyperspectral imagery, LiDAR data). Tuia et al. [2011] provide a comparative study of active learning methods applied to remote sensing imagery.

Semi-supervised learning

Methods for processing of remote sensing images have also benefited from the introduction of the semi-supervised paradigm. It has been used in the context of classification [Huo et al., 2015; Wang et al., 2014] as well as segmentation [Munoz-Mari et al., 2012] of multi- and hyperspectral imagery. Erkan et al. [2010] propose a semi-supervised logistic regression formulation based on the principle of generalized maximum entropy for classifying hyperspectral images. In [Vatsavai et al., 2005] a Gaussian mixture model with different weights for labeled and unlabeled data is developed for image classification, which is optimized using the Expectation-Minimization algorithm. Persello & Bruzzone [2014] perform a theoretical and experimental comparison of active and semi-supervised learning applied to the classification of remote sensing images.

Combining the two paradigms

Because of the success of both active and semi-supervised learning in practical applications, some effort has also been dedicated to combining these two learning paradigms and retaining the best features of both within a single methodology. Yu et al. [2009] propose a unifying framework based on maximizing entropy reduction over the data pool. The examples considered for inclusion in the training set are weighted according to how much they reduce the uncertainty of classification on the remaining data, where the uncertainty is measured by the class distribution entropy. Both of the learning paradigms are viewed as realizations of this framework, and they are applied sequentially to the initially unlabeled data. The method of Tur et al. [2005] proceeds in the spirit of self-training. A classifier is trained on the labeled examples and applied to the remaining part of the data, which yields putative labels. These machine-labeled examples are then introduced into the training set, and during training they are associated with a lower weight compared to the original human labeled objects to account for their lesser confidence. The next approach due to McCallum & Nigam [1998] integrates a query-by-committee active learning strategy with probabilistic labeling of the unlabeled samples through the Expectation-Maximization (EM) algorithm under a generative (naive Bayes) model. By nature of EM, the assigned labels are continuous expected values as opposed to discrete categories. Together with the true training labels, these expected values influence the parameters of the classifiers which form the active learning committee. Zhu et al. [2003] explore the use of Gaussian random fields for combining active and semi-supervised learning. The labeled and unlabeled instances together form a graph with edge

weights derived from a distance metric on pairs of nodes in the Euclidean feature space. Due to the harmonic property of Gaussian random fields, the continuous 'label' of each node is the weighted average of its neighbors' labels, which ensures that objects lying close in feature space also receive similar labels. During active learning, the utility of each sample is judged based on its reduction of the estimated expected classification error. Erkan [2010] defines an uncertainty sampling strategy with an entropy-regularized logistic regression model in the role of the classifier. Wan et al. [2015] report a iterative self-training variation where the pseudolabels obtained from the classifier trained on expert-labeled data are verified using an active learning procedure based on the margin sampling criterion.

1.2.4 Contextual classification with conditional random fields

Due to their attractive properties, CRFs have attracted the attention of researchers from the remote sensing community. Concerning raster images, Wegner et al. [2011] used a CRF binary with a regular neighborhood structure for enhanced building detection from InSAR data combined with optical imagery. Also, some authors propose to build graphical models on arbitrarily shaped superpixels, which results in irregular neighborhoods. Yang & Förstner [2011] define a multi-scale CRF with pairwise interactions on superpixels resulting from mean-shift segmentation for classifying man-made objects in optical images. In a similar setting of urban area classification, Volpi & Ferrari [2015] construct a multiple-ring neighborhood on the superpixels based on the distances between their centers. They then utilize a structured SVM to learn the binary CRF's interaction potential weights, yielding a prior on the localized pairwise class interactions. In the context of road network extraction, Wegner et al. [2015] depart from the purely binary model in favor of a higher-order CRF. They show that the higher-order potentials enforcing within-clique label consistency are suitable for detecting roads which appear as thin, elongated structures. Some effort has also been dedicated to transfer CRF-based methods to the 3D point cloud domain. This was done mainly with respect to urban area classification in ALS [Niemeyer et al., 2014] and MLS [Weinmann et al., 2015b] point clouds. Note that due to the characteristics of the data acquisition process, it is no longer possible to define a regular neighborhood as was the case with raster images. Instead, a k-nearest neighbor or fixed radius based neighborhood of a cylindrical or spherical shape is usually employed.

1.3 Objectives and contributions

The general goal of this thesis is to develop adaptive, data driven methods for the accurate detection and delineation of individual fallen and standing dead trees from ALS and TLS point clouds. The following research questions, which have not yet been addressed or fully answered in literature as per the current state of the art, are considered:

I Is it possible to automatically detect fallen trees from aerial and terrestrial laser scanning data with an accuracy which allows operational application as part of large-scale mapping of forested areas, replacing or complementing manual field inventories ?

II What are the necessary characteristics of forest plots, fallen stems and laser scanning point clouds which must be met to ensure successful reconstruction of lying dead trees ?

III How reliably can standing dead trees both with and without crowns be detected on a large scale based on ALS and aerial infrared imagery ? Is it sufficient to utilize 2D raster data, or is 3D information indispensable ?

IV What gain in terms of the classification accuracy vs. training set size tradeoff can be achieved by combining active and semi-supervised learning, compared to using each of these paradigms separately ?

Based on the analysis of techniques utilized for solving related pattern recognition problems as presented in the previous section, the research questions shall be answered through pursuing the following specific goals within this thesis. From a methodological perspective, they can be categorized into three groups.

1.3.1 Fallen trees

It appears that the task of detecting fallen trees from sparse ALS point clouds is quite challenging. On the one hand, the low point density near the ground makes it impossible to apply shape cues (e.g. cylindrical form of stems), while dense ground vegetation and closely packed adjacent stems limit the utility of general-purpose line detection methods. The quantitative results reported thus far indicate that this problem has not yet been completely solved. One of the shortcomings of the methods discussed above is that they attempt to model quite complex objects using either multiple user-defined thresholds, or inadequate features which fail to capture the full extent of the variability of the target objects' appearance. Another problem is the use of fixed-length templates, which further constrains the class of fallen tree shapes that can be detected. Motivated by many successful applications of machine learning methods to remote sensing problems in general, and contextual classification using CRFs in particular, an alternative approach is proposed where the target object appearance can be learned from either simulated or user-labeled examples. The method shall proceed in a bottom-up fashion. First, equal-length primitives (linear segments) in the point cloud will be detected, and their spatial relationships shall be modeled using a CRF which encodes a statistical prior on elongated, linear structures (like fallen stems). In the second step, the primitives chosen by the CRF will undergo a merging process, resulting in individual detected objects. Both the CRF and merging algorithm parameters shall be learned from training data. In summary, the two goals related to fallen tree detection in ALS point clouds can be stated as follows:

- Define a general framework for reconstructing linear structures in sparse point clouds, using a bottom-up approach where (i) equally sized primitives are detected and modeled using contextual classification, and (ii) positively labeled primitives are merged to form entire objects.
- Implement a concrete instance of this framework based on the characteristics of ALS data and the target class of fallen trees, including the design of features for classification, the object pair similarity function for merging and the means of learning them from training data

Regarding dense point clouds, there has been little prior work on the detection of fallen trees. On the other hand, under favorable conditions the point density is high enough to capture the distinctive cylindrical shape of the stems. Allowing the possibility of stem fragmentation, this thesis also proposes a two-tiered strategy where cylindrical shapes are first detected, and then merged into stems. Based on the reported prior work related to 3D cylinder detection via parameter space accumulation, a voting-based cylinder detection method is envisioned which uses nonparametric estimation to aggregate the votes in a continuous space and hence avoid the difficulty of manually defining the parameter quantization size. The goals associated with TLS are therefore:

- Introduce a 3D cylinder detection method based on accumulation in parameter space, which (i) does not require the user to specify the accumulation bin size and (ii) significantly reduces the number of samples generated during the parameter voting compared to a Hough transform based approach

- Incorporate this cylinder detection method into a full workflow for detecting fallen trees from dense point clouds, by defining a merging procedure which will form entire fallen trees from cylindrical primitives

1.3.2 Standing dead trees

Up to now, the methods for detecting standing dead trees from aerial imagery and ALS point clouds have developed mostly separately. In case of ALS point clouds, the reflection intensity proved to be an important discriminative feature, however the quantitative results remain inconclusive (71-92% detection accuracy). This could be associated with different types of structural damage (fire damage vs. bark beetle infestation) of the trees. On the other hand, the near infrared band of multispectral aerial imagery can reliably determine a deficiency in chlorophyll pigment content, which makes for a convenient detector of dead vegetation. However, the delineation of single trees from 2D imagery only is more difficult compared to 3D point clouds, because the vertical structure of the forest is not available. Also, many studies have shown that it is advantageous to combine LiDAR and spectral data for the general task of tree species classification and segmentation (e.g. Ke et al. [2010a]). Therefore, one of the goals is to introduce a dead tree detection and delineation method which combines ALS data and multispectral imagery. The next goal is to find a set of discriminative 3D features facilitating the detection of standing dead trunks which have already lost their crown and do not produce a visible footprint in aerial imagery. Finally, this thesis is also concerned with developing a specialized segmentation method for delineating dead tree crowns in aerial imagery only (without 3D information). The approaches reported so far are all based on the strategy of first segmenting the image into individual trees, without considering they are dead or living, and only performing the classification in the next step. In this work it is proposed to incorporate prior information about the shape and color of dead tree crowns into an active contour segmentation [Cremers et al., 2007] process, resulting in an accurate simultaneous detection and delineation procedure. The goals concerning standing dead trees can be summarized as:

- Define a method for the detection of individual standing dead trees with crowns using combined 3D (ALS point cloud) and 2D (multispectral aerial imagery) information

- Propose 3D features for detecting standing dead stems from ALS point clouds

- Construct statistical priors on the shape and appearance of dead tree crowns in multispectral aerial imagery through learning from labeled examples, and combine them with a level-set segmentation to obtain an individual dead crown delineation procedure

1.3.3 Active and semi-supervised learning

As noted in Section 1.2.3, several authors have demonstrated the utility of combining active and semi-supervised learning to minimize the labeling effort associated with learning classifiers from large data pools. This is especially important in the context of remote sensing data, which may cover vast areas of land containing millions of objects (e.g. trees). In particular, to ensure the practical ability of the methods developed in this thesis to inventories of large forested areas, the idea of combining the two paradigms on the grounds of entropy minimization is explored.

For that purpose, it is desirable to generalize the Shannon entropy regularized logistic classifier of Grandvalet & Bengio [2006] to handle the broader class of Renyi entropies. Next, the objective is to incorporate the Renyi entropy regularized classifier into the active learning framework of expected error reduction [Roy & McCallum, 2001], where the posterior class distribution entropy of unlabeled objects is regarded as an estimate of the expected classification error. To conclude, the goals related to this section are:

- Extending the semi-supervised regularized logistic regression classifier to support the more general class of Renyi entropies
- Developing an active learning framework based on expected error reduction and utilizing the entropy-regularized classifier, which can be regarded as a way of combining the active and semi-supervised learning paradigms

1.4 Structure of thesis

This thesis is structured as follows. In Chapter 2, the theoretical background is provided, which serves as a foundation for the development of specialized methods in later chapters. In particular, the mathematical and statistical underpinnings of machine learning techniques are reviewed, and an account of fundamental concepts such as parametric and non-parametric models, maximum-likelihood parameter estimation and cross-validation is given. Also, the Normalized Cut segmentation and Sample Consensus model selection procedures are described, which are used as building blocks for some algorithms developed in this thesis. The chapter concludes with introducing several existing classes of 3D shape descriptors, such as 3D Shape Contexts and Point Feature Histograms, later utilized to derive features for classification.

Chapter 3 deals with methodological contributions of this thesis which are formulated in a general fashion, independent of the particular application of dead tree detection. This includes (i) the framework for detecting linear structures in remote sensing data, based on modeling the spatial interactions of primitives through contextual classification with CRFs, followed by merging using a graph-cut procedure with a similarity function and stopping criterion learned from training data, (ii) the combined active and semi-supervised learning approach based on Renyi entropy regularization and expected error reduction, as well as (iii) the method for detecting cylinders from dense point cloud using voting in a continuous parameter space modeled by a kernel density estimator.

Chapter 4 describes the specific methodologies and workflows designed for the detection of fallen trees from both sparse (ALS) and dense (TLS) point clouds. In the former case, the linear structure detection framework is instantiated with concrete appearance models of fallen trees and a physical simulation is applied to generate training data for learning the graph cut segmentation parameters. The latter case makes use of the statistical voting framework for reconstructing cylindrical shapes, followed by a disambiguation step to ensure spatial consistency.

In Chapter 5 the emphasis is shifted to standing dead trees. Here, the methods are presented for (i) delineation of individual dead crowns from only multispectral images (without 3D data) using level-set segmentation with shape and intensity priors, (ii) detection of dead trees with crowns from ALS combined with aerial imagery based on photogrammetric projection of 3D points onto the image, and (iii) detection of dead trunks without crowns from ALS data by applying a specialized 3D shape descriptor.

Chapter 6 outlines all experiments conducted to determine the performance of the proposed methods, the used evaluation strategies and reference data. Detailed information regarding the

test plots from the Bavarian Forest National Park as well as data collection campaigns and utilized sensors (ALS, TLS, multispectral) is given. The experimental description is followed by the presentation and discussion of results (Chapter 7), grouped by the type of dead trees considered (fallen or standing) and input data type (ALS, TLS, aerial imagery). For each specific experiment, first the quantitative results are described and then a discussion and interpretation is conducted.

Finally, Chapter 8 states the conclusions which can be drawn from the obtained results, in the context of answering the research questions posed in Sec. 1.3 and explaining how the specific goals have been met. In particular, in response to Questions I and III the main conclusion is that methods for mapping both lying and standing dead trees have attained an accuracy level permitting operational deployment in forests. Furthermore, the chapter proposes certain interesting future research directions which could lead to enhancements of methods described in this thesis.

2 Basics

This thesis makes extensive use of machine learning techniques such as classification and segmentation. In this chapter, such concepts are developed within the formal framework of statistical learning. Appropriate notation is introduced which will be used throughout the remaining chapters. Furthermore, certain specific techniques and algorithms for segmentation, classification and model fitting are described which are of particular importance for the methods developed in this thesis. Also, an account is given of several state-of-the art 3D shape descriptors which are used as a basis for classifying points and primitives with respect to the geometric properties of their spatial neighborhoods. The specific algorithms introduced in Chapters 4 and 5 build upon the fundamentals presented here.

2.1 Overview of statistical learning

Intuitively, the goal of statistical learning can be described in the following general manner: given a set of sample data, infer the probability distribution which generated them [Wasserman, 2010]. More formally, consider the set of $d-$dimensional vectors $x_1, \ldots, x_n \in \mathcal{X}$. The vectors x_i can be viewed as realizations of a random vector $X = (X_1, \ldots, X_d)$, which is distributed according to an unknown probability distribution F. The task is to estimate F (or some statistical functional of F such as the variance or mean value) based on the observed samples $x_1, \ldots, x_n$. Traditionally, in statistical literature a distinction is made between *parametric* and *nonparametric* models. A statistical model (set of probability distributions) is called parametric if it can be fully described by a finite-dimensional parameter vector θ, whose cardinality is fixed and independent of the input sample size. Such a model can be concisely written as $\mathcal{F} = \{F_\theta | \theta \in \Theta\}$. Estimation methods based on parametric models make stronger assumptions about the underlying probability distribution compared to their nonparametric counterparts. For example, it may be assumed that the sample data was generated by a distribution from the normal family, i.e. $N(\mu, \sigma)$, parameterized by the mean and standard deviation: $\theta = (\mu, \sigma)$. In contrast, non-parametric methods do not make *global* assumptions about the shape of the probability distribution. Instead, they rely on exploiting the *local* structure of the neighborhoods of samples x_i to infer the shape of the distribution around a target point $x \in \mathcal{X}$. For this reason, nonparametric approaches are most useful when the data dimensionality is low (e.g. $d <= 4$) and sufficiently many input samples are available across the entire domain $\mathcal{X}$ [Hastie et al., 2001]. Perhaps the most prominent and well-studied representative of this class of methods is kernel density estimation, described in detail in Section 2.3.

2.1.1 Supervised learning

In statistical learning applications, oftentimes the observed data samples x_i are coupled with responses y_i, producing pairs of data (x_i, y_i). In this setting, termed the *supervised learning* scenario, the X_j are called *predictors, features* or *independent variables*, whereas Y is the *dependent* or *response variable*. Here, the goal is to learn the mapping $h : \mathcal{X} \mapsto \mathcal{Y}$. If Y is continuous, this learning task is called *regression*, and $h(x)$ is the conditional expectation of Y given the observed

variables: $h(x) = E(y|X = x)$. On the other hand, the case of a discrete Y is referred to as *classification*, and $h(x) = \arg\max_{y \in \mathcal{Y}} P(y|X = x)$ is the *Bayes classifier*. This intuitive rule states that the classifier should pick the category having the highest posterior conditional probability given the data. Using $h(x)$, it is possible to predict values of the response variable given previously unseen independent variable samples. As an example, in Section 2.2, the family of generalized linear models (GLM) is introduced, which encompasses a broad range for both regression (e.g. ordinary least squares linear regression) and classification (e.g. logistic regression). In case of supervised learning, it is relatively easy to derive a measure of a method's error, based on the discrepancy between the true and predicted values of the response variable. This measure, named the *loss function* in statistical decision theory, is usually denoted as $L(y, h(x))$. Its average over a set of samples (x_i, y_i) may be regarded as a quantity which estimates the predictive model's expected error. Popular loss functions include the *squared loss*, $(y - h(x))^2$, as well as *0-1 loss*, $[y \neq h(x)]^*$.

Discriminative vs. generative models

In statistical and machine learning literature, there is a distinction between two kinds of models for classification: *discriminative* and *generative*. The former attempts to learn only the posterior class distribution $P(Y|X = x)$ conditioned on the feature values, whereas the latter learns the joint probability distribution of features and labels $P(X, Y)$ (or equivalently, $P(X|Y = y)$ together with $P(Y)$). Generative models can be thought of as richer (and more complex) than their discriminative counterparts, because they model the distributions of the features inside each class, as opposed to only the class boundaries in case of discriminative models. Owing to this property, it is possible to obtain new labeled object samples from generative models; they provide a complete description of the data *generation* process, which makes them an ideal tool for data simulation. To use the joint probability $P(X, Y)$ for classification, Bayes' theorem is applied:

$$P(y|X = x) = \frac{P(x|Y = y)P(y)}{P(x)} = \frac{P(x, y)}{P(x)} \propto P(x, y) \tag{2.1}$$

From the above, we see that the joint probability is proportional to the posterior class label probability, because for a given dataset the data probability $P(x)$ is only a normalizing constant. Therefore, it suffices to pick the class label maximizing $P(x, y)$.

It might seem counterproductive to learn a more complex model than strictly necessary if the ultimate goal is classification. Indeed, discriminative classifiers often attain a lower asymptotic error compared to generative models. However, for small sample sizes the latter class of models might prove superior. In particular, Ng & Jordan [2002] have shown that generative classifiers may attain their (higher) asymptotic error rate using only $O(\log n)$ training examples compared to the discriminative version's $O(n)$.

2.1.2 Unsupervised learning

Another important and frequent learning scenario arises when no response variables are distinguished in the data generation model. This case, known as *unsupervised learning*, encompasses tasks such as finding similarities in data (clustering/segmentation), data simplification (e.g. by dimensionality reduction or discovering representative elements) as well as anomaly detection. From a statistical perspective, these problems are all related to estimation of random vector X's probability distribution. In a low-dimensional setting ($d \leq 3$), the density may be robustly estimated using non-parametric methods, however for high-dimensional problems it becomes impractical to gather sufficient examples as to obtain an accurate estimate [Hastie et al., 2001].

*The Iverson bracket $[P]$ has a value of 1 if the logical expression P is true and a value of 0 otherwise.

This can be attributed to the so-called *curse of dimensionality*: the number of training samples for a reliable estimation grows exponentially with the data dimensionality. Also, contrary to the case of supervised learning, there is no single natural measure of quality of the learned models. For these reasons, a multitude of methods have evolved separately for the aforementioned various problems, such as dimensionality reduction (e.g. principal component analysis, manifold learning) or clustering (e.g. k-means, Gaussian mixture models). The latter task has received a great deal of attention within the computer vision and remote sensing communities due to its immediate applicability to challenging practical problems such as semantic segmentation and perceptual grouping within images, as well as scene understanding. Clustering methods usually make use of a distance metric defined $D(\,\cdot\,,\,\cdot\,)$ on the space of input feature pairs: $D : \mathcal{X} \times \mathcal{X} \mapsto \mathcal{R}$. Equivalently, a pairwise object similarity function may be applied. Given the input data $(x_i)_{i=1..n}$ and the function D, the distance (or similarity) matrix $W = (D(x_i, x_j))_{i,j}$ can be constructed. Clustering algorithms examine W to reveal the underlying structure of the data. In Section 2.4, the Normalized Cut clustering procedure is described, which is a crucial element of the methods and workflows proposed in this thesis.

2.1.3 Active learning

Consider the supervised learning scenario with a classification task. Oftentimes in practical applications there are vast amounts of unlabeled data available, but the unit cost of obtaining class labels can be high and require the support of human experts. This can make it infeasible to acquire labels for all procurable objects. The task of *active learning* is to make the best use of the limited resources through interactively selecting the most informative objects for labeling by an external 'oracle', with the goal of training a classifier on this object subset that is representative of the entire dataset. There are several settings of active learning, e.g. stream based sampling or synthesis of new informative examples [Settles, 2009]. However, in this thesis the focus is on the pool-based scenario. Let U be the pool of available unlabeled examples, and let T denote a small initial training set ($|T| << |U|$) containing pairs (x, y) of objects and their class labels. The symbol $f_T(x)$ refers to the classifier model trained on set T, which yields the class label for object x. Usually, the classifier can produce some measure of confidence in its decision alongside the class label, such as a probability or degree of belief. Then, pool-based active learning proceeds iteratively by repeating these steps: (i) select the most informative example $x^* \in U$ based on the current model f_T and unlabeled pool U, (ii) ask the external oracle for a label y^* of element x^*, (iii) remove x^* from U and add to (x^*, y^*) to T, retrain the classifier on the augmented T. The iteration is continued until some convergence criterion is met, e.g. the maximum number of labeled examples. The various active learning approaches differ mostly by how they define the 'informativeness' of the unlabeled samples. For example, uncertainty sampling selects the object associated with the classifier's least confidence, whereas the query-by-committee method assigns the utility score of a sample based on the extent to which a 'committee' of independent classifiers disagrees about its label. Settles [2009] gives a review of state-of the art active learning methods.

2.1.4 Statistical decision theory

The regression function and Bayes classifier $h(x)$ presented in Section 2.1.1 seem intuitively correct, however they also possess a profound theoretical justification which can be derived on the grounds of statistical decision theory. Let $\mathcal{H}$ be the space of valid hypotheses. Starting with a nonnegative loss function $L(u, v)$ which assumes a value of zero iff $u = v$, the *risk* of a model

(hypothesis) $h(x)$ is defined as the expected loss with respect to the joint probability of X and Y[†]:

$$R(h) = \int L(h(x), y)P(x, y)dxdy = E_{X,Y}L(h(x), y) \tag{2.2}$$

Naturally, we are interested in finding the hypothesis $h^*(x)$ for which $R(h)$ is minimal. The solution depends on the choice of the loss function. For squared loss, $R(h)$ becomes:

$$R(h) = E_X E_{Y|X}((h(x) - Y)^2|X = x) \tag{2.3}$$

It can be proven that the function minimizing R in this case is precisely the conditional expectation, $h^*(x) = E(Y|X = x)$ [Hastie et al., 2001]. Similarly, taking the 0-1 loss, we obtain:

$$R(h) = E_X E_{Y|X}([h(x) \neq Y]|X = x) = E_X \sum_{y'=1}^{|\mathcal{Y}|} [h(x) \neq y']P(y'|X = x) \tag{2.4}$$

It is straightforward to show that the optimal hypothesis for 0-1 loss is the aforementioned Bayes rule for classification, $h^*(x) = \arg\max_{y \in \mathcal{Y}} P(y|X = x)$.

In practice, the *true* risk of a hypothesis cannot be computed, since the probability distribution $P(X, Y)$ is not known. Therefore, an approximation to Eq. 2.2 known as *empirical risk* is optimized instead. The empirical risk is defined as the average loss on the training set:

$$R_{emp}(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x_i), y_i) \tag{2.5}$$

Since the training set may not be representative of the entire domain $(\mathcal{X}, \mathcal{Y})$, overfitting may occur. To counteract this, oftentimes a *regularization* term $J(h)$ is added to the empirical loss. Regularization is meant to discourage too complex hypotheses which might fit the training data extremely well, but are not likely to generalize well to previously unseen examples. The general form of the augmented optimization objective is then:

$$\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} L(h(x_i), y_i) + \lambda J(h) \tag{2.6}$$

In the above, λ denotes a coefficient which defines the balance between the data fitting term and regularization. By plugging the specific functional form of h, L, and J into Eq. 2.6 a concrete optimization objective is obtained, which is then minimized to yield the optimal hypothesis h^*.

2.1.5 Maximum likelihood estimation

In this section, one of the most influential and frequently used methods for parameter estimation in probabilistic models is introduced. Consider once again the collection of training samples $x_1, \ldots, x_n$, with the additional assumption that the samples are independent and identically distributed (iid) according to a parametric probability model $P(x; \theta)$. We wish to estimate the parameter θ. Due to the iid assumption, the joint probability of observing the data is:

$$\mathcal{L}(\theta; X) = P(x_1, \ldots, x_n; \theta) = \prod_{i=1}^{n} P(x_i; \theta) \tag{2.7}$$

In the above, $\mathcal{L}(\theta; X)$ denotes the *likelihood function*, where the joint data probability is seen as a function of the parameter θ. The fundamental idea underlying maximum likelihood estimation

[†]For discrete Y, the integral is replaced with an appropriate sum

is to pick the parameter value $\hat{\theta}$ which maximizes the likelihood function $\mathcal{L}(\theta; X)$ for the given data. In practice, for numerical reasons it is often easier to work with the log-likelihood function $\ell(\theta; X) = \log \mathcal{L}(\theta; X)$, which averts the problem of multiplying many small numbers (potentially resulting in arithmetic underflow):

$$\ell(\theta; X) = \sum_{i=1}^{n} \log P(x_i; \theta) \tag{2.8}$$

Note that the method of maximum likelihood is not restricted to probability models on the $\mathcal{X}$ domain. It is also applicable to the supervised setting, for estimating parameters of joint or posterior probability distributions $P(X, Y), P(Y|X = x)$.

The maximum likelihood estimator (MLE) has a number of desirable theoretical properties which justify its widespread practical use. Under certain regularity conditions on the probability distribution P and the function $\mathcal{L}$, the following holds:

- The MLE is consistent: the MLE estimate $\hat{\theta}$ converges in probability to the true value θ, i.e. the probability $P(|\hat{\theta} - \theta| > \epsilon)$ tends to zero with increasing sample size n, for all $\epsilon > 0$.
- The MLE is asymptotically efficient: among all consistent estimators, the MLE has the smallest asymptotic variance. This results in the tightest confidence intervals for the estimated parameter.
- The MLE is approximately normally distributed, where the standard error can be derived from the Fisher information associated with $\hat{\theta}$. This means that an analytical expression exists for an (approximate) confidence interval of the estimated parameter θ.

The utility of the MLE can also be justified using the aforementioned statistical decision theory. Consider again the loss function L, this time in the context of measuring discrepancies between the estimated and true parameter value, e.g. $L(\theta, \hat{\theta})$. In analogy to the case of hypotheses, we can define the *maximum risk* of an estimator $\hat{\theta}(x)$ as:

$$\bar{R}(\hat{\theta}) = \sup_{\theta} \int_{x} L(\hat{\theta}(x), \theta) P(x|\theta) dx \tag{2.9}$$

Remarkably, for parametric models satisfying mild regularity conditions and of sufficiently low dimensionality, the MLE is an approximate *minimax rule*: it (approximately) minimizes the maximum risk over the space of all estimators. More details can be found e.g. in the book by Wasserman [2010].

2.1.6 Expectation-Maximization algorithm

Sometimes, the functional form of the log-likelihood (Eq. 2.8) may be particularly challenging in terms of maximization w.r.t. the parameter θ. Here, an alternative method is presented which can in certain cases simplify the optimization process. The *expectation-maximization* (EM) algorithm, due to Dempster et al. [1977], assumes that the observed data $\boldsymbol{x} = \{x_i\}, i = 1..n$ is only a part of the complete dataset $\boldsymbol{z} = \{z_i = (x_i, y_i)\}$, where the unknown values y_i are referred to as *latent* or *missing* data and cannot be directly measured or observed. The main idea underlying the EM approach is to make use of the case when optimizing the complete data log-likelihood, i.e. $\ell_c(\theta) = \sum_i \log P(x_i, y_i; \theta)$ is significantly easier compared to the observed log-likelihood. Since part of the complete data is missing, the complete data likelihood needs to be estimated based on the observed portion $\boldsymbol{x}$. This estimate could then be maximized w.r.t. θ. However, note that

for estimating the likelihood of $\boldsymbol{z}$, a fixed value of θ is required. Therefore, it is not possible to simultaneously estimate the likelihood and maximize it. This gives rise to an iterative scheme, where the following two steps are repeatedly executed in sequence:

E-step. The conditional expectation (function) of the complete data likelihood w.r.t. the missing data distribution is constructed using the current estimate $\hat{\theta}^i$ of the parameters:

$$\begin{aligned} Q(\theta; \hat{\theta}^i) &= E[\log P(\boldsymbol{x}, \boldsymbol{y}|\theta)|\boldsymbol{x}, \hat{\theta}^i] \\ &= \int P(\boldsymbol{y}|\boldsymbol{x}, \hat{\theta}^i) \log P(\boldsymbol{x}, \boldsymbol{y}|\theta) \end{aligned} \tag{2.10}$$

M-step. The function constructed in the previous step is optimized to yield the new parameter estimate $\hat{\theta}^{i+1}$:

$$\hat{\theta}^{i+1} = \arg\max_{\theta} Q(\theta; \hat{\theta}^i) \tag{2.11}$$

The iteration is repeated until convergence of $\hat{\theta}$. An important theoretical result which justifies the use of EM in practice is that the procedure is guaranteed to converge to a (local) maximum of the original observed data likelihood $P(\boldsymbol{x}|\theta)$. However, many such local maxima may exist and the final optimization result is dependent on the choice of the initial estimate $\hat{\theta}^0$.

2.1.7 Computing the generalization error

When constructing predictive models, it is crucial to obtain an estimate of the expected prediction quality on previously unseen data. In other words, we would like to know how well the model generalizes the information learned from the training sample. It is well known that the empirical risk (i.e. training error) $R_{emp}(h)$ is not a good estimate of the true risk $R(h)$ over the entire domain $\mathcal{X}$, in fact it is biased downward, because the same data are taken as the basis for estimating both the parameters and the error associated with them [Wasserman, 2010]. Therefore, other techniques must be applied to obtain a better estimate. One simple way of achieving that is through using an additional, external dataset (disjoint with the training set) for validation. However, in many practical scenarios labeled data is scarce or expensive to acquire. This has led to the development of alternative methods that do not require an explicit test set. A prime example is *k-fold cross-validation*, where the training data is randomly partitioned into k disjoint parts (folds). Let $h^{-k(i)}$ denote the model trained on all folds *except* the one which contains the training example i. Then, the cross-validated prediction error can be written as:

$$CV(h) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, h^{-k(i)}(x_i)) \tag{2.12}$$

Cross-validation is a powerful tool which can be used for determining the meta-parameters of a predictive model (e.g. the regularization coefficient λ in Eq. 2.6). It is applied extensively for that purpose within the methods described in this thesis.

2.2 Generalized Additive and Linear Models

Generalized additive models (GAM) are a class of predictive models which emerged as an attempt to generalize ordinary linear regression to other distributions of the dependent variable Y. Let $\mu(x) = E(Y|X = x)$ be the conditional mean of the response given the independent variables. In

the GAM setting, it is assumed that the conditional mean is related to a linear combination of smooth functions f_i of predictors X_i through the *link function* $g(\mu)$:

$$g[\mu(x)] = f_0 + \sum_{i=1}^{d} f_i(X_i) \tag{2.13}$$

If we constrain Y to follow a probability distribution from the exponential family, and utilize linear functions in the role of the f_i's (i.e. $f_i(X_i) = \theta_i X_i$), we obtain the class of generalized linear models (GLM):

$$g[\mu(x)] = \theta_0 + \theta x \tag{2.14}$$

While GLMs are parametric (with parameter vector θ), GAMs admit a wider range of models, also allowing nonparametric smoothing through the f_i's. Depending on the choice of the link function and the distribution of Y, different models appropriate for various assumptions about the data generation process may be derived. For example, assuming a Gaussian distribution on Y and using an identity link function $g(v) = v$, the original linear regression is specified.

2.2.1 Logistic regression

Consider a task of binary classification with $Y \in \{0, 1\}$. In this case, it is natural to associate a Bernoulli distribution with Y, where the success probability parameter p corresponds to the likelihood of the '1' label being assigned given the feature values x. For the Bernoulli distribution, we have that $E(Y|X = x) = 0 * (1 - p) + 1 * p = p$, so p is also the conditional mean of Y. This constitutes the first component $\mu(x)$ of the GLM. On the other hand, a link function g is required such that g^{-1} will map the range of the unbounded linear predictor θx onto the interval of valid probability values: $g^{-1} : \mathcal{R} \mapsto [0; 1]$. Many such functions exist. A common choice is the *logistic function*:

$$f_{log}(z) = g^{-1}(z) = \frac{1}{1 + e^{-z}} \Rightarrow g(\mu) = \ln \frac{\mu}{1 - \mu} \tag{2.15}$$

Putting all the components together, we obtain the classic logistic regression (LR) model:

$$\ln \frac{\mu(x)}{1 - \mu(x)} = \theta_0 + \theta x \tag{2.16}$$

Logistic regression therefore directly models the posterior probability of obtaining a positive class label as a function of a linear combination of the covariates. The use of LR for data classification is widespread due to its predictive power, a straightforward generalization to more than two classes, as well as a natural interpretation of the coefficients θ in terms of how they change the *odds ratio* of the positive/negative class $\mu(x)/(1 - \mu(x))$.

Kernelization

In Equation 2.16, the standard parametric form of LR is given. The size of the parameter vector θ is dependent on the number of dimensions associated with the feature space $\mathcal{X}$, but is fixed with respect to the number of training examples. However, a non-parametric version of LR also exists and can be constructed using a technique known as *kernelization*. To achieve this, the parameter vector θ is first represented by a linear combination of training examples: $\theta = \sum_{i=1}^{n} \alpha_i x_i$. Then, the log-odds for a new example x from Eq. 2.16 becomes:

$$\ln \frac{\mu(x)}{1 - \mu(x)} = \theta_0 + \sum_{i=1}^{n} \alpha_i x_i^T x \tag{2.17}$$

Examining the above equation, we observe that the training data x_i only interacts with the new data x through the dot product. The main idea behind kernelization is to replace the dot product in the original feature space $x_i^T x$ by a positive-definite binary *kernel function* $K(x_1, x_2)$, which measures the similarity between its two argument vectors. Substituting into Eq. 2.17, the kernelized form of LR is expressed as:

$$\ln \frac{\mu(x)}{1-\mu(x)} = \theta_0 + \sum_{i=1}^{n} \alpha_i K(x_i, x) \tag{2.18}$$

By applying the kernel function K, the dot product is carried out in a lifted (possibly infinite-dimensional) implicit feature space, in analogy to the kernel trick introduced in the context of support vector machines. The potential benefit is that a problem which is linearly nonseparable in the original feature space may become separable in the lifted feature space. Note that the new parameters of this model are the $\alpha = [\alpha_1, \ldots, \alpha_n]$ coefficients, which measure the influence of each training example. Their count is now no longer fixed, but rather related to the size of the training data, making the kernelized logistic regression a nonparametric technique.

2.2.2 Regularization

As the number of parameters (dimension of θ or number of coefficients α) becomes larger, the risk of overfitting the training data increases. Therefore, high-dimensional GLMs are often augmented with a regularization term in the spirit of Eq. 2.6. The regularization functional $J(h)$ then becomes $J(\theta)$, a function of the parameter vector. For generalized linear models, so-called *shrinkage* methods are commonly used. These methods impose a penalty on the magnitude of θ, giving the following specific form to J (cf. 2.6):

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} L(h(x_i; \theta), y_i) + \lambda \sum_{j=1}^{d} |\theta_j|^p \tag{2.19}$$

Two popular choices are $p = 2$ and $p = 1$. The former is known as *ridge* penalty and has the effect of proportionally shrinking all θ coefficients, whereas the latter is called the *lasso* and it translates the θ_i's by a fixed value (depending on λ), truncating at zero [Hastie et al., 2001]. The lasso penalty leads to an implicit predictor selection, as some elements of θ may be shrunk to zero, effectively eliminating the influence of their corresponding independent variables. A weighted sum of ridge and lasso penalties has also been proposed and is known as the *elastic net*, i.e. $J(\theta) = \beta||\theta||_1 + (1-\beta)||\theta||_2^2/2$, where $0 \leq \beta \leq 1$.

2.2.3 Fitting GLMs

Owing to the fact that generalized linear models have a sound statistical foundation, the estimation of their parameters is amenable to maximum-likelihood approaches. Indeed, the usual way of learning a GLM from training data is by formulating the joint data likelihood as a function of θ and maximizing the log-likelihood $\ell(\theta)$ (see Section 2.1.5). In general, there are no closed form solutions for the maximizer $\hat{\theta}$. The iterative Newton-Raphson method can be applied to optimize ℓ:

$$\theta^{k+1} = \theta^k - \left(\frac{\partial^2 \ell(\theta)}{\partial \theta \partial \theta^T} \right)^{-1} \frac{\partial \ell(\theta)}{\partial \theta} \tag{2.20}$$

The regularized versions of GLMs (Sec. 2.2.2) are usually trained by minimizing the sum of the negative log-likelihood and the appropriate penalty term weighted by λ, as in Eq. 2.19, where the loss L is defined as $L(h(x_i; \theta), y_i) = -\ln P(y_i | X = x_i; \theta)$. This augmented problem may become

non-differentiable for certain choices of p, which requires the use of specialized optimization procedures such as coordinate descent [Friedman et al., 2010] or the proximal gradient method [Boyd & Vandenberghe, 2004].

2.3 Kernel density estimation

Kernel density estimators (KDE) are a powerful and versatile tool for smoothly approximating an arbitrary probability density function (PDF) f in a nonparametric way. It is assumed that f is bounded and twice continuously differentiable with bounded derivatives, but no assumptions are made as to the shape of f. Instead, a KDE represents the local shape of f around a point x as a weighted sum of contributions from training examples in the neighborhood of x. Specifically, consider again the set of training vectors $x_i, 1 \leq i \leq n$. The KDE puts a mass of $1/n$ at each training point, distributed according to the weighting function $K(||x - x_i||)$ which is centered at x_i and diminishes with increasing distance between x and x_i. For one independent variable ($d = 1$), this can be expressed as:

$$\hat{f}_{kde}(x) = \frac{1}{nb} \sum_{i=1}^{n} K\left(\frac{x - x_i}{b}\right) \tag{2.21}$$

In the above, the weighting function K is referred to as the kernel. Although it shares the name with the kernel functions discussed in the context of nonparametric classification (Sec. 2.2.1), it should not be confused with the latter since it represents a somewhat different concept. Here, the name *kernel* indicates a nonnegative smooth function which integrates to one over its domain, has a zero mean and positive variance on $\mathcal{X}$. The constant b is the kernel *bandwidth*. It controls the amount of smoothing and its appropriate selection is critical to obtaining a high-quality estimate of the sought density f [Wasserman, 2010]. A natural generalization of KDE to $d > 1$ dimensions exists. In this case, the bandwidth parameter, now denoted B, becomes a $d \times d$ matrix. In multiple dimensions, new types of interactions between bandwidth elements occur, because the strength of smoothing may vary along different directions. The general form of the multivariate KDE is given by the following expression, where $K_B(x) = |B|^{-1/2} K(B^{-1/2}x)$:

$$\hat{f}_{kde}(x) = \frac{1}{n} \sum_{i=1}^{n} K_B(x - x_i) \tag{2.22}$$

A common choice for K is the multivariate Gaussian kernel $K_G(u) = (2\pi)^{-d/2} e^{-x^T x/2}$. Like in the univariate case, the choice of the kernel K is not as essential as the choice of B for achieving a quality approximation. Nevertheless, the Gaussian kernel has a convenient functional form and it shows the role of the bandwidth matrix more explicitly:

$$\hat{f}^G_{kde}(x) = \frac{(2\pi)^{-d/2} |B|^{-\frac{1}{2}}}{n} \sum_{i=1}^{n} e^{-\frac{(x-x_i)^T B^{-1} (x-x_i)}{2}} \tag{2.23}$$

In the above, the bandwidth B plays the role of the shared *covariance matrix* associated with the multivariate distributions centered at each of the training points x_i. Examining the expression $(x-x_i)^T B^{-1}(x-x_i)$, we recognize the Mahalanobis distance between x and x_i under a covariance matrix B. Therefore, B can be seen as defining the distance metric in the feature space, which in turn specifies the region of influence of the $1/n$ mass centered at each data point. Depending on the desired model complexity, B can be a full-rank matrix, or be restricted to a either a diagonal matrix or a single scalar, i.e. σI_d, where I_d is the d-dimensional identity matrix. In all cases, B must be symmetric and positive definite.

Kernel density estimation possesses certain attractive properties which have sparked interest from both a theoretical and practical standpoint. First, under mild regularity conditions, the estimate $\hat{f}_{kde}$ converges in distribution to the true f, i.e. as the sample size increases, $\hat{f}_{kde}(x)$ tends to $f(x)$ for all x where f is continuous. Moreover, the convergence rate of KDE is optimal among the class of nonparametric methods [Wahba, 1975]. The rate of convergence in terms of n and d is of the order of $n^{-p/(2p+d)}$, where p is the number of existing bounded derivatives of f [Stone, 1980]. This can be seen as another manifestation of the curse of dimensionality (see Sec. 2.1.2): for high-dimensionality problems, convergence becomes prohibitively slow, with an unrealistically large quantity of training examples n required. Therefore, in practice KDE is mostly useful for low-dimensional estimation tasks.

2.3.1 Bandwidth estimation

As mentioned previously, selecting an appropriate bandwidth is crucial for ensuring good performance of the KDE. Therefore, many approaches to kernel bandwidth estimation have been developed in statistical literature. Here, two approaches are reported which are often used in practice.

Cross-validation

First, consider the following error measure for the estimate $\hat{f}$ of f, based on the risk of the hypothesis $\hat{f}$ under a squared loss, integrated over the domain of f:

$$R(\hat{f}) = E_X[\int_{\mathcal{R}^d} (\hat{f}(x) - f(x))^2 dx] \tag{2.24}$$

This quantity is also known as the *mean integrated squared error* (MISE). It is desirable to find a hypothesis $\hat{f}^*$ which is a minimizer of MISE. By expanding the squared expression in the above integral, we can rewrite the risk as:

$$R(\hat{f}) = E_X[\int_{\mathcal{R}^d} \hat{f}^2(x)dx - 2\int_{\mathcal{R}^d} \hat{f}(x)f(x)dx + R(f)] \tag{2.25}$$

In the above, $R(f) = \int f^2(x)dx$ depends on the unknown density f, but is a constant with respect to the hypothesis $\hat{f}$, therefore it can be dropped. Of course, in practice the true f is unknown (since it is being estimated), so directly minimizing criterion 2.25 is intractable. The cross-validation approach proposes an empirical risk where the term dependent on f is approximated by the average cross-validated estimate $\hat{f}(x_i)$ over all data points:

$$UCV(B) = \int_{\mathcal{R}^d} \hat{f}^2(x;B)dx - \frac{2}{n}\sum_i \hat{f}^{-i}(x_i) \tag{2.26}$$

The term UCV indicates *unbiased cross validation*, so named because it is an unbiased estimate of MISE (shifted by $R(f)$). In Eq. 2.26, the dependency on the bandwidth B is made explicit. The expression $\hat{f}^{-i}(x_i)$ refers to leave-one-out cross-validation, reusing the notation introduced in Sec. 2.1.7. The UCV criterion may be written in a more explicit form involving the kernels K_H [Duong & Hazelton, 2005] ($\star$ indicates convolution):

$$UCV(B) = \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}(K_B \star K_B - 2K_B)(x_i - x_j) + n^{-1}R(K)|B|^{-1/2} \tag{2.27}$$

By minimizing UCV, we obtain the optimal cross-validation based bandwidth B_{UCV}. By Stone's theorem, it can be proven that as n tends to infinity, B_{UCV} converges in probability to B_{MISE}, i.e. the true minimizer of MISE [Wasserman, 2010]. However, the finite sample behavior might be unsatisfactory, in particular high sample variability may pose a problem [Wand & Jones, 1994].

Plug-in estimation

To alleviate the problems associated with cross-validation and oversmoothing-based bandwidth estimation techniques, Wand & Jones [1994] proposed an extension of the theoretically and practically appealing plug-in estimation approach to the multivariate case. Instead of minimizing MISE directly, they work with an approximation referred to as *asymptotic mean integrated squared error* (AMISE), given by:

$$AMISE(B) = \frac{1}{4}\mu_2(K)^2(\operatorname{vech} B)^T \Psi_4(\operatorname{vech} B) + n^{-1}R(K)|B|^{-1/2} \tag{2.28}$$

In Eq. 2.28, $\mu_2(K)$ and $R(K)$ depend only on the choice of kernel type (e.g. Gaussian), but not on the specific bandwidth B or the estimated true density f. The term $\operatorname{vech} B$ refers to the vector half operator, which stacks the elements of the lower triangle of B to produce a vector. The matrix Ψ_4 is the only part of AMISE which depends on the unknown density f. In fact, all elements of Ψ_4 can be expressed in terms of functionals ψ_r of integrated derivatives of f for a vector $r = (r_1, \ldots, r_d)$ of nonnegative integers:

$$\psi_r = \int_{\mathcal{R}^d} f^{(r)}(x) f(x) dx, \quad f^{(r)}(x) = \frac{\partial^{|r|}}{\partial^{r_1} x_1 \ldots \partial^{r_d} x_d} f(x) \tag{2.29}$$

To obtain the optimal bandwidth B^*, the AMISE criterion should be minimized. For general positive-definite B and $d > 2$ no closed form solution of Eq. 2.28 is available, therefore the minimization must be carried out using numerical methods. In particular, in [Wand & Jones, 1994] the Newton-Raphson algorithm is applied, which requires the calculation of up to second-order derivatives w.r.t. B. These derivatives in turn depend on the unknown functionals ψ_r. The core idea behind the plug-in approach for kernel bandwidth selection is related to how ψ_r is estimated from the training data. Wand and Jones consider the estimators of ψ_r of the form:

$$\hat{\psi}_r(A) = n^{-1} \sum_{i=1}^{n} \hat{f}^{(r)}(x_i; A) \tag{2.30}$$

Here, the estimated quantities depend on their own internal bandwidth matrices A_r. Therefore, an embedded problem of finding optimal bandwidths A_r must be solved. Assuming a simple scalar model for these sub-bandwidths, i.e. $A_r = a_r^2 I_d$, the authors showed that the a_r minimizing the local AMISE depend on f only through other $\psi_{r'}$ values. This gives rise to an iterative scheme where the estimates $\hat{\psi}_{r'}$ are successively plugged into the formulae for higher order $\hat{\psi}$. To end the recursion, at bottom level the estimation is made based on the assumption that the underlying probability distribution is normal. The estimates $\hat{\psi}_r$ can then be directly applied to calculate the derivatives of AMISE. The outer optimization loop is carried out until convergence. Plug-in bandwidth estimation was shown to be more stable and reliable in revealing the structure of the data compared to cross-validation [Wand & Jones, 1994], however it should be noted that the computational costs can be considerable.

2.4 Normalized Cut segmentation

The Normalized Cut (also called Ncut) algorithm is a generic method for clustering arbitrary objects $O = \{o_1, \ldots, o_n\}$ organized over a discrete graph structure $G = (V, E)$, where the vertices V correspond to the individual objects, while the edges E define the neighborhood topology. This approach was proposed by Shi & Malik [2000] and belongs to the category of spectral clustering methods. This class of methods makes use of the eigenvalues associated with the object similarity matrix (see Sec. 2.1.2) to construct a low-dimensional representation of the

input points, which better reveals the underlying structure of the data than the initial features. In this case, the similarity matrix W is formed according to the graph structure: for an edge $(u, v) \in E$, $W(u, v) = W(v, u)$ is set to $f_S(o_u, o_v)$, whereas for $(u, v) \notin E$ the corresponding entry in W is zero. The symbol $f_S(o_u, o_v)$ represents the pairwise object similarity function, which is non-negative and attains its maximum at $o_u = o_v$, applied to the objects $o_u, o_v \in O$. Given the similarity matrix W, the goal is partition the input objects (or equivalently graph vertices V) into k disjoint groups $A_1, \ldots, A_k$, such that the Normalized Cut criterion is minimized:

$$Ncut(A_1, A_2, \ldots, A_k) \equiv \sum_{i=1}^{k} \frac{cut(A_i, V \setminus A_i)}{assoc(A_i, V)} \tag{2.31}$$

In the above, $cut(A, B) \equiv \sum_{a \in A, b \in B} W(a, b)$ and $assoc(A, V) \equiv \sum_{a \in A, v \in V} W(a, v)$. The binary case ($k = 2$), i.e. the two-way Ncut and is frequently applied in practice:

$$Ncut(A, B) \equiv \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \tag{2.32}$$

An important advantage of this partition criterion (Eq. 2.31) is that maximizing the intra-cluster association of objects is equivalent to minimizing their inter-cluster disassociation, so both objectives are achieved simultaneously at the function's minimum and the resulting clusters tend to be balanced. Unfortunately, finding an exact minimizer of Eq. 2.31 is an NP-hard problem, and in practice the minimizer is approximated by solving a generalized eigenvalue system. This approximation is constructed as follows. Consider the diagonal matrix D with the ith entry on the diagonal defined as the sum of weights on edges containing node i: $D_{ii} = \sum_{k:(i,k) \in E} w(i, k)$. Also, define the indicator vector x of length $|V|$, where x_i assumes a value of 1 if node i belongs to part A of the partition, and -1 otherwise. Using this notation, the binary Ncut criterion may be written explicitly as a function of the indicator vector: $Ncut(x)$. Letting $b = \sum_{x_i > 0} D_{ii} / \sum_{x_i < 0} D_{ii}$, an alternative indicator vector y is introduced as $y = (1 + x) - b(1 - x)$. Consider the following optimization problem based on the Rayleigh coefficient $R(y)$:

$$\begin{aligned} &\min_y R(y) = \frac{y^T (D - W) y}{y^T D y} \\ &\text{subject to } y_i \in \{1, -b\}, \quad y^T D \mathbf{1} = 0 \end{aligned} \tag{2.33}$$

Using matrix algebra, Shi and Malik have shown that the minimum of Problem 2.33 coincides with the minimum of $Ncut(x)$ over the space of indicator vectors. To make the problem amenable to solution through techniques from linear algebra, the constraint on y admitting only discrete values is relaxed. Then, based on established linear algebraic results, the (real valued) solution y^* minimizing $R(y)$ can be retrieved as the eigenvector corresponding to the second smallest eigenvalue of the generalized eigenvalue system $(D - W)y = \lambda D y$. To obtain the actual assignment of objects to the parts A, B, the elements of y^* must be discretized. Many strategies are possible for performing this step. The authors propose to discretize the range $(\max_i y_i^* - \min_i y_i^*)$ and find a splitting point y_{thr} such that $Ncut(\{i : y_i^* \leq y_{thr}\}, \{i : y_i^* > y_{thr}\})$ is minimized.

Regarding the similarity function f_S, usually the original multiplicative exponential model proposed by Shi and Malik is applied:

$$f_S(o_i, o_j) \equiv \prod_{k=1}^{M} e^{-\frac{d_k(o_i, o_j)^2}{\sigma_k^2}} \tag{2.34}$$

The values $d_k(o_i, o_j)$ correspond to the M abstract differential features which quantify various aspects of the difference between objects o_i and o_j, e.g. spatial distance, color disparity etc. The

magnitude of each feature's influence is controlled by the coefficients σ_k. The two-way Normalized Cut algorithm is often implemented as a recursive bi-partitioning strategy: for an input set Z, an optimal sub-division of Z into disjoint sets $X, Y : X \cup Y = Z$ is found such that $Ncut(X, Y)$ is minimal, and then the procedure is re-run separately with X and Y as inputs until an external stopping criterion is met. Such an approach has the advantage that the number of clusters need not be specified a priori, but rather it results from the structure of the data and the characteristics of the stopping criterion. A common choice for the stopping criterion is a threshold $Ncut_{thr}$ on the maximum Ncut value (Eq. 2.32): if for any partition $Ncut(A, B) > Ncut_{thr}$, the parts are considered too similar and the splitting is aborted for that node subset.

2.5 Sample Consensus fitting

Sample consensus (SAC) methods are a family of approaches for robustly fitting parameterized mathematical models (i.e. functional descriptions) to sets of observed data points in the presence of outliers, introduced by Fischler & Bolles [1981]. Given a class of models $L(\theta)$ described by parameter $\theta \in \Theta$, and a set of input points $P = (p_1, \ldots, p_n)$, SAC attempts to find the parameter value θ^* which has the strongest support in P. The term 'strong support' is intentionally vague since it depends on the specific variation of the algorithm. Let $m(L)$ be the minimal number of points from P which uniquely determine θ, e.g. a line in 3D space is defined by two points, having $|\theta| = 6$. The template for a generic SAC algorithm is the following sequence of steps, executed iteratively: (i) select a random sample of $m(L)$ points from P, (ii) determine the value of θ only from the sample, and (iii) evaluate the cost of the fitted model $L(\theta)$ on the entire dataset P using an additive cost function $C(L(\theta)) = \sum_{p \in P} \rho(L(\theta), p)$. These steps are repeated for a prespecified number of iterations N_I, and the result of the algorithm is the model $L(\theta^*)$ which attains the minimum cost. Following ideas presented in [Schnabel et al., 2007], it is possible to relate the number of iterations to the desired probability P_{opt} of recovering the true optimal model present in the data. Assume that this true, undistorted model, denoted $L(\theta_{opt})$, is represented in the dataset P by n_{opt} points. Then, the probability of picking a random sample consisting of only inliers, e.g. the points belonging to $L(\theta_{opt})$, is expressed as:

$$P_L = \binom{n_{opt}}{m(L)} \Big/ \binom{|P|}{m(L)} \approx \left(\frac{n_{opt}}{|P|}\right)^{m(L)} \tag{2.35}$$

Consequently, the probability of picking a correct sample at least once in N_I independent retries, denoted $P_{succ}(N_I)$, can be obtained by considering the Bernoulli process with P_L as the probability of success. This leads to $P_{succ}(N_I) = 1 - (1 - P_L)^{N_I}$. Conversely, to find the minimal number of trials which ensures that the correct model is recovered with probability at least P_{succ}, we can solve the equation w.r.t. N_I:

$$N_I \geq \frac{\ln(1 - P_{succ})}{\ln(1 - P_L)} \tag{2.36}$$

Many specific pointwise cost models $\rho(L, p)$ have been developed in order to deal with different scenarios of model fitting. Usually, they are a function of the distance between the model L to the point p, denoted as e. The original cost function due to Fischler and Bolles, termed *RANSAC* (random sample consensus), assigns a zero cost to all points lying within a specified distance threshold d_{thr}, and a positive constant T otherwise:

$$\rho_{RANSAC}(e^2) = \begin{cases} 0 & \text{if } e^2 \leq d_{thr}^2 \\ T^2 & \text{else} \end{cases} \tag{2.37}$$

In effect, RANSAC selects the model with the largest number of inliers within the distance threshold. One drawback of this approach is that all models having the same inlier counts are

indistinguishable, even though their total point-to-model distance may vary significantly. To remedy this, Torr & Zisserman [2000] proposed the *M-estimator SAC* cost function, which also considers the inlier distances in the cost calculation:

$$\rho_{MSAC}(e^2) = \begin{cases} e^2 & \text{if } e^2 \leq d_{thr}^2 \\ T^2 & \text{else} \end{cases} \tag{2.38}$$

The SAC framework can also be applied in a probabilistic setting [Torr & Zisserman, 2000]. This requires defining probability models $P_{in}(e)$ and $P_{out}(e)$ on the point-to-model distances/errors e respectively for inlier and outlier points. Then, the probability of observing an error value e is given by the following mixture model, where γ is the mixing coefficient:

$$P(e) = \gamma P_{in}(e) + (1-\gamma)P_{out}(e) \tag{2.39}$$

Similarly, the log-likelihood of observing the errors e_i of all data points $p_i \in P$ can be expressed by:

$$\ell = \sum_{i=1}^{|P|} \ln[\gamma P_{in}(e_i) + (1-\gamma)P_{out}(e_i)] \tag{2.40}$$

The negative log-likelihood can then be used directly in the role of the cost function $C(L(\theta))$. The coefficient γ is unknown and must be estimated from the data, e.g. using the Expectation-Maximization procedure. In this case, a new estimate $\hat{\gamma}$ is formed by iteratively averaging the probability that point i is an inlier, given by $\gamma P_{in}(e_i)/(\gamma P_{in}(e_i) + (1-\gamma)P_{out}(e_i))$, over the data P and repeating until convergence:

$$\hat{\gamma}^{j+1} = \sum_{i=1}^{|P|} \frac{\hat{\gamma}^j P_{in}(e_i)}{\hat{\gamma}^j P_{in}(e_i) + (1-\hat{\gamma}^j)P_{out}(e_i)} \tag{2.41}$$

2.6 3D shape descriptors

This section introduces several well-known descriptors of local point neighborhoods in 3D point clouds, which are used in this thesis as a basis for point-level classification of surfaces. These 3D point descriptors originated in the computer vision and robotics communities with the purpose of providing a compact representation of the geometric properties associated with the neighborhood around the target point within the point cloud. To formalize the concept of neighborhood, define the 3D point cloud $P = (p_1, \ldots, p_n)$ and consider a target point $p_i \in P$. Let $d(\,\cdot\,,\,\cdot\,)$ be a metric on $\mathcal{R}^3 \times \mathcal{R}^3$. The *fixed-radius neighborhood* of p with radius r w.r.t. the metric d is defined as $N_r(p) = p' \in P : d(p, p') \leq r$. Similarly, the *k-nearest neighborhood* $N^k(p)$ of p w.r.t. d is the set of points $p' \in P$ such that their distance $d(p', p)$ does not exceed the distance to p's k-th nearest neighbor. The choice of the metric d determines the shape of the neighborhood. For Euclidean distance, the resulting shape is spherical, whereas if the Mahalanobis distance is used, the neighborhood becomes ellipsoidal. Other choices are also possible.

Many approaches to constructing point descriptors are based on the 3D structure tensor of $N(p)$ [Weinmann et al., 2015a], i.e. the covariance matrix of the x, y, z coordinates associated with points from $N(p)$ ($\bar{p}$ indicates the data centroid):

$$S(p) = \frac{1}{n} \sum_{i=1}^{n} (p_i - \bar{p})(p_i - \bar{p})^T \tag{2.42}$$

The eigenvalues and eigenvectors of S provide important information about the geometry of points inside $N(p)$. In particular, the eigenvector associated with the smallest eigenvalue is an estimate

of the surface normal at p assuming a locally planar model. Moreover, the relationships between the eigenvalues can be used to derive functions quantifying specific geometric properties. Letting $\lambda_1 \geq \lambda_2 \geq \lambda_3$ be the sequence of $S(p)$'s eigenvalues, the quotient $(\lambda_2 - \lambda_3)/\lambda_1$ measures the planarity of $N(p)$, whereas $(\lambda_1 - \lambda_2)/\lambda_1$ corresponds to its linearity. Similarly, the curvature may be defined as $\lambda_3 / \sum_i \lambda_i$. More frequently used shape functions derived from eigenvalues of S are available in [Weinmann et al., 2015a]. Together these features can be used to distinguish between the type of surface which point p is situated on.

Another class of shape descriptors makes direct use of the spatial distribution of points inside the neighborhood. Frome et al. [2004] proposed the *3D Shape Context* (3DSC), which is based on a spherical neighborhood around the point p. The sphere is oriented according to the estimated local surface normal at p, and its volume is subdivided into bins in the azimuth, elevation and radial dimensions. Each point inside the neighborhood contributes to its enclosing bin. A histogram can then be constructed based on the contributions in each bin. This histogram constitutes a high-dimensional feature set for classification or nearest neighbor search.

Finally, a number of approaches rely on the relationships between surface normals of adjacent points to describe the neighborhood's geometry. Specifically, *Point Feature Histograms*(PFH) introduced by Rusu et al. [2008] consider a spherical neighborhood $N_r(p)$ centered on the target point p. First, the surface normals for all points in $N_r(p)$ are estimated. Then, for each pair of normals n_i, n_j a Darboux coordinate frame is defined, which allows the calculation of the difference between n_i, n_j expressed in 3 angular coordinates. The angles obtained from all pairs are then binned in a 3-dimensional histogram, forming the PFH at p. To alleviate the computational burden and reduce the dimensionality of the obtained feature set, a variation known as *Fast Point Feature Histograms*(FPFH) was also proposed. In this case, not all pairs of normals in $N_r(p)$ are used, and also the histogram of angular coordinates is decorrelated, i.e. each angular coordinate's histogram is constructed separately, instead of the 3-dimensional histogram which encoded the co-occurrence of all 3 angles. This decreases the computational time and vastly reduces the number of feature bins.

3 Extended methods for segmentation and learning

In this chapter, three general frameworks are presented which constitute the core of the methodological contributions introduced by this thesis. Although the ultimate application is closely tied to the goals of fallen and standing dead tree reconstruction, the methods described here are formulated in a modular fashion at a level of abstraction which hopefully can permit their use in other, unrelated domains.

First, the problem of reconstructing linear structures from remote sensing images (e.g. raster or point cloud, 2D or 3D) is considered. The goal is to handle complex scenarios where elongated objects of varying lengths are randomly scattered throughout the scene in close proximity, possibly stacked upon each other and highly overlapping. We are interested in delineating individual objects. The proposed approach is based on the idea of simplifying this complex task by first finding equally-sized object parts (primitives), and then combining related parts into individual objects. The algorithm learns from training data how to optimally detect parts which together form probable linear structures via contextual classification, and how to partition them into disjoint clusters based on their appearance within the scene.

In the second part of this chapter, a method is developed for the combination of the active and semi-supervised learning paradigms in the context of classification tasks. These two paradigms are unified on the grounds of posterior probability entropy minimization over the unlabeled object pool. This is based on the observation that there exist both active and semi-supervised approaches which minimize this entropy, and therefore their combined use could lead to a synergy effect resulting in faster convergence of the learning process.

The final section focuses on the task of detecting cylinders in dense 3D point clouds (e.g. obtained from terrestrial laser scanning or photogrammetric dense matching). Here, ideas related to the Hough transform for accumulation-based shape detection are expanded upon and cast in a nonparametric estimation setting with a continuous voting space. This allows to eliminate need for manually selecting the accumulation bin size, which is one of the deficiencies of the baseline approach.

3.1 Segmentation of complex linear structures through decomposition into primitives

Let I be an abstract image, e.g. a 2D raster grid or a 3D point cloud. The goal is to segment individual linear structures within I, which represent the objects of interest in the specific application. This could be fallen trees in sparse point clouds as investigated within this thesis, but also groupings of pipelines or road networks in high-resolution satellite imagery. Instead of trying to immediately detect entire objects, the proposed framework takes a bottom-up approach and

tries to find equally sized short *segments* which are then merged to form individual structures. This is in accordance with experiences related to 2D and 3D shape classification, where it was found that local-based shape descriptors are more robust to object appearance variability and presence of clutter within the scene compared to their global counterparts [Knopp et al., 2010]. The framework consists of four parts: (i) segment generation and unary classification, (ii) contextual classification using conditional random field, (iii) merging via Normalized Cut algorithm and (iv) scene ranking (see Figure 3.1). A distinguishing feature of this processing pipeline is that most of its elements are learned automatically from user-labeled or simulated data, as discussed further in this paragraph. The initial step has the purpose of creating an overcomplete set of

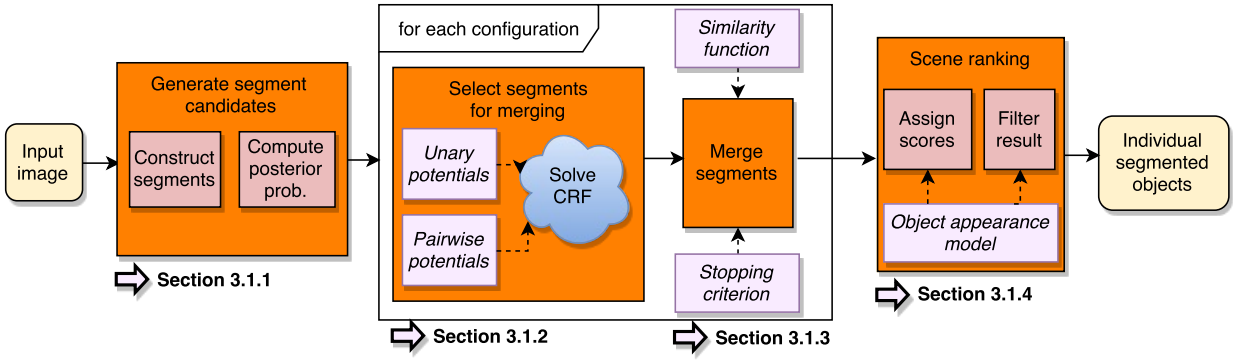


Figure 3.1: Overview of processing pipeline for detecting linear structures.

possibly overlapping segments based on the image I, and providing a soft classification for each candidate which can be though of as the unary probability that it is indeed a part of a target object. In the next step, the segment candidates undergo a crisp classification procedure based not only on their unary characteristics, but also on their spatial interactions with neighboring candidates. The goal of this contextual classification is to select a subset of segment candidates which together form probable linear structures, by explicitly modeling the pairwise relationships between segments such as collinearity or overlap. These pairwise models are learned from training data and may be regarded as a prior term in a probabilistic classification scheme encoded by the CRF which they form together with the unary probability from step (i). The CRF is parameterized by a set of coefficients which define the balance between the unary and pairwise terms. An important aspect of step (ii) is that it not only finds likely linear structures, but also reduces the potentially vast number of segment candidates obtained from the previous step to a more manageable quantity from a computational standpoint. The chosen segments then enter step (iii), where the unorganized set of primitives is partitioned into clusters using the Normalized Cut segmentation algorithm (see Section 2.4). Departing from the usual scenario of how Ncut is applied, in this framework both the segmentation stopping criterion and the similarity function are treated as binary classifiers and as such are also trained from labeled examples. The clusters obtained from the merging step represent the individual detected objects. Executing steps (ii) and (iii) corresponds to running the processing pipeline for a single parametrization. However, usually the optimal set of balance coefficients is unknown for a previously unseen dataset. Therefore, steps (ii)-(iii) are executed in sequence for a variety of parameterizations, and each time the resulting set of objects is recorded. Finally, step (iv) aims at ranking the segmented scenes based on the saved detection results. The result of the entire processing pipeline is the best-ranked set of objects, filtered by an object appearance model (also learned from data) to remove spurious elements. The rest of Section 3.1 is dedicated to explaining these four steps in detail.

3.1.1 Segment generation

Since the nature of the abstract input image I is not specified, the task of generating segment candidates is highly dependent on the concrete application. Therefore, in this framework, candidate generation is viewed as an external procedure which yields a set of segments $S = \{s_1, \ldots, s_N\}$ of equal length l_{seg}, given by their endpoints ($s_i = (p_{i,s}, p_{i,e})$). The user of this framework needs to provide a concrete implementation of this procedure. Moreover, the unary class posterior model $P_u(y|x)$ encodes the probability of a segment with appearance features x within image I being part of a true target object ($y = 1$) versus being irrelevant ($y = 0$). The aggregate unary posterior may consist of sub-potentials which correspond to different aspects of the segment's appearance:

$$P_u(y|x) = \frac{1}{Z_u(x)} \prod_k P_{u,k}(y|x)^{\gamma(k)} \tag{3.1}$$

In the above, $Z_u(x)$ is a normalizing constant, and the $\gamma(k) > 0$ exponents control the strength of potential k's influence. As $\gamma(k)$ tends to zero, $P_{u,k}(y|x)^{\gamma(k)}$ tends to the uniform distribution and hence its influence diminishes because it assigns nearly the same probability to both labels. Conversely, when $\gamma(k)$ tends to infinity, the entire probability mass becomes concentrated at one label. This model is additive in the log-potentials, where the $\gamma(k)$ are the linear combination coefficients (C is a constant):

$$\ln P_u(y|x) = \sum_k \gamma(k) \ln P_{u,k}(y|x) + C \tag{3.2}$$

It is assumed that the probability models $P_{u,k}$ are given, whereas the coefficients $\gamma(k)$ will become part of the CRF's parametrization and as such need not be specified in advance.

3.1.2 Contextual classification

The proposed method of selecting optimal segments for merging is based on the hypothesis that by considering how well the chosen segments fit together, i.e. by including classification *context* into the process, a set of primitives can be obtained which is easier to cluster and hence results in higher-quality segmentations. The rest of this section is dedicated to explaining the details of how the selection problem may be posed as a CRF energy minimization, the components of the energy as well as its efficient optimization.

Conditional Random Fields

Consider a set of N objects $S = (X, Y)$ given by the independent variables (features) X and a categorical dependent variable (class label) Y. Conditional Random Fields [Lafferty et al., 2001] are a type of undirected probabilistic graphical model which aims at describing the conditional probability distribution $P(Y|X)$ defined over a discrete structure (lattice or graph) capturing the neighborhood relation $\sim$ on the elements of S. Let $G = (V, E)$ be a graph with vertices V corresponding to objects from S and edges E induced by the relation $\sim$: $E = \{(v_i, v_j) : s_i \sim s_j\}$. We can then write the conditional label probability as:

$$P(Y|X) = \frac{1}{Z(X)} \exp[-\sum_{c \in \mathcal{C}} \psi_c(y_c|x)] \tag{3.3}$$

The probability can therefore be written as a product of factors $\exp \psi_c$ over the cliques c of G. The functions ψ_c are clique potentials, which assign a non-negative value to every possible labeling y_c of clique members $v \in V \cap c$. The partition function $Z(X)$ ensures normalization over the entire domain of labelings. Note that since the probability distribution is conditioned on all independent

variables X, each clique potential has access to *all* values x, not just x_c. The CRF can also be expressed in terms of energy:

$$E(Y) = -\log P(Y|X) = \sum_{c \in \mathcal{C}} \psi_c(y_c|x) - \log Z(X) \tag{3.4}$$

In contrast to standard classification methods where each object is classified independently (see Section 2.1.1), all object labels in a CRF are calculated jointly by maximizing the a posteriori probability (or equivalently, minimizing the energy). The partition function Z is a constant with respect to the labeling Y, so it can be omitted. In this work, pairwise CRFs are utilized, where the cliques have at most two members and the labels are binary. The energy then simplifies to:

$$E(Y) = \sum_i \psi_i(y_i|x) + \sum_{i \sim j} \psi_{ij}(y_i, y_j|x) \tag{3.5}$$

Constrained energy formulation

Here, the energy corresponding to the segment selection problem is formulated. Let S be the set of all N generated candidate segments, and $Y \in \{0,1\}^N$ their associated labels. The basic energy of a labeling Y is composed of k unary terms obtained from the unary probability models $P_{u,k}$ (Eq. 3.2) and one or more binary terms which describe pairwise interactions:

$$E_0(Y) = \sum_i \sum_k \gamma(k) E_{u,k}(y_i) + \sum_{i \sim j} \sum_l \beta(l) E_{b,l}(y_i, y_j) \tag{3.6}$$

The symbol $\sim$ denotes an abstract, application-dependent neighborhood relation between segments. The coefficients β, γ determine the balance between energy terms. As in Section 3.1.1, a labeling convention is assumed where values of '1' and '0' correspond respectively to choosing and omitting a segment. The unary and pairwise energy terms are defined as follows based on their respective probabilities:

$$E_{u,k}(y_i) \equiv \begin{cases} -\log P_{u,k}(1|x_i) & \text{if } y_i = 1 \\ 0 & \text{if } y_i = 0 \end{cases} \tag{3.7}$$

$$E_{b,l}(y_i, y_j) \equiv \begin{cases} -\log \phi_{b,l}(x_i, x_j) & \text{if } y_i = y_j = 1 \\ 0 & \text{else} \end{cases} \tag{3.8}$$

Here, $\phi_{b,l}(i,j)$ refers to an abstract pairwise potential normalized on $[0;1]$ which encodes a 'degree of belief' that segments i, j should be chosen (obtain a '1' label) together. The CRF formulation admits multiple pairwise potentials, which are once again application-specific and may quantify different aspects of segment similarity. However, a generic collinearity potential can be defined based solely on the spatial locations of the segments. This generic potential ϕ_{lin} is explained in detail below, at the end of this section. Note that all the energy terms defined so far are nonnegative. Indeed, positive energy may only arise when all labels within a clique are '1', or in other words, only interactions between selected segments contribute energy, while omitted segments are ignored. Since the goal is to minimize energy, the system would achieve its global minimum at a solution consisting only of '0' values: assuming that all $P_{b,l} < 1$, it is never beneficial to select any segment. Clearly, an incentive is needed to promote solutions with a positive number of selected items. A simple way to achieve this is through augmenting the minimization problem with a constraint on the minimal number N_0 of selected segments:

$$\begin{aligned} &\text{minimize } E_0(Y) \\ &\text{subject to } \sum_i y_i \geq N_0 \end{aligned} \tag{3.9}$$

Because all potentials are positive, the inequality constraint will always be satisfied with strict equality at optimum (otherwise the energy could be decreased by flipping any '1' label to '0'). This formulation allows the random field to find a subset (with size N_0) of segments that match each other well, while not committing to labels for the remaining objects. This leads to reducing the number of primitives which enter the next stage of processing. No energy is assigned to differently labeled object pairs because of the semantics of the labels and their subjective nature. The '1' label indicates a true segment which matches the other currently selected '1' segments, but the '0' class encompasses both truly irrelevant objects and good segments which were not selected due to the constraint N_0. Consequently, a positive energy term for neighboring objects with different labels would penalize the selection of segments within dense regions of the scene, because many '0' labeled neighbors would contribute to the pairwise energy. This is contrary to the goal of finding representative segments uniformly along the linear structure. Due to the inhomogeneity of the '0' class, the pairwise term for '0' labeled pairs is also omitted.

Several methods for solving constrained pseudoboolean optimization problems have been proposed in the computer vision community [Woodford et al., 2009]. In particular, energy optimization with equality constraints on assigned label counts have been explored for grid-based and planar graphs [Lim et al., 2010]. Here, an algorithm based on dual space search is applied, similar to the method of [Lim et al., 2011]. First, the Lagrangian dual of (3.9) is formed:

$$\begin{aligned} L(Y,\lambda) &= E_0(Y) - \lambda(\sum_i y_i - N_0) \\ g(\lambda) &= \min_Y L(Y,\lambda) \end{aligned} \tag{3.10}$$

The characteristic set χ_g of g is defined as the set of all labelings Y' such that Y' minimizes $L(Y,\lambda)$ for some fixed λ, i.e. $\chi_g = \{Y' | \exists_\lambda L(Y',\lambda) = g(\lambda)\}$. The following lemma connects the minimizers of (3.9) and (3.10) [Lim et al., 2011]:

Lemma 1. *Let $Y' \in \chi_g$ and $\sum_i Y'_i = b$. Then $E_0(Y') = \min_{Y\in\{0,1\}^n} E_0(Y)$ subject to $\sum_i Y_i = b$.*

This means that by solving the dual problem for a certain λ value, we also obtain the associated solution of the primal problem (3.9) for the corresponding constraint value $N_0(\lambda)$. To minimize $L(Y,\lambda)$, the unary potentials of the original problem are transformed by adding the dual constraint term, which gives rise to the final formulation of the proposed implicitly constrained CRF (icCRF) energy [Polewski et al., 2017a]:

$$E_{ic}(Y) = \sum_i -\lambda y_i + \sum_k \gamma(k) E_{u,k}(y_i) + \sum_{i\sim j}\sum_l \beta(l) E_{b,l}(y_i, y_j) \tag{3.11}$$

This yields an unconstrained pseudoboolean energy minimization problem, for which a number of state-of-the-art methods is available [Szeliski et al., 2006]. Note that is an NP-hard problem for general potentials [Kolmogorov & Rother, 2007], and therefore only approximate solutions can be obtained. The Lagrange multiplier λ can be interpreted as the unit payoff for selecting one segment. As λ increases, it becomes worthwhile for the network to select more candidates. In practice, the optimal number of segments N_0 which should be chosen to produce the best result is not known in advance, therefore a trajectory of solutions is generated for an increasing sequence of λ values, employing a warm start strategy. The initial constraint coefficient value $\lambda_0 = \min_i[-\log P_{app}(i) - \alpha \log P_{ctr}(i)] + \epsilon$ ensures that at least 1 segment is selected. The subsequent coefficient values follow an exponential scheme: $\lambda_{i+1} = F\lambda_i$, where $F > 1$. As λ increases, more candidates are selected. The process is terminated when the number of selected segments reaches a threshold ratio k_{thr} of the initial N candidates. For minimizing each instance of E_{ic} for a fixed

λ, the extended roof duality approach of Rother et al. [2007] is utilized, which builds upon results due to Hammer et al. [1984]. This method is based on two steps. First, the set of variables whose labels are fixed in all optimal solutions are found by examining the linear programming relaxation of the energy. After this step, some variables may remain unlabeled. In the second step, an improvement heuristic is iteratively applied to a starting configuration of variables. This heuristic guarantees to not increase the energy, and it may find a lower energy solution. The improvement procedure is carried out for a predefined number of iterations. The local search-like nature of this method is well suited to the target scenario, where the λ parameter is increased slowly, allowing the use of the previous λ value's optimization result as the starting point for the new search (warm start).

Pairwise collinearity potential

The role of the generic collinearity potential ϕ_{lin} is to estimate how well two segments s_i, s_j match each other, in the sense of forming part of the same linear structure, based on their spatial locations. Specifically, two differential features are considered: the angular deviation in headings d_{dir} as well as the average distance between the segments d_{seg}. The former is calculated using the endpoint positions as $cos^{-1}[(p_{i,e} - p_{i,s}) \cdot (p_{j,e} - p_{j,s})/l^2_{seg}]$. The latter is obtained by projecting equally spaced points from one segment onto the other, computing the projection distance and averaging the results (Fig. 3.2). A joint probability distribution is defined over (d_{dir}, d_{seg}) conditioned on the two segments being true object parts. This can be seen as a generative model of the angle deviation and average distance for actual linear structures, whereas no interactions with outliers are considered. The bivariate kernel density estimator (Section 2.3) is proposed as a concrete modeling tool for this probability, due to its flexibility and expressive power in approximating arbitrary distributions. Therefore, assuming a notation of $d^{ij} = (d_{dir}(s_i, s_j), d_{seg}(s_i, s_j))$, the probability of observing the differential features d^{ij} for two parts s_i, s_j of the same object is expressed by the familiar KDE form:

$$\phi_{lin}(s_i, s_j) \equiv P_{kde}(d^{ij}) = \frac{1}{n_t} \sum_k |B|^{-1/2} K(B^{-1/2}(d^{ij} - d_k)) \tag{3.12}$$

In the above, d_k represents any of n_t training examples supplied externally based on the concrete application, while K and B represent respectively a kernel function and its positive definite bandwidth as described in Section 2.3. The techniques presented in Section 2.3.1 can be used to estimate the optimal B from training data, given that the dimensionality of the model is low (two variables).

3.1.3 Merging

In this step, the segments chosen by the contextual classification procedure are merged to form individual objects. This is done based on the Normalized Cut algorithm, characterized in detail in Section 2.4. The original Normalized Cut approach does not provide a method of specifying the feature weights σ_i, and proposes a generic stopping criterion based on thresholding the Ncut criterion. In practice, the weights and the Ncut threshold are often determined empirically, via sensitivity analysis or simply trial-and-error. For some applications where the resolution and quality of the input data is high, and the objects within the scene are well separated based on their appearance, such approaches may provide sufficiently good results. However, for lower-quality data with potential ambiguity, manually finding appropriate weights and thresholds may prove challenging. This section describes a more systematic strategy for automatically learning both the similarity function and the stopping criterion of Ncut using training data, by viewing them from a classification perspective.

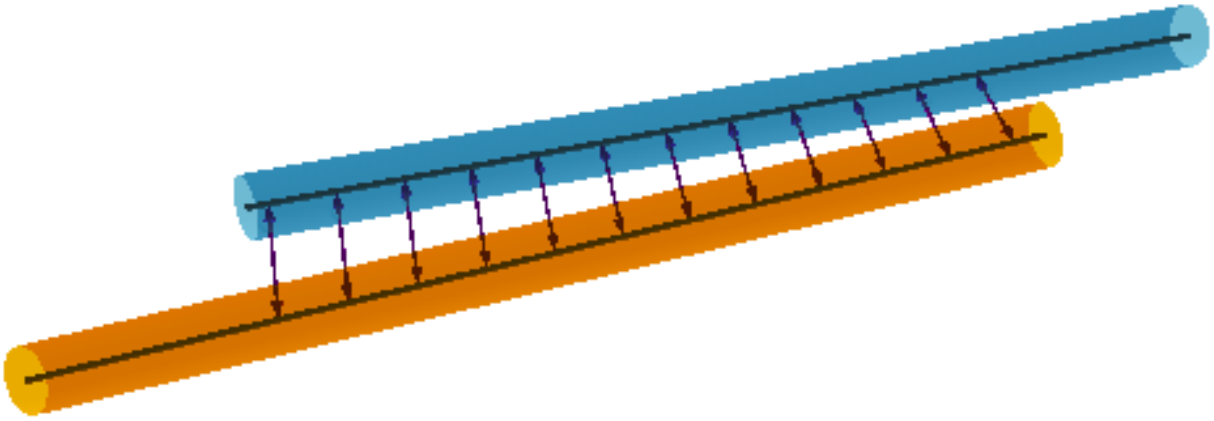


Figure 3.2: Projecting points lying on one segment onto the other. The average segment distance is calculated as the mean projection distance over the discrete points.

Differential features for merging

The Ncut similarity function (Sec. 2.4;Eq. 2.34) is based on differential features which quantify the different aspects of distance between a pair of segments, for all neighboring pairs under the abstract neighborhood relation $\sim$. In general, these features are also application and data type dependent. However, similar to the case of the pairwise potentials of the contextual classification CRF (Sec. 3.1.2), some generic features may be defined based on the spatial distribution of segments. In the following, let s_1, s_2 be the target pair of segments for feature calculation.

Starting point and direction.

This group of features extends the distance in heading d_{dir} defined in Sec. 3.1.2. Let the unit direction vector of segment s_k be denoted by $v_k = p_{k,e} - p_{k,s}/||p_{k,e} - p_{k,s}||_2$. Then, alongside the angular heading deviation d_{dir}, the vector difference in heading can be defined as $\Delta v_{i,j} = v_i - v_j$. Invariance with respect to the choice of the start and end points is ensured by picking the direction vector v_i sign which minimizes $||\Delta v_{i,j}||$, i.e. $v_i - v_j$ or $-v_i - v_j$. The final feature in this group is given by the Euclidean distance between the starting points of the direction-aligned segments. It is defined as $min(||q_i - q_j||_2, ||r_i - r_j||_2)$, where q_i and r_i are respectively the start and end points of segment s_i according to the direction vector v_i.

Distance along axis.

The features described here aim at providing a detailed description of the spatial relationship between s_1 and s_2. The idea of projecting points along the segment's axis, which was the basis for d_{seg} (Sec. 3.1.2), is reused for this purpose. The asymmetric distance profile $P_D(s_1, s_2)$ relative to s_1 is defined as a sequence of measurements of the distance between the segments along equally spaced points on s_1 (see Fig.3.2). The resulting descriptor is a concatenation of $P_D(s_1, s_2)$ and $P_D(s_2, s_1)$. Choosing the first profile in the concatenation to be associated with the segment possessing the starting point with the minimal x, y and z coordinate (tie breaking in this order) affords symmetry to the aggregate descriptor.

Learning the similarity function

The specification of the individual feature weights σ_i is a prerequisite for calculating object similarities (Eq. 2.34). Oftentimes a trial-and-error approach is used. Alternatively, an exhaustive search may be carried out on a discretized grid of weight values, where each weight configuration

is evaluated against an appropriate segmentation quality criterion. The computational burden of such a grid search can be immense, because every weight combination requires a complete re-calculation of the clustering, while the number of combinations grows exponentially with the number of features (or feature groups sharing the same weight). Therefore, this method may become infeasible for large data sets. Moreover, it is not obvious how to select the appropriate ranges for each weight as well as a grid width which will result in finding a reasonably performing configuration. Clearly, the weight definition process could benefit from a more robust method. This thesis proposes an alternative approach where the similarity is learned from external training data. Such an approach has been attempted before in the context of the k-way Normalized Cut [Bach & Jordan, 2003]. There, the authors propose a new formulation of the optimization objective which makes it possible to learn the similarity matrix from a known reference segmentation. Their approach relies on approximately optimizing the spectral clustering cost function directly with respect to the similarity matrix, which can lead to difficult eigenvalue problems. In this work, the task of directly learning the similarity function is approximated with a simpler one of training a probabilistic classifier that labels a pair of segments as either belonging to the same or different objects. This is based on the idea that classifiers which achieve a high classification accuracy in this setting will also perform well in the role of similarity functions for Normalized Cut. In the remainder of this section, it is shown that setting the weights σ_i is equivalent to implicitly specifying a specialized Generalized Linear Model (Sec. 2.2) classifier. Also, an efficient method of optimizing the model using the Newton-Raphson algorithm is proposed.

Similarity function as a GLM

Shi & Malik [2000] note that the similarity function for two elements should reflect the likelihood that they both belong to the same cluster. From such a perspective, the problem of defining a similarity function gives rise to a regression task on the set of all object pairs $O_P = O \times O$ with the differential features $d_{i,j}$ (Sec. 2.4):

$$\{d_{i,j} \equiv [d_1(o_i, o_j), \ldots, d_M(o_i, o_j)]^T\}_{i=1..N, j=1..N} \tag{3.13}$$

For an element $o_{i,j} \in O_P$, the model should estimate the posterior probability that objects o_i and o_j belong to the same cluster A_x, conditioned on the differential features:

$$f_s(o_{i,j}) \approx P(o_i \in A_x \wedge o_j \in A_x | d_{i,j}) \tag{3.14}$$

A binary response variable Y may be used to encode the event that two objects o_i, o_j are in the same cluster with value 1 and the opposite event with value 0. It follows that f_s approximates $P(Y = 1|d_{i,j}) = \mu_Y(d_{i,j})$, the conditional mean of Y (see Sec. 2.2.1).

Now the canonical Normalized Cut similarity function is examined more closely. Eq. 2.34 can be rewritten as:

$$f_s(o_i, o_j) = e^{-\sum_{k=1}^{M} \theta_k d_{i,j,k}^2} \tag{3.15}$$

where $\theta \equiv [\frac{1}{\sigma_1^2}, \frac{1}{\sigma_2^2}, \ldots, \frac{1}{\sigma_M^2}]^T$. Defining $r_{i,j} \equiv [d_{i,j,1}^2, d_{i,j,2}^2, \ldots, d_{i,j,M}^2]^T$ and setting $\theta_0 = 0$ leads to the following concise form:

$$f_s(o_i, o_j) = e^{-(\theta_0 + \theta^T r_{i,j})} \tag{3.16}$$

The non-negativity of the sum $\theta_0 + \theta^T r_{i,j}$ follows from the fact that each element of $\theta, r_{i,j}$ is itself non-negative as a squared real value. Also, note that the function $h(z) = e^{-z}$ maps the interval $[0; \infty)$ onto the interval $(0; 1]$, which corresponds to valid probability values. We can rewrite Eq. 3.16 as:

$$-ln[f_s(o_i, o_j)] = -ln[\mu_Y(r_{i,j})] = \theta_0 + \theta^T r_{i,j} \tag{3.17}$$

A comparison with Eq. 2.14 makes it apparent that Eq. 3.17 defines a Generalized Linear Model with a link function $g(\mu) = -ln(\mu)$ on the squared differential features, having a Bernoulli-distributed response variable. Therefore, specifying a set of weights σ_i for the Normalized Cut similarity function implicitly determines a probabilistic regression model. This could be interpreted as a theoretical justification for Eq. 2.34, but also as an explanation why it performs so well in practice.

Optimizing the model

It is possible to use the canonical method for fitting GLMs described in Sec. 2.2.3 to optimize the model specified by Eq. 3.17 on a training set by maximizing the corresponding log-likelihood induced by f_s. However, it should be noted that this would require the introduction of non-negativity constraints for each weight θ_i to the maximization problem. This is not part of the original formulation and would involve the use of more complicated optimization methods. Therefore, the functional form of f_s (Eq. 3.16) is slightly altered to handle negative feature weights:

$$f_{s'}(r) = e^{-|\theta_0 + \theta^T r|} \tag{3.18}$$

With the adjusted similarity function, standard unconstrained optimization techniques can be applied to obtain the maximum-likelihood weights. In particular, the Newton-Raphson method is used. Assume $\theta' = [\theta_0, \theta_1, \ldots, \theta_M]^T$ and $r'_i = [1, r_{i,1}, \ldots, r_{i,M}]^T$. Plugging $f_{s'}$ into the general log-likelihood equation (Eq. 2.8) as the probability P and differentiating twice with respect to the weights θ' yields:

$$\nabla l(\theta') = \frac{\partial l(\theta')}{\partial \theta'} = \sum_{i=1}^{N} r'_i \frac{sgn(f_{s'}(r_i))(f_{s'}(r_i) - Y_i)}{1 - f_{s'}(r_i)} \tag{3.19}$$

$$\nabla^2 l(\theta') = \frac{\partial^2 l(\theta')}{\partial \theta' \partial \theta'^T} = \sum_{i=1}^{N} r'_i r'^T_i \rho(r_i)[\frac{Y_i}{1 - f_{s'}(r_i)} - \rho(r_i) - 1] \tag{3.20}$$

In the above, the function $\rho(x) = \frac{f_{s'}(x)}{1 - f_{s'}(x)}$, whereas $sgn(x)$ indicates the sign function. The optimization starts with an initial guess θ'_0 and proceeds in an iterative fashion, where the current solution is updated according to the Newton ascent direction $p_{i+1} = -[\nabla^2 l(\theta'_i)]^{-1} \nabla l(\theta'_i)$:

$$\theta'_{i+1} = \theta'_i + \alpha_{i+1} p_{i+1} \tag{3.21}$$

The step lengths α_i are obtained using a backtracking line search, which ensures that they satisfy the Armijo condition (sufficient increase in the function value):

$$l(\theta'_i + \alpha_{i+1} p_{i+1}) \geq l(\theta'_i) + c_1 \alpha_{i+1} \nabla l(\theta'_i)^T p_{i+1} \tag{3.22}$$

where c_1 is a small positive constant.

Stopping criterion

In most applications of the recursive binary Normalized Cut, the prevailing method of implementing the stopping criterion seems to be based on a threshold for the Ncut value (Eq. 2.31), as proposed by Shi & Malik [2000]. The advantage of this approach is its generality, which makes it possible to apply to any matrix of object similarity weights, regardless of the nature of the scene that is being clustered. On the other hand, it could be argued that the threshold method suffers from several problems. First of all, the Ncut value is closely tied to the scale and properties of the similarity function. This makes it necessary to find a new threshold each time one wishes

to alter the similarity weighting scheme. Moreover, since the Ncut value is not related to any physical or scene-specific parameter whose value can be intuitively estimated, we must usually resort to some manner of exhaustive search or sensitivity analysis for determining the threshold value. Also, if we view the question of whether to stop or to continue partitioning of a scene as a binary classification problem, the feature set is rather sparse with only the Ncut value in the role of the independent variable. It is therefore reasonable to assume that one threshold will not be appropriate to describe all objects even in one scene, let alone for a variety of input scenarios. In this thesis, it is proposed to resolve these drawbacks by applying an appearance-based stopping criterion, wherein the decision to stop or continue partitioning depends exclusively on what the scene part looks like, and is in particular independent of the similarity weights as well as the Ncut value. Such a criterion is very application-specific and hinges on knowing in advance what we expect to find in the scene. However, in contrast to very general tasks of image segmentation, for the case considered in this work one usually focuses on a specific type of linear structure in any concrete application (e.g. fallen trees). Therefore it is possible to train an object appearance model tailored to the target object class and the data/sensor type, which is capable of distinguishing whether a scene contains exactly one versus more than one target object. Such a binary classifier can be directly used as a stopping criterion in the bi-partitioning strategy. The concrete implementation of this appearance model is once again application-specific. In Section 4.1, the concrete stopping criterion will be instantiated for the case of fallen trees and ALS point clouds.

3.1.4 Scene ranking

The conditional random field formulated in Eq. 3.11 has a number of term balance coefficients ($\beta(k)$,$\gamma(k)$), the values of which must be determined for processing any real input data. Moreover, it is also not clear a priori what value of the constraint coefficient λ will lead to the optimal segmentation result. Therefore, instead of committing to a single set of coefficients, a different approach is proposed here. The segment selection step using the icCRF formulation is executed on a grid of coefficient values β,γ, each time resulting in a sequence of solutions corresponding to different λ values (Sec. 3.1.2). For every point on the coefficient grid and every solution, the merging step is performed, which yields the set of segmented objects, referred to as a *scene*. The scenes are ranked based on the appearance of the segmented objects. Specifically, a numerical score is assigned to each scene in an additive fashion. Since the delineated objects are disjoint (by nature of the partitioning process), the score of a scene can be expressed as the sum of scores of individual objects. These individual scores are obtained from an external appearance model, which can output a normalized quality value for a single segmented object. This can be seen as a realization of the machine-learned ranking paradigm [Liu, 2009]. The highest-ranking scene constitutes the output of the entire processing pipeline, after undergoing a final classification step using the object appearance model to remove false positives. The specific form of this appearance model must be tailored to the concrete application.

3.2 Combining active and semi-supervised learning through Renyi entropy regularization

This section develops the methodology for combining two learning paradigms, active and semi-supervised learning, into a single workflow based on entropy regularization. Pool-based active learning (see Sec. 2.1.3) is the target scenario. From the semi-supervised side, the class of entropy-regularized discriminative models is considered, whereas the expected error reduction (EER) approach is the representative of active learning. The relationship between these two methodologies is discussed, with an emphasis on the aspect of minimizing the posterior class

label entropy of unlabeled samples. Also, the regularization term of the discriminative model is generalized from Shannon entropy to the broader class of Renyi entropies. A specialized method for optimizing the regularized model, based on deterministic annealing expectation-maximization, is then proposed, and finally the algorithm for integrating the Renyi entropy regularized model into the EER framework is given. Also, to ensure applicability in practice, a preprocessing step is introduces which reduces the unlabeled data quantity through a low-rank matrix approximation technique aiming at preserving most of the dataset's variance while selecting a subset of original unlabeled examples.

3.2.1 Candidate pre-selection

In real-world applications of machine learning techniques, the cardinality of the raw input data may be prohibitively large. Particularly for remote sensing problems, input data may comprise millions of entities (single trees, ground patches, 3D points etc.) However, the active learning method of choice in this thesis (Sec. 3.2.2) is characterized by high computational complexity, making it impossible to process data pools of more than several thousand samples within reasonable time on average computers. Instead of a direct application to the raw data pool, it is necessary to pre-select a subset of input instances with manageable size [Polewski et al., 2015a]. The simplest way of achieving this is through random sampling of the initial object pool. Instead, a greedy and unsupervised method for selecting the most representative objects is proposed based solely on their properties in the feature space. This has the advantage of not executing cost-intensive queries to any external labeling agent, while potentially discovering more characteristic instances compared to uniform random selection. Specifically, the sparse greedy matrix approximation technique developed by Smola & Schölkopf [2000] is applied. Consider the set of d-dimensional feature vectors x_i of N objects. The method starts with the *design matrix* $K = (K)_{ij} = k(x_i, x_j)_{i,j \in 1..N}$. The elements of K are essentially the similarities between feature vectors measured by the semi positive-definite kernel k which itself defines a reproducing kernel Hilbert space. The objective is to find a matrix $\bar{K}$ of lower rank such that $\bar{K} = KP_S$ is similar to K, where P_S is a matrix that projects K into the subspace S defined by the m largest eigenvalues of K. It is possible to obtain a good approximation if the $m \ll N$ largest eigenvalues account for the vast majority of the variance present in the data. This corresponds to the situation where a small subset of examples from the dataset contains almost all information about the entire pool. More formally, we are interested in finding a matrix $\bar{K}$ such that the Frobenius norm of the residual matrix is minimized:

$$||\bar{K} - K||^2_{Frob} = \sum_{i,j=1}^{N} (\bar{K} - K)^2_{ij} \tag{3.23}$$

This approximation is done in a greedy fashion, by iteratively picking a column K_i from K according to an evaluation criterion, orthogonalizing the remaining columns on K_i, and checking for convergence in the residual sum (Eq. 3.23). These ideas are formalized by Alg. 1. Let $\bar{K}_i$ denote the approximation of the ith column of K. Using the expansion coefficient matrix T, we can write:

$$\bar{K}_i = \sum_{j=1}^{n} K_{i(j)} T_{ji} \tag{3.24}$$

The Frobenius norm (Eq. 3.23) can be expressed in terms of the column residuals:

$$||\bar{K} - K||^2_{Frob} = \sum_{i=1}^{N} ||K_i - \bar{K}_i||^2 \tag{3.25}$$

At each step n, the evaluation criterion of a column j corresponds to the reduction in residuals which takes place when j is picked: $Q(j) = ||K - \bar{K}^I||^2_{Frob} - ||K - \bar{K}^{I\cup\{j\}}||^2_{Frob}$. Because the contribution of every picked column is removed by projecting it out of all remaining columns, the algorithm effectively works with column *residuals*. Therefore, the optimal new residuals sum (when choosing column j) can be written as:

$$\sum_{i=1}^{N} ||K_i - \bar{K}_i^{I\cup\{j\}}||^2 = \sum_{i=1}^{N} ||K_i - \bar{K}_i^I - t_i(K_j - \bar{K}_j^I)||^2 \tag{3.26}$$

The coefficient t_i in the above can be determined using the fact that in Hilbert spaces, orthogonal projections are optimal. It follows that ($< \cdot, \cdot >$ denotes the inner product):

$$t_i = ||K_j - \bar{K}_j^I||^{-2} < K_i - \bar{K}_i^I, K_j - \bar{K}_j^I > \tag{3.27}$$

Putting Eqs. 3.26 and 3.27 together, the final form of the evaluation criterion $Q(j)$ is given by the expression:

$$Q(j) = ||K_j - \bar{K}_j^I||^{-2} \sum_{i=1}^{N} < K_i - \bar{K}_i^I, K_j - \bar{K}_j^I >^2 \tag{3.28}$$

Therefore, in Algorithm 1 picking the best column is done with respect to the criterion given by Eq. 3.28, and the residuals are calculated based on Eq. 3.25. The output of the algorithm comprises a list of the indices of the most representative objects in the unlabeled pool, which is taken as the basis for the active learning procedure.

Algorithm 1 Greedy matrix approximation

```
1: function GreedyApprox(K, ϵ)
2:     n ← 0, I ← {}, T ← 0
3:     repeat
4:         n ← n+1
5:         M ← random sample of m columns ∉ I
6:         i_n ← PickBestColumn(M,K,T)
7:         I ← I ∪ i_n
8:         T ← ProjectOutColumn(K,T,i_n)
9:         res ← BoundResiduals(K,T)
10:    until res < ϵ
11:    return n,T,I,res
```

3.2.2 Expected Error Reduction

As previously stated, the principal role of an active learning system is to iteratively select the most informative example from an input data pool for labeling by the oracle. An appealing and theoretically well-founded strategy for performing this selection is the expected error reduction criterion, introduced by Roy & McCallum [2001]. To specify this method, a probabilistic setting is assumed where the classifier must learn an unknown conditional probability distribution $P(Y|x)$ of the class labels given the object features. To this end, the classifier is given an initial labeled training pool T and a pool U of unlabeled examples. Let $\hat{P}_T(Y|x)$ denote the distribution learned from T. Moreover, define the expected error of the classifier:

$$E(\hat{P}_T) = \int L(P(Y|x), \hat{P}_T(Y|x))P(x)dx \tag{3.29}$$

In Eq. 3.29, L indicates the loss function quantifying the cost associated with the discrepancy between the true and learned distributions (see Sec. 2.1.1). One of the more popular loss functions in case of classification is the *log-loss*:

$$L(x) = -\sum_{y \in \mathcal{Y}} [P(Y = k|x) \ln(\hat{P}_T(Y = k|x))] \tag{3.30}$$

The log-loss is thus equivalent to the cross entropy between distributions P and $\hat{P}_T$. In order to calculate the classifier error (Eq. 3.29), the knowledge of the true probability distribution P is necessary. Evidently, this information is usually not available during training. The EER algorithm circumvents this problem by making two simplifying assumptions. First, it only estimates the expected error on the pool U as opposed to the entire domain $\mathcal{X}$. Also, it approximates the true distribution P with the currently trained classifier's estimation $\hat{P}_T$. In this setting, the estimated expected error can be written as:

$$E(\hat{P}_T) = -\frac{1}{|U|} \sum_{x \in U} \sum_{y \in \mathcal{Y}} \hat{P}_T(y|x) ln(\hat{P}_T(y|x)) \tag{3.31}$$

Therefore, the expected error is effectively approximated using the entropy of the posterior class label probability of the underlying classifier. The EER method proceeds in a greedy iterative fashion, at each iteration picking the example x^* for which the error estimate $\hat{P}_{T \cup (x^*, y^*)}$ is the smallest. Since the actual label y^* is not known, EER takes the expectation of the estimated error weighted according to the current classifier's posterior (Eq. 3.32). The entire EER procedure is detailed by Alg. 2, where n_{max} refers to the number of objects that can be labeled by the expert within the specified resource constraints (time etc.).

$$e(x) = \sum_{y \in \mathcal{Y}} \hat{P}_T(y|x) E(\hat{P}_{T \cup (x,y)}) \tag{3.32}$$

Algorithm 2 Expected Error Reduction

```
1: procedure EER(T, P, U)
2:     while |T| < n_max do
3:         (c, P̂_T) ← trainClassifier(T)
4:         for x ∈ U do
5:             for y ∈ 𝒴 do
6:                 (c', P̂) ← trainClassifier(T ∪ (x, y))
7:                 e(x) ← e(x) + P̂_T(y|x) E(P̂)
8:         x* ← arg min e(x)
9:         T ← T ∪ (x*, y*), U ← U \ {x*}
```

Entropy can be seen as a measure of uncertainty in the posterior, therefore EER tends to select examples which increase the classifier's confidence in its decisions, i.e. shift the posterior probabilities towards the extreme values of 0 and 1. Guo & Greiner [2007] offer an alternative perspective on EER. They note that by choosing the example which minimizes the average entropy of posteriors on the unlabeled pool, we are also maximizing the mutual information between the example's label and the labels of all remaining unknown instances. The considerable effort of constantly retraining the classifier (Alg. 2) is mitigated by reducing the candidate pool through example pre-selection (Sec. 3.2.1), while the variance of the posterior probability estimates $\hat{P}_T$ may be reduced through bagging [Roy & McCallum, 2001].

3.2.3 Entropy Regularization

The entropy regularization framework, introduced by Grandvalet & Bengio [2006], is an implementation of the semi-supervised learning paradigm which allows to include unlabeled data into a discriminative classification model. The core idea is to bias the model to favor a decision boundary within a low-density area of the feature space, thus resulting in a possibly clear class separation. This is motivated by the fact that the information content of unlabeled samples decreases with an increasing overlap between the classes. The authors employ entropy conditioned on the features x as a measure of class overlap:

$$H(Y|X) = -\int \sum_{i \in \mathcal{Y}} ln[P(Y = i|x)]P(Y = i, x)dx \tag{3.33}$$

To avoid explicit modeling of the joint probability $P(Y, X)$, the sample average over objects in the unlabeled pool is plugged into Eq. 3.33, resulting in an 'empirical' entropy:

$$H_{emp}(Y|X) = -\frac{1}{|U|} \sum_{x \in U} \sum_{i \in \mathcal{Y}} P(i|x) ln[P(i|x)] \tag{3.34}$$

The entropy term defined by Eq. 3.34 may be applied as a regularizer to any posterior distribution model parameterized by θ:

$$C(\theta, \lambda; T, U) = \ell(\theta; T) - \lambda H_{emp}(Y|X; U, \theta) \tag{3.35}$$

The new optimization objective is therefore a sum of the original model's log-likelihood function ℓ on the (labeled) training set and the negated entropy on the unlabeled pool. The coefficient λ controls the influence of the entropy term and is usually determined empirically.

Renyi entropy

The ER methodology was originally developed for Shannon entropy, however the authors remark that it is also applicable to other kinds of class overlap measures. In this work, the use of Renyi entropies for this purpose is explored. The family of Renyi entropies [Xu & Erdogmuns, 2010] was introduced as a means of generalizing Shannon entropy to a larger class of functions while preserving additivity of information for independent events and also remaining consistent with the axioms of probability. Renyi entropy is parameterized and its general functional form is:

$$H_\alpha(P) = \frac{1}{1 - \alpha} \log(\sum_{k=1}^{N} p_k^\alpha) \tag{3.36}$$

In the above, P represents a probability distribution over N events, and $\alpha > 0$ is the parameter, where $\alpha \neq 1$. Renyi entropy H_α is a generalization of Shannon's entropy H_S in the sense that as α tends to 1 (from both sides), H_α approaches H_S:

$$\lim_{\alpha \to 1^-} H_\alpha(P) = \lim_{\alpha \to 1^+} H_\alpha(P) = H_S(P) \tag{3.37}$$

The choice of Renyi entropy is motivated by its successful application in several fields, such as statistical inference [Pardo, 2005] or quantum mechanics [Bengtsson & Zyczkowski, 2006], which suggests that the most widely-known Shannon entropy is neither the only available, nor the best suited entropy measure for a particular purpose.

Deterministic Annealing Expectation-Maximization

Optimizing the ER model defined by criterion C (Eq. 3.35) is not a trivial task. Even if the two elements of the model - log-likelihood and entropy - are concave w.r.t. the parameter θ, their weighted difference is (in general) neither convex nor concave, which entails the existence of local extrema. Grandvalet and Bengio propose using a variation of the Deterministic Annealing Expectation-Maximization algorithm (DAEM) [Ueda & Nakano, 1998] to deal with this issue. The core idea behind DAEM is to redefine the posterior probability distribution of the missing data in order to alter the E-step within the standard Expectation-Maximization algorithm (see Sec. 2.1.6). Let X_o and X_m denote, respectively, the observed and missing data. The E-step then consists in constructing the conditional expectation of the joint (over observed and missing data $Z = (X_o, X_m)$) log-likelihood given X_o and current parameter estimate $\hat{\theta}^i$ (cf. Eq.2.10):

$$E[\ell(\theta; Z)|X_o; \hat{\theta}^i] = \int P(X_m|X_o; \hat{\theta}^i) \log P(Z; \hat{\theta}^i) dX_m \tag{3.38}$$

In standard EM, the missing data posterior is obtained by applying Bayes's theorem:

$$P(X_m|X_o; \hat{\theta}^i) = \frac{P(X_o, X_m; \hat{\theta}^i)}{\int P(X_o, X'_m; \hat{\theta}^i) dX'_m} \tag{3.39}$$

Unfortunately, this standard posterior is highly dependent on the parameter initialization $\hat{\theta}^0$, and therefore may be unreliable in the early iterations of EM. To remedy this, Ueda & Nakano [1998] propose a new posterior P' derived through the maximum entropy principle. They start with the assumption that the specific values of P' are not known, but the *macro state* associated with the observed data is known, i.e. the expectation of the complete data log-likelihood (Eq. 3.38) w.r.t. P'. They define P' as the probability distribution with the maximum entropy among all distributions which are consistent with the macro state. This can be formalized using the following functional (H denotes entropy):

$$J[P'] = H(P') + \beta(E_{P'}[\ell|X_o] - c) + \rho(\int P' dX_m - 1) \tag{3.40}$$

In the above, β, ρ are Lagrange multipliers associated with constraints, c is a constant and the last constraint arises from the condition that a probability distribution must sum to unity. The maximizing P' may be found by applying calculus of variations. In particular, the authors show that for Shannon entropy H_S, the optimal P' is given by:

$$P'_S(X_m|X_o) = \exp(\beta \log P(X_o, X_m; \hat{\theta}) + \rho - 1) \tag{3.41}$$

The normalizing constant ρ's value can easily be determined from the property of P'_S summing to unity, resulting in the final form of the posterior (for Shannon entropy):

$$P'_S(X_m|X_o; \hat{\theta}) = \frac{P(X_o, X_m; \hat{\theta})^\beta}{\int P(X_o, X'_m; \hat{\theta})^\beta dX'_m} \tag{3.42}$$

The value of $1/\beta$ can be interpreted in analogy to the 'temperature' in a simulated annealing process. By plugging P'_S into the E-step (Eq. 3.38), the basis of the DAEM algorithm is obtained. The entire DAEM consists of iteratively running the EM algorithm with the altered E-step for a series of β, where an initially high temperature value is gradually lowered, while always starting at the solution obtained at the preceding temperature. Grandvalet & Bengio [2006] notice that the class of optimization problems solved through DAEM has the same functional form as the ER criterion C (Eq. 3.35). Each β parameter value determines a corresponding λ' coefficient defining

the balance between log-likelihood and entropy. The idea is to start the DAEM iteration at $\beta = 1$, where the original EM posterior is recovered and entropy is not considered at all ($\lambda' = 0$). As the temperature decreases, the posterior becomes more peaked and the entropy term gains significance. The iteration is executed until the chosen (fixed) λ value (Eq. 3.35) is attained. In the ER setting, the observed data X_U corresponds to the feature values of objects from the unlabeled pool, while the missing data Y_U are their unknown labels. The posterior P'_S can be expressed using the discriminative model's class probabilities for each instance:

$$P'_S(y|x_i;\hat{\theta}) = \frac{P(y|x_i;\hat{\theta})^{\frac{1}{1-\lambda}}}{\sum_{y'} P(y'|x_i;\hat{\theta})^{\frac{1}{1-\lambda}}} \tag{3.43}$$

The optimization objective in the M-step thus takes the form:

$$\theta^{i+1} = \arg\max_{\theta} \ell(\theta;T) + \sum_{x\in U}\sum_{y'} P'_S(y'|x;\hat{\theta}^i)\log P(y'|x;\theta) \tag{3.44}$$

3.2.4 Regularizing with Renyi entropy

In this section, the DAEM algorithm, developed specifically for Shannon entropy, is extended to the class of Renyi entropies. Starting with the functional of the posterior P' (Eq. 3.40), Renyi entropy H_α may be plugged in the role of $H(P')$. Unfortunately, because of the logarithm in H_α (Eq. 3.36), the new functional is no longer of the form required to apply the Euler-Lagrange equation, where all terms are integrated over a common interval. This problem is circumvented through a simplifying approximation. The logarithm in H_α is dropped and instead optimization is carried out on following functional based on the logarithm's argument, with the sign due to the constant $1/(1-\alpha)$:

$$\begin{aligned} J_\alpha[P'_\alpha] = \int [&\frac{1}{1-\alpha} P'_\alpha(Y)^\alpha + \beta \log P(Y|x) P'_\alpha(Y) \\ &- c_1 + \rho P'_\alpha(Y) - c_2] dY \end{aligned} \tag{3.45}$$

In the above, the dependence on the model parameter θ is omitted to ease the notational burden, and the missing data are represented explicitly as Y, the labels of objects in the unlabeled pool. As the logarithm is a strictly increasing function, maximizing its argument will also lead to the maximum logarithm value. However, the balance between the 3 non-constant terms in J_α will be altered compared to J, so optimizing these two functionals is not strictly equivalent. Nevertheless, intuitively it should still be possible to achieve a good maximizer of J by following the trajectory formed by maximizers of J_α obtained through annealing over a wide range of β values. The function P'_α maximizing J_α may be found analytically by applying the Euler-Lagrange equation, which in this case boils down to equating the derivative of the integrand in Eq. 3.45 with respect to P'_α to zero, resulting in:

$$P'_\alpha(Y) = [(-\beta \log P(Y|x) - \rho)\tau]^{\frac{1}{\alpha-1}} \tag{3.46}$$

The constant τ is defined as $\tau = (1-\alpha)/\alpha$. Contrary to the case of Shannon entropy, the normalizing constant ρ is not multiplicative, but additive. Its value may be determined numerically from the normalization equation $\sum_i P'_\alpha(y_i) = 1$. By imposing bounds on the probabilities, $0 \le P'_\alpha(y_i) \le 1$, an admissible interval for ρ may be derived ($p_i \equiv P(y_i|x)$):

$$\rho \in \begin{cases} [-\infty; \min_i -\beta \log p_i - \tau^{-1}] & \text{if } \alpha < 1 \\ -\beta[\min_i \log p_i; \max_i \log p_i - \tau^{-1}] & \text{if } \alpha > 1 \end{cases} \tag{3.47}$$

The optimization problem remains one-dimensional regardless of the label domain $\mathcal{Y}$'s cardinality, since to apply the prior to a new input distribution, we only need to find the root of the single-variable normalization equation w.r.t. ρ. Note that for certain combinations of $\alpha > 1, \beta$ and certain probability distributions, the admissible interval may be empty and a normalizing constant will not exist. A potential drawback of the variational approach is that it is an approximation and as such may fail to provide the exact maximizer of J.

Finally, the issue of finding the right scale for the annealing parameter β should be addressed. For the Shannon prior (Eq.3.42), clearly a value of $\beta = 1$ causes $p'(y_i)$ to be exactly equal to $P(y_i|x)$. Also, for $\beta > 1$ the prior becomes more peaked, and for $\beta < 1$ it flattens towards the uniform distribution. This makes it easier to determine the boundary values β_{min}, β_{max} for use in the annealing scheme. However, in case of the Renyi prior there is no longer any such single β value with this property. Instead, the equilibrium β_0 is now a function of $P(y_i|x)$ and α. In this setting, finding β_0 for a chosen p_i reduces to solving a system of two equations in two variables β, ρ:

$$\begin{cases} [(-\beta \log p_i - \rho)\tau]^{\frac{1}{\alpha-1}} = p_i \\ \sum_{j \neq i} [(-\beta \log p_j - \rho)\tau]^{\frac{1}{\alpha-1}} = 1 - p_i \end{cases} \tag{3.48}$$

In particular, for two classes, β_0 and the corresponding ρ can be found analytically, using the fact that the Bernoulli distribution possesses only 1 degree of freedom and so $p_2 = 1 - p_1$. Exponentiating both equations to $\alpha - 1$ and substituting β_0 into the second equation, we obtain:

$$\begin{aligned} \rho &= \frac{\tau^{-1}[(1-p_1)^{\alpha-1} \log p_1 - p_1^{\alpha-1} \log(1-p_1)]}{\log[(1-p_1)/p_1]} \\ \beta_0 &= -(\tau^{-1} p_1^{\alpha-1} + \rho)/\log p_1 \end{aligned} \tag{3.49}$$

Gathering the results discussed in this section, Listing 3 explicitly describes the DAEM$_\alpha$ algorithm, which generalizes ordinary Deterministic Annealing Expectation-Maximization to Renyi α-entropy [Polewski et al., 2016b].

3.2.5 Expected Error Reduction with Entropy Regularization

The algorithm for combining active learning, in the form of Expected Error Reduction, with semi-supervised learning, implemented by the Entropy Regularization Framework, is described in this section. It seems reasonable to integrate these two approaches, because both draw on the concept of entropy. Indeed, by comparing the EER candidate selection criterion (Eq. 3.31) and the ER optimization objective (Eq. 3.35), we see that in each of them the average entropy of the classifier's posterior probabilities on the unlabeled objects is minimized. Note that the entropy serves different purposes in the two methods. Whereas for EER, the entropy estimates the total classification error, in case of ER it is a measure of overlap between the classes. Nevertheless, an intuitive motivation for using these two methods together is that they may benefit from a synergy effect, because EER actively chooses examples whose addition to the training set yields the minimal unlabeled entropy, and ER attempts to bias the classification model further towards low entropy and away from the pure labeled data likelihood-based solution. Based on computational cost considerations, a 'shallow' version of integration is proposed, where the ER classifier is used at the top level of the EER loop (Alg. 2, line 3), but not for training the classifiers on the data augmented with the candidates (Alg. 2, line 6). Therefore, the bulk of the additional computational effort associated with re-training the entropy-regularized classifier in the inner algorithm loop can be avoided. This cost would not be sustainable in any real-world application, since it could exceed the effort of training the baseline (pure-likelihood) model by 1-2 orders of magnitude because of potentially many outer iterations in the Deterministic Annealing optimization scheme.

Algorithm 3 DAEM$_\alpha$ for classification

function EM-STEP$_\alpha(T,U,\theta,\beta)$
  for $x \in U$ **do**
    $\rho \leftarrow solve\left\{\sum_i [(-\beta \log P_\theta(y_i|x) - \rho)\tau]^{\frac{1}{\alpha-1}} = 1\right\}$
    $\forall_y P'_y(x) \leftarrow [(-\beta \log P_\theta(y|x) - \rho)\tau]^{\frac{1}{\alpha-1}}$
  return $\arg\max_{\theta'} \ell(\theta';T) + \sum_{x\in U}\sum_y P'_y(x) \log P(y|x;\theta')$
function FIND-$\beta_0(\alpha)$
  for $i = 1..N_P$ **do**
    generate random categorical distribution P over $\mathcal{Y}$
    $\beta^i \leftarrow \max_{p\in P} \beta_0(\alpha, P, p)$ ▷ *Use Eq. 3.48*
  return $\max \beta^i$
function DAEM$_\alpha(T,\ U)$
  $\beta_{min} \leftarrow \epsilon \approx 0$ ▷ *Start with dominating entropy term*
  $\beta_{max} \leftarrow k \cdot$ *Find-$\beta_0(\alpha)$* ▷ *Ensure enough iterations*
  $\beta \leftarrow \beta_{min}$, $\theta^0 \leftarrow 0$, $i \leftarrow 1$
  while $\beta \leq \beta_{max}$ **do**
    while *not converged* **do**
      $\theta^i \leftarrow$ *EM-Step$_\alpha(T,U,\theta^{i-1},\beta)$*
      $i \leftarrow i + 1$
    $\beta \leftarrow \beta\sigma$ ▷ $\sigma > 1$*: annealing factor*
  return $\arg\max_{\theta^i} \ell(\theta^i;T) - \lambda\frac{|T|}{|U|} H_\alpha(Y|X;U,\theta^i)$

Algorithm 4 Expected Error Reduction with Renyi Entropy Regularization

procedure EER-ER(T, U, α)
  while $|T| < n_{max}$ **do**
    ▷ Solve using DAEM$_\alpha$:
    $\hat{\theta}_T \leftarrow \arg\max \ell(\theta;T) - \lambda\frac{|T|}{|U|} H_\alpha(Y|X;U,\theta)$
    for $x \in U$ **do**
      for $y \in \mathcal{Y}$ **do**
        $\hat{\theta} \leftarrow \arg\max \ell(\theta; T \cup (x,y))$
        e(x) $\leftarrow$ *e(x)* $+ P(y|x;\hat{\theta}_T) E(P(Y|X;\hat{\theta}))$
    $x^* \leftarrow \arg\min$ *e(x)*
    $T \leftarrow T \cup (x^*, y^*)$, $U \leftarrow U \setminus \{x^*\}$

On the other hand, note that the top-level ER classifier still influences the choice of objects, because it is used for weighting the entropies resulting from adding the candidate objects to the training pool (Eq. 3.32) in order to form the final selection criterion. The entire procedure is given in Alg. 4.

3.3 Continuous space voting-based detection of cylindrical surfaces

Consider a dense point cloud P. In this section, a method is presented for detecting and reconstructing cylindrical shapes within P based on a voting scheme in a continuous parameter space. Several shapes may be contained in P, and the goal is to recover them simultaneously, i.e. without the need for multiple iterations where the previously detected inlier points are removed. The proposed approach is based on ideas associated with the Hough transform, notably the concept of parameter accumulation and voting, however it is designed to overcome some limitations of its predecessor which appear in the setting of cylinder detection. First, the Hough transform requires an exhaustive generation of surface samples from different parameterizations of the target shape (e.g. circle). While this is, perhaps, less of a concern for 2D raster images with limited size and one- or two-dimensional parameter spaces, in dense point clouds representing large outdoor areas this procedure may become prohibitively expensive in terms of computation time. Moreover, the size of the parameter accumulation bin is crucial to the quality of the obtained parameter values [Tran et al., 2015], but usually this bin size is determined empirically, which might be difficult to specify a priori. To alleviate the former issue, the method presented here does not generate shape samples and instead relies on votes cast by pairs of neighboring points from the input point cloud to determine the parameter space based on relationships between adjacent surface normals intrinsic to cylindrical surfaces. In order to address the latter problem, rather than discretizing the parameter space, a continuous strategy is proposed. A kernel density estimator (see Sec. 2.3) is constructed, and its maxima are located using continuous optimization methods. A key advantage is that for the KDE a number of robust methods are available for automatically determining the bandwidth size, relieving the user of the necessity to manually specify a quantization level for the parameter space.

A cylinder can be represented by 5 parameters, 4 for the axis and 1 for the radius [Beder & Förstner, 2006]. Here, a sequential parameter estimation scheme is employed, similar to [Rabbani & van den Heuvel, 2005]. First, probable orientations of cylinder axes are determined. Then, for each axis candidate, potential cylinder center locations are found based on points which voted for that axis. Finally, for each such pair of axis orientations and centers, the most likely radii are calculated. All cylinder parameters are estimated based on relationships between adjacent surface normals which are specific to cylindrical shapes (Sections 3.3.2,3.3.3). Each pair of surface normals of nearby points casts a vote for specific parameter values, and the set of all votes serves as a basis for constructing a kernel density estimator. Strong local maxima of the KDE (relative value of at least k_{kde} of global maximum) can then be located and are considered the estimated quantities (Sec. 3.3.4). A pruning step is carried out at the end of processing to remove false positive cylinder hypotheses (Sec. 3.3.7). The entire procedure is summarized in Algorithm 5.

3.3.1 Surface normal estimation

Since the surface normals play a crucial role in the proposed cylinder detection framework, it is imperative that they be estimated as accurately as possible. Here, a normal calculation method based on quadric surface fitting is utilized. The motivation for this choice is twofold. First, Klasing et al. [2009] have shown that quadric normals may, under certain conditions, be more accurate

Algorithm 5 Statistical cylinder detection

```
function getLocalMaxSamples(kde,localMax)
    return all samples from kde within
            the basin of attraction of localMax
function DetectCylinders(points,normals)
    cylinders ← {}
    axisV ← collectAxisVotes(points,normals)
    kde_ax ← constructKDE(axisV)
    axes ← getLocalMaxima(kde_ax, k_kde)
    for a ∈ axes do
        idxA ← getLocalMaxSamples(kde_ax,a)
        (pA,nA) ← points[idxA],normals[idxA]
        (ctrV,radV) ← collectCtrRadiusVotes(pA,nA)
        kde_ct ← constructKDE(ctrV)
        ctr ← getLocalMaxima(kde_ct, k_kde)
        for c ∈ ctr do
            idxC ← getLocalMaxSamples(kde_ct,c)
            kde_r ← constructKDE(radV[idxC])
            rad ← getLocalMaxima(kde_r, k_kde)
            for r ∈ rad do
                cylinders ← cylinders ∪ C(a,c,r)
    return pruneFalsePos(cylinders)
```

compared to the standard local plane fitting method. Second, since cylinders are themselves quadric surfaces, it is hypothesized that cylindrical surfaces could benefit from this specialized form of normal estimation. To reduce the computational burden, a constrained quadric fitting procedure is applied [Vanco, 2002] where the Z axis is assumed to be known. Specifically, to calculate the normal vector of a point p, first PCA is performed on the 3D structure tensor (see Sec. 2.6) of its neighborhood $N(p)$ to determine the dominant direction z_p. The points $N(p) \cup p$ are then rotated so that z_p becomes the Z axis of the new coordinate system. Let $Q = \{q_i\}_{i=1..n(p)}$ indicate the rotated points. Then, a reduced quadric surface S is fitted to Q in the least squares sense:

$$S = k_1 x^2 + k_2 y^2 + k_3 xy + k_4 x + k_5 y + k_6 - z$$

$$k^* = \arg\min_k ||Ak - Q_z||_2$$

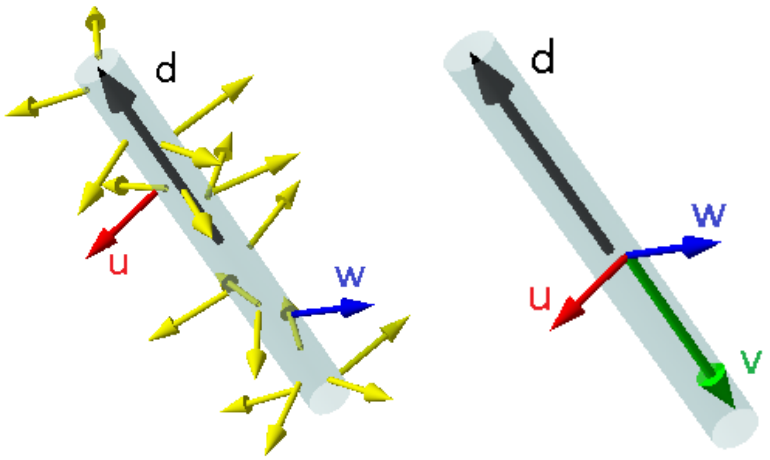
$$A = \begin{pmatrix} q_{1,x}^2 & q_{1,y}^2 & q_{1,x}q_{1,y} & q_{1,x} & q_{1,y} & 1 \\ & & \vdots & & & \\ q_{n(p),x}^2 & q_{n(p),y}^2 & q_{n(p),x}q_{n(p),y} & q_{n(p),x} & q_{n(p),y} & 1 \end{pmatrix}$$

In the above, $k = [k_1, \ldots, k_6]$ and Q_z is the third column of Q containing the transformed Z coordinates. The optimal solution k^* of this overdetermined linear system can be found through eigendecomposition of $A^T A$. Then, the surface normal at p may be expressed (in the rotated coordinate system) as $n_Q = [1, 0, k_4^*]^T \times [0, 1, k_5^*]^T$. The final step consists in rotating n_Q back to the original coordinate frame.

3.3.2 Calculating orientation

The algorithm for determining potential cylinder orientations is based on the observation that on a cylindrical surface, the cross-product between two surface normals (excluding parallel and

antiparallel pairs) results in a vector that is collinear with the cylinder's axis (Fig. 3.3). This



(a) (b)

Figure 3.3: Relationship between cylinder's axis and surface normals. (a) Cylinder with axis d and surface normals. (b) The cross product v of normals u, w is collinear with d.

gives rise to the following voting scheme. For each point p within the input cluster, its spherical neighborhood $N_r(p)$ of radius r is considered, and the cross product between all pairs of surface normals of points from $N_r(p)$ is calculated. Then, the *spatial median* of all cross products can be expressed as [Polewski et al., 2016a]:

$$v_p = \arg\min_{v \in \mathbf{R}^3} \sum_{p_i, p_j \in N_r(p)} ||(n_i \times n_j)/||n_i \times n_j|| - v|| \tag{3.50}$$

The spatial median of a set of vectors is a multivariate generalization of the usual univariate median, defined as the vector minimizing the average distance to all the set's elements. The algorithm by Kärkkäinen & Äyrämö [2005] is used for efficiently calculating the spatial median. The *principal direction* at point p is defined as $d_p \equiv v_p/||v_p||$. The set of principal directions over all the input points represents potential cylinder orientations voted for by the observed data. Intuitively, dominant directions should produce densely populated clusters within the space $\mathbf{R}^3$, so the task of obtaining candidate orientations boils down to finding local maxima in the voting space. This task can be further simplified by projecting the principal directions onto the Gaussian sphere, which yields spherical coordinates for each d_p [Rabbani & van den Heuvel, 2005]:

$$\begin{cases} r & = \sqrt{x^2 + y^2 + z^2} = 1 \\ \theta & = \tan^{-1} \frac{y}{x} \\ \phi & = \cos^{-1} z \end{cases} \tag{3.51}$$

Note that since d_p has unit length, the r coordinate is constant and does not need to be considered, reducing the task to two dimensions. To disambiguate the signs of the principal directions, all d_p are flipped so that they all lie on one hemisphere of the Gaussian sphere. The actual location of maxima in this reduced parameter space (θ, ϕ) is done on the basis of a kernel density estimator (Sec. 3.3.4).

3.3.3 Calculating position and radius

Estimating the axis position is carried out for a particular cylinder orientation d (see Algorithm 5). First, the input points P_d are found which lie in the basin of attraction associated with d's

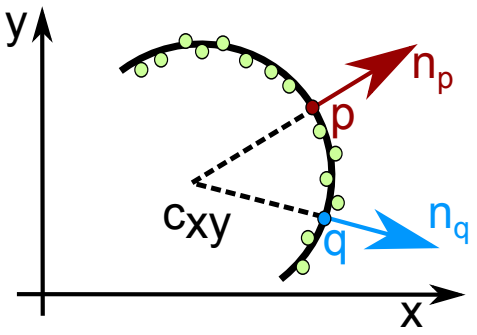


Figure 3.4: Points on cylindrical surface projected onto plane perpendicular to the cylinder's axis form an arc. The intersection of lines containing points p, q and their associated normals n_p, n_q lies at the cylinder's planimetric center c_{xy}.

local maximum in the KDE (Sec. 3.3.6). The points P_d are transformed to a new coordinate system, the Z axis of which is aligned with d. In this frame of reference, the (true) surface normals of the cylinder lie on the XY plane. This suggests the following strategy for conducting the position estimation in 2D. Let Q_d, U_d represent respectively the transformed coordinates of P_d and the associated surface normals. $P_{XY}(v)$ indicates projection of v onto the XY plane. Consider an input point $q \in Q_d$ and its neighborhood $N_r(q)$. For two points $q_i, q_j \in N_r(q)$, the line passing through $P_{XY}(q_i)$ and containing $P_{XY}(u_i)$ should intersect with the line passing through $P_{XY}(q_j)$ and containing $P_{XY}(u_j)$ precisely at the planimetric center of the cylinder in the transformed coordinate system (see Fig. 3.4). This yields the following linear equation system in two parameters s, t [Polewski et al., 2017b]:

$$\begin{pmatrix} u_{i,x} & -u_{j,x} \\ u_{i,y} & -u_{j,y} \end{pmatrix} \begin{pmatrix} s \\ t \end{pmatrix} = \begin{pmatrix} q_{j,x} - q_{i,x} \\ q_{j,y} - q_{i,y} \end{pmatrix} \tag{3.52}$$

From s, t, the votes for the center position $c(q_i, q_j)$ and for the radius $r(q_i, q_j)$ may be directly recovered:

$$\begin{cases} c(q_i, q_j) & = P_{XY}(q_i) + sP_{XY}(u_i) = P_{XY}(q_j) + tP_{XY}(u_j) \\ r(q_i, q_j) & = ||c(q_i, q_j) - P_{XY}(q_i)|| = ||c(q_i, q_j) - P_{XY}(q_j)|| \end{cases} \tag{3.53}$$

In practice, the surface normals are calculated with a certain error. For points situated very close to each other, their normal vectors are nearly parallel, and even small errors in normal directions may lead to large discrepancies in solving Eq. 3.53. Therefore, a vote from a pair of vectors q_i, q_j is considered valid only if $||q_i - q_j|| >= d_{min}$ and the voted-for radius $r(q_i, q_j)$ is within a pre-specified limit $[r_{min}; r_{max}]$, otherwise the vote is discarded. For aggregating the votes and extracting the local maxima, the spatial median of the position votes is calculated, followed by application of the same KDE based technique as in the case of orientation (Sec. 3.3.4). At the bottom level of the cylinder reconstruction pipeline, for each recovered cylinder center position, the points which voted for it are determined from the associated KDE peak's basin of attraction. Then, a simple 1D KDE is constructed on the radius votes $r(q_i, q_j)$. The local maxima from this KDE provide the most likely radius values for the cylinder hypotheses, thus completing the information obtained in the previous two steps (orientation, position) as described in Alg. 5. Finally, the extracted cylinder positions must be transformed back to the original coordinate reference system.

3.3.4 Constructing KDE and maxima location

Given the set of input votes V, a kernel density estimator is constructed using V as samples. For direction and position determination, the voting space is two-dimensional, whereas in case of

radius estimation, the dimensionality is 1. This ensures that the number of samples required to reliably estimate the modes of V's distribution is moderate. For selecting the kernel bandwidth, the plug-in estimation method (see Sec. 2.3.1) is used due to its performance in practice and good theoretical properties for finite sample data of moderate dimensionality. This part of the voting mechanism operates in continuous space and is effectively parameterless from the user's perspective. To determine the modes of the underlying probability distribution, the KDE is first discretized on a grid followed by local maxima detection on the grid by examining each cell's neighbors. Each local maximum is a starting point for gradient ascent search, which yields the refined continuous extremum positions. Finally, local maxima are suppressed for which there exists a stronger local maximum within d_g grid cells, as well as ones whose value lies below k_{kde} of the KDE's global maximum.

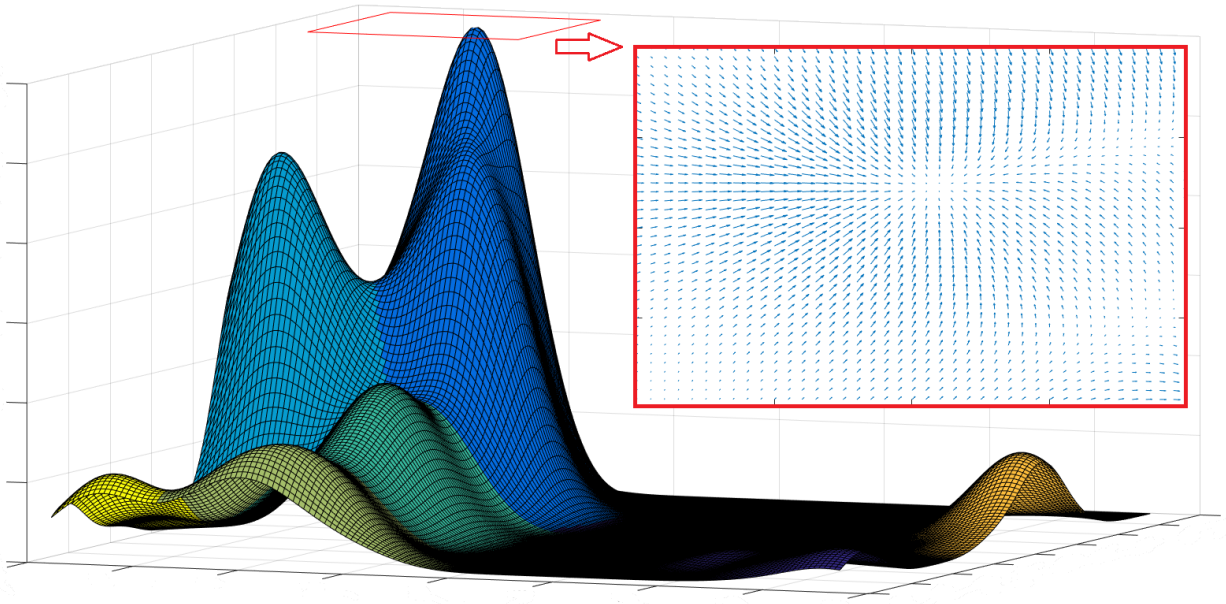


Figure 3.5: Basins of attraction for a complex KDE surface with several local maxima, marked by different colors. Each local maximum in 2D parameter space determines a cylinder's orientation or position. Red box: top view of discrete KDE gradient calculated over grid. Arrows show direction and magnitude of gradient.

3.3.5 Weighting the votes

The voting scheme presented thus far operates under the tacit assumption that all votes have the same influence on the cylinder parameter selection process. However, oftentimes additional, application-dependent information is available which describes the suitability of individual laser points as inliers of a cylindrical surface. Such information could be used to differentiate the relative strength of votes cast by each point. One of the benefits of applying the KDE framework is that it provides a natural extension which allows the introduction of *weights* into the voting process. First, recall that due to the aggregation of votes through the spatial median, one point casts exactly one vote for each of the cylinder parameters (orientation, position, radius). Therefore, it is straightforward to formulate a KDE-based model P_{kde} where the contribution of each point p_k is weighted by a nonnegative scalar w_k:

$$P_{kde}(\pi) = \frac{|B|^{-1/2}}{n} \sum_{k=1}^{n} w_k K(B^{-1/2}(\pi - \pi_k)) \tag{3.54}$$

The term π represents the currently estimated parameter, i.e. the cylinder's orientation, position, or radius, whereas the values π_k are the votes cast by each of the n data points. The weights w_k are such that $\sum_k w_k = n$, to maintain compatibility with the uniformly weighted version. This ensures that the total 'mass' of influence remains the same, it is only distributed unevenly among the points based on their relative quality. The symbol K denotes a multivariate kernel function. The kernel bandwidth matrix B must be estimated from the data. Unfortunately, the introduction of weights into the KDE significantly complicates optimal bandwidth selection. As noted by Wang & Wang [2007], even in the univariate case a closed-form solution for minimizing the asymptotic mean integrated squared error (AMISE) criterion, which is the basis of the plug-in bandwidth selection method, is not available. Therefore, a simple approximation proposed in [Wang & Wang, 2007] is applied, where the bandwidth is estimated based on the unweighted version of the data.

3.3.6 Finding points belonging to KDE maximum

As the KDE surface may be arbitrarily complex with multiple adjacent local maxima, a method is necessary for delineating the area of influence, or basin of attraction, of each peak. This is to ensure that only points associated with a certain cylinder orientation vote together for the axis position, and only points which voted for a common position are allowed to determine the radius for that particular axis/position combination. One way to check which basin of attraction a point p belongs to would be to apply gradient ascent search starting at p. However, for potentially millions of input points this becomes computationally impractical. In this thesis a different, highly efficient approximation is proposed. To that end, the discretized KDE grid described in Sec. 3.3.4 is utilized. Let $N_8(i,j)$ be the 8-pixel neighborhood of element (i,j) on the grid. $G(i,j)$ denotes the discrete KDE gradient over the grid, evaluated at (i,j). The gradient 'points to' the grid cell yielding the highest increase in function value (Fig. 3.5). To find grid points associated with a local maximum grid cell c, a breadth-first search (BFS) is executed starting at c on the graph with vertices corresponding to all grid elements, and a directed edge for every pair of points $(i,j),(k,l)$ such that $(k,l) \in N_8(i,j)$ and $G(k,l)$ points to (i,j). Then, it is determined which original points belong to the grid cells that were visited by the BFS procedure. The result of separating basins of attraction for a complex KDE surface is shown in Fig. 3.5.

3.3.7 Pruning false positives

The final part of the cylinder detection framework serves to eliminate cylinder hypotheses which are not supported by the data to a sufficient degree. This pruning step is the most application-dependent, since it usually makes assumptions about the appearance of the target cylindrical shapes in the scene, and/or about the sensor and data acquisition process. Nevertheless, some generic pruning criteria may be defined at this step. First, let $I(C_i)$ denote the set of cylinder C_i's inlier points, which are within a distance of at most d_{in} from C_i's surface. The inliers may be projected to the surface, which results in a new set of 2D coordinates l, a for each point $p_{i,j} \in I(C_i)$:

$$l_{i,j} = \frac{p_{i,j} \cdot \boldsymbol{d_i} - c_i \cdot \boldsymbol{d_i}}{\boldsymbol{d_i} \cdot \boldsymbol{d_i}} \tag{3.55}$$

$$a_{i,j} = \begin{cases} \cos^{-1}[\boldsymbol{v}_{i,j} \cdot \boldsymbol{z}_i], & \text{if } (\boldsymbol{v}_{i,j} \times \boldsymbol{z}_i) \cdot \boldsymbol{d_i} \geq 0 \\ -\cos^{-1}[\boldsymbol{v}_{i,j} \cdot \boldsymbol{z}_i], & \text{else} \end{cases}$$

In the above, $\boldsymbol{d_i}, c_i$ refer to C_i's direction and point on axis, whereas $\boldsymbol{v}_{i,j} = p_{i,j} - (c_i + l_{i,j}\boldsymbol{d_i})$ is the normalized vector difference between an inlier point and its projection onto the cylinder axis. The term z_i denotes cylinder C_i's reference 0° direction, which may be chosen based on the specific

application. The l coordinate expresses the projected point's position along the cylinder axis, while the a coordinate corresponds to the signed angle between $\boldsymbol{v}_{i,j}$ and the reference direction z_i. The length of C_i can then be expressed as $max_{j,k}|l_{i,j} - l_{i,k}|$. Different filtering criteria may then be applied based on $I(C_i)$ as well as $\{l_{i,j}, a_{i,j}\}$, for example the minimum number of inliers or minimum cylinder length. Moreover, the 2D coordinates l, a make it possible to characterize the distribution of the inlier points on the cylinder's surface, and design application-specific criteria such as the minimum occupancy ratio etc.

4 Detection of fallen trees from point clouds

This chapter describes the specialization of two general methods from Chapter 3 for detecting fallen trees from ALS and TLS point clouds of forested areas. Section 4.1 outlines the former case, where the framework of segmenting linear structures through decomposition into primitives (Sec. 3.1) is applied to ALS data. The latter scenario is treated in Section 4.2, which builds upon the voting-based cylinder reconstruction approach (Sec. 3.3). Both of the methods presented in this chapter utilize a coarse-to-fine, bottom-up strategy of starting with point-level information, generating an intermediate representation in terms of primitives, and merging them to obtain individual objects (Fig. 4.1). At the point level, the goals and techniques are similar for the two processing pipelines. First, a DTM filtering step expresses the prior knowledge about the distribution of fallen tree points within an above-ground height interval. Furthermore, a point classification step is performed to retain only stem points for further processing and in particular eliminate points likely associated with shrubs, ground vegetation as well as other irrelevant objects. At the level of forming primitives, the methodologies strongly diverge as a consequence of the dramatically different 3D point densities. In case of the TLS data, the point cloud is detailed enough to capture cylindrical shapes, therefore a set of disjoint primitives can be generated by means of the cylinder reconstruction method. In contrast, for ALS data the cylindrical shapes are not recognizable; the method must resort to the more elaborate scheme of generating an overcomplete, overlapping set of segments and finding a subset which forms probable linear structures, implementing the framework described in Sec. 3.1. The merging is conducted on the basis of collinearity and spatial overlap of the primitives. For the TLS data, additionally to the axis endpoints, radius estimates for the detected cylinders are available and may be used as auxiliary cues within the object similarity function which drives the clustering process. Finally, after merging a post-processing step is applied to refine the reconstructed stems and eliminate false positives. The common result of the both processing pipelines is the set of individual point subclouds, one for each segmented stem. The vectorized representations of the results depend on the data. For ALS point clouds, a set of 3D lines is derived, representing the central axes of the recovered stems. In case of TLS data, the stems are directly modeled by the recovered cylinders.

Point-level processing

Here, the point-level steps are presented which are common to both the ALS and the TLS processing pipeline. First, the DTM is generated from the raw point cloud. Since the objects of interest (fallen trees) are by definition located near the ground, a sufficiently fine DTM resolution must be used to ensure that the above-ground elevation of laser points can be accurately determined in possibly steep, mountainous terrain. The DTM filtering then consists in retaining points within a specified height interval over the DTM. However, in case of terrestrial point clouds care must be taken when considering points for removal due to the potentially high local differences in DTM

density and accuracy, which can be attributed to occlusions related to the terrestrial scanning perspective. Therefore, only points lying over a reliable portion of the DTM can be removed. In the last step, the posterior probability of belonging to a fallen stem is calculated for each filtered point, resulting in a probabilistic classification.

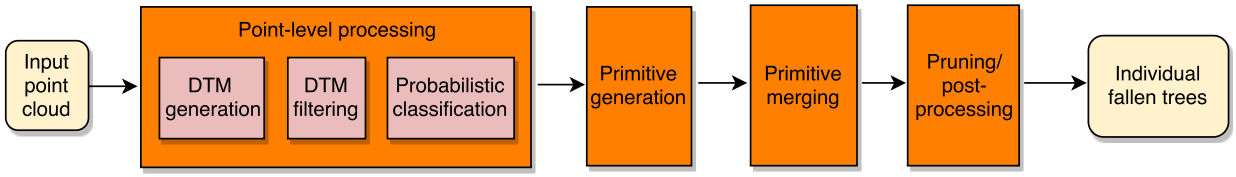


Figure 4.1: Processing pipeline for segmenting individual fallen trees. The point-level steps of DTM filtering and point classification are common for the ALS and TLS versions.

DTM generation

The process of generating the DTM starts with partitioning the area defined by the 2D bounding box of the input points into square cells of width d_{DTM} cm. For every cell c_k, the measured height H_k is defined as the minimum height of all points $p \in c_k$. It is assumed that outliers and spurious points (e.g. points below the ground, artifacts of the ringing effect) have already been filtered out at this stage. For calculating the DTM an Active Shape Model formulation is used [Elmqvist, 2002]. In this setting, the terrain model is an elevation function $z(p)$ defined over an evenly spaced grid. A surface energy can be associated with every $z(p)$:

$$E(z) = \sum_p E_{img}(p) + E_{int}(p) \tag{4.1}$$

In Eq. 4.1 the summing is over the discrete grid points. The term E_{img} refers to the image energy and usually is defined as sum of the distances between z and the measured heights on the grid. The term E_{int} is related to the energy of the surface. It introduces smoothness/rigidity constraints into the model. The formulation of the energy functional presented here is inspired by a similar approach in optical flow estimation. Following [Brox et al., 2004], a robust symmetric function $\phi(x^2) = \sqrt{x^2 + \epsilon}$ is utilized in the image distance metric:

$$E_{img}(p) = w_p \phi(|z_p - H_p|^2) \tag{4.2}$$

In the above, H_p denotes the measured height at grid point p. The small positive constant ϵ ensures convexity of ϕ. The role of the function ϕ is to reduce the influence of outliers, which would grow out of proportion under a purely quadratic distance penalty. Note the presence of an additional factor w_p. This term is introduced into the functional in order to include prior knowledge about the likelihood of the point p lying on the ground. This weight w_p is normalized on the interval $[0; 1]$ and can be obtained from an external ground point classification scheme or through manual labeling. If no such probabilities are available, uniform weights for all points can be used. For the internal energy term, once again the robust function ϕ is applied to the magnitudes of gradients of the elevation function z in both the horizontal and vertical directions on the grid:

$$E_{int}(p) = \alpha \phi(|\nabla_x z_p|^2 + |\nabla_y z_p|^2) \tag{4.3}$$

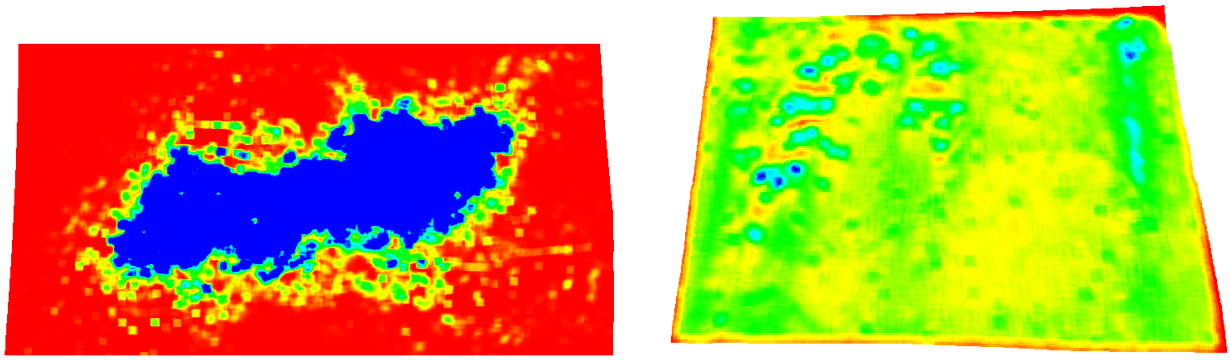
The parameter α controls the relative influence of the data and smoothness terms in the total energy. Putting (4.2) and (4.3) together, the total energy can be expressed as:

$$E(z) = \sum_p w_p \phi(|z_p - H_p|^2) + \alpha \phi(|\nabla_x z_p|^2 + |\nabla_y z_p|^2) \tag{4.4}$$

The DTM is obtained by minimizing the energy functional (Eq. 4.1) on the grid defined by the cells c_k with their associated measured heights H_k with respect to the model heights $z(p)$. The minimization is carried out using the iteratively reweighted least squares method. Since this is a highly non-convex problem, attaining a global minimum is not guaranteed. Therefore, multiple randomized re-runs are executed and the best solution is retained.

Density-based filtering

The objective of this step is to retain only input points having an above-ground height between h_{min} and h_{max}. However, the filtering procedure is slightly altered by taking into account the *confidence* of the local DTM measurements in order to compensate for incompleteness of the DTM acquired from a terrestrial perspective. Specifically, consider the DTM depicted in Fig. 4.2a, color-coded by ground point density. Due to the nature of the terrestrial scanner, the point density near the scanning positions is significantly higher compared to more distant parts of the scene. Also, as the distance from the scanner increases, it becomes less likely to capture ground points due to occlusions. Therefore it may happen that only the top part of a fallen stem is captured without its surrounding ground. In that case, the DTM based on the lowest point grid becomes unreliable in that location, and height filtering would lead to loss of relevant information. This problem is alleviated by calculating a point density map of the DTM and utilizing the smoothed local point density as a proxy for DTM confidence. A point is removed only if its normalized height is outside $[h_{min}; h_{max}]$ and the DTM local point density is above a threshold ρ_{DTM}. For ALS point clouds, this is less of an issue (Fig. 4.2b), since the points are acquired from an aerial perspective and the occlusions are mostly due to a dense tree canopy cover. Therefore, if a DTM patch contains few or no points, it is unlikely that the detailed structure of a fallen stem has been captured immediately above it.



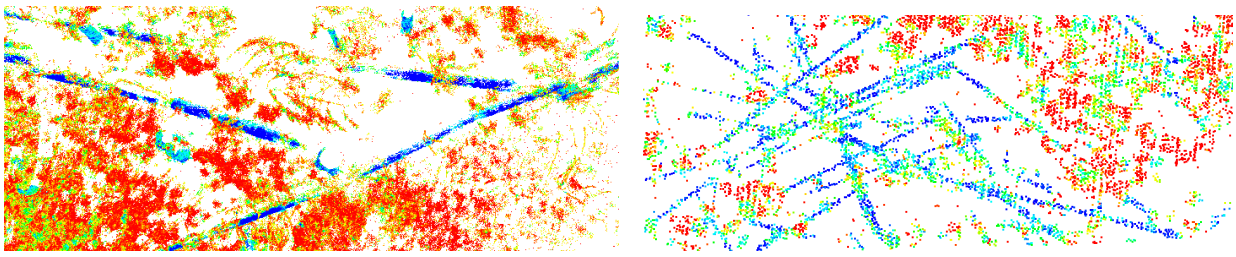
(a) (b)

Figure 4.2: DTM point densities of (a)TLS and (b)ALS point clouds. For TLS, the point density is markedly lower outside the area where the scanner was located, whereas for ALS the density is determined mostly by canopy cover. Red,green,blue colors indicate respectively low, medium and high density.

Probabilistic classification

After DTM filtering, the remaining points can belong to several object types found within the above-ground height interval. Aside from the target class of fallen stems, they could also represent e.g. shrubs, ground vegetation, forest regeneration, DTM artifacts etc. To find actual stem points, a probabilistic binary classification is carried out, with the classes representing (i) fallen stem points and (ii) all other objects. For each point, its posterior probability of belonging to the 'fallen stem' class given its feature values is calculated. The classification is based on two types

of geometric descriptors (Sec. 2.6): the Point Feature Histograms [Rusu et al., 2008] as well as the features derived from eigenvalues of the 3D structure tensor [Weinmann et al., 2015a], both utilizing a spherical neighborhood with fixed radius r_{3D}. These posterior probabilities then form the basis for subsequent processing at primitive (segment/cylinder) level. A color-coded probability map is shown in Figs. 4.3a, 4.3b respectively for TLS and ALS point clouds.



(a) (b)

Figure 4.3: Fallen stem probabilities for two forest scenes acquired with (a) TLS and (b) ALS. Red, green, blue colors indicate respectively low, medium and high probability.

4.1 ALS point clouds

Starting from the primitive (segment) level of processing, the methodology for reconstructing fallen stems in ALS point clouds is an instantiation of the linear structure segmentation framework proposed in Section 3.1. This section describes the concrete implementation of the application-specific parts of the framework, resulting in a complete fallen tree detection algorithm.

4.1.1 Segment generation

The generation of candidate segments is based on high-probability stem points obtained from the point-level processing stage. Specifically, all pairs of nearby points (distance below l_{seg}) with a posterior probability of belonging to a fallen stem above $p_{thr,p}$ are examined. Each point pair determines a segment in 3D, which can be regarded as the axis of a cylinder with fixed radius r_{seg} and fixed length l_{seg}. The segments are then filtered based on the properties of their corresponding cylinders. Let S be the set of DTM-filtered points which lie inside the target cylinder. The first condition requires that the cardinality of S be at least s_{thr} points. The second condition states that the average stem probability over points in S must not be below $p_{thr,p}$. Finally, the points in S should be evenly distributed along the entire length of the segment. To evaluate this criterion, all points in S are projected onto the segment, thus obtaining a 1-dimensional coordinate (position on the line) for each point. A histogram of the line positions is then formed, and segments for which at least $c_{thr,1}\%$ of the length is not covered by any points are removed from further processing. The exhaustive nature of the segment generation process results in a set of highly overlapping, redundant set of candidates (Fig. 4.4).

4.1.2 Contextual classification

To perform the contextual classification and selection of the optimal segments for merging, the framework requires the definition of concrete unary and pairwise potentials for the CRF, as well as the neighborhood relationship $\sim$. As for the latter, two segments s_i, s_j are considered neighbors, i.e. $s_i \sim s_j$, iff the center of segment s_j is inside a cylinder centered at the midpoint of s_i with length $l_{max} > l_{seg}$ and radius $r_{max} > r_{seg}$, whose axis contains s_i. Regarding the potentials, two

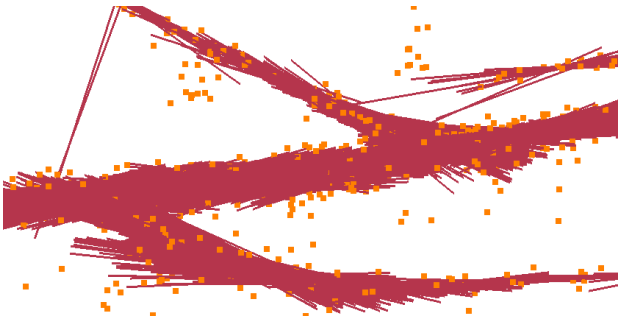
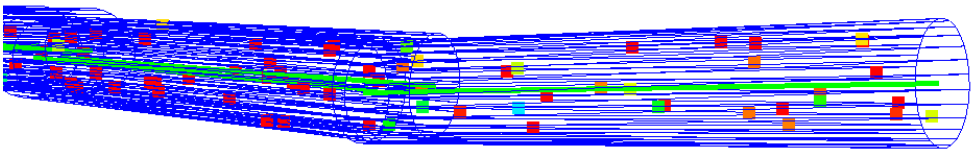


Figure 4.4: Segment candidates generated for a set of 3D points. Many segments overlap with each other due to the exhaustive manner of the generation procedure.

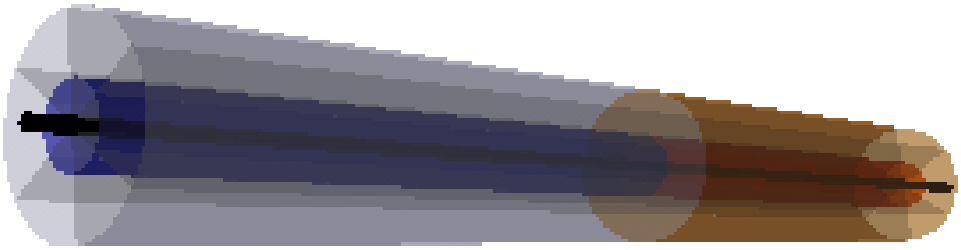
unary and two binary potentials are used [Polewski et al., 2017a], including the generic collinearity potential defined in Sec. 3.1.2:

Point-based appearance

The first of the unary appearance models, P_{app}, describes the geometric distribution of points around the target segment (Fig. 4.5a). It represents the posterior probability of the segment being part of a fallen stem given its 3D point distribution. For feature extraction, a cylindrical variation of the 3D Shape Context descriptor ([Frome et al., 2004], see Sec. 2.6) is used. The descriptor can be though of as a cylindrical volume with the segment of interest as its axis, partitioned into bins along 3 dimensions: radial, angular and axial (Fig. 4.5b). The features for classification are formed based on the histogram of the counts of laser points occupying the volume of each bin. The number of angular, axial and radial subdivisions is 6, 10 and 3, respectively, for a total of 180 features. The probability P_{app} is obtained from any appropriate binary classifier capable of providing probability estimates, e.g. logistic regression, random forest, or boosted decision trees.



(a)



(b)

Figure 4.5: Constructing 3D Shape Contexts around segments. (a) Cylindrical neighborhood around several segment candidates (green line) containing laser points. (b) a cylindrical 3D Shape Context with radial, axial and angular subdivisions around a segment.

Centrality

The second appearance model, P_{ctr}, is independent of the input point coordinates. Instead, it is based on the spatial configuration of other segments around the segment of interest. The intuition behind this is that in a cluster of neighboring segment candidates, the ones in the *center* will be a better approximation to the fallen stem points' skeleton, having a more similar orientation and situated more closely together than candidates lying on the outside (Fig. 4.6). Here, a point set's skeleton is understood as a polygonal chain which represents the 'axis' or 'central curve' of the points. This concept is discussed further in Section 4.1.3. To quantify the segment distribution, once again 3D Shape Contexts are used. The cylindrical volume around the target segment is considered, and all other candidates having a nonempty intersection with this volume are found. Then, points lying on the line defining each found candidate are sampled in equally spaced intervals. The 3DSC is applied to the set of all sampled points, resulting in the histogram features. Similarly to the case of P_{app}, the centrality model P_{ctr} is the discriminative class probability of the segment belonging to the stem skeleton, obtained from a binary classifier.

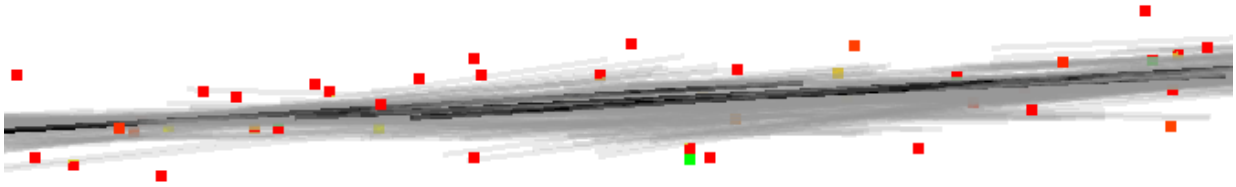


Figure 4.6: Bundle of highly overlapping segment candidates. The black segments lie close to the tree's skeleton, making them a better approximation of the stem than the remaining gray segments.

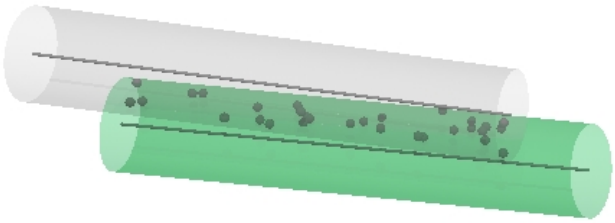


Figure 4.7: Estimating the cylinder intersection ratio.

Pairwise spatial overlap

The additional pairwise potential is derived from the spatial overlap between the cylindrical volumes around the segments s_i, s_j. It is meant to discourage the selection of segments which overlap strongly, so that the chosen segments are not concentrated in one location of the scene, but rather they are more evenly distributed along the stem axes. To estimate the ratio of volume $0 \leq r_{i,j} \leq 1$ shared by s_i and s_j as a fraction of a single cylinder's volume, a Monte Carlo-style simulation method is utilized. Let C_i, C_j denote the equally-sized cylindrical volumes respectively around segments s_i, s_j. If a point p is drawn with uniform probability from the interior of C_i, then the probability that p_C is inside the intersection of the two volumes $C_i \cap C_j$ is precisely $r_{i,j}$. Let $I_1^{i,j}, \ldots, I_R^{i,j}$ denote a sequence of random variables associated with R uniform draws of points $p_1, \ldots, p_R$ from C_i. Each variable $I_k^{i,j}$ is an indicator for the event that $p_k \in C_i \cap C_j$, i.e. assumes a value of 1 if the point p_k is inside *both* C_i as well as C_j, and a value of 0 otherwise. It is evident

that the $\{I_k^{i,j}\}$ are independent, identically distributed Bernoulli variables with probability of success $r_{i,j}$. To estimate the parameter $r_{i,j}$, the maximum likelihood estimator (see Sec. 2.1.5) is applied, which in this case has the functional form $\hat{r_{i,j}} = \sum_{k=1}^{R} I_k^{i,j}/R$. Because the expectation of $\hat{r_{i,j}}$ is equal to $E(\sum_{k=1}^{R} I_k^{i,j}/R) = Rr_{i,j}/R = r_{i,j}$, the estimator $\hat{r_{i,j}}$ is also unbiased. The idea behind calculating the intersection volume using a random simulation is depicted in Fig. 4.7. The pairwise CRF potential based on the overlap ratio assumes a triangular probability distribution $P_{ovp}(s_i, s_j) = 1 - r_{i,j}$ on the two segments being selectable together given $r_{i,j}$, with a probability of 1 and 0 respectively for disjoint and completely overlapping segments.

4.1.3 Merging

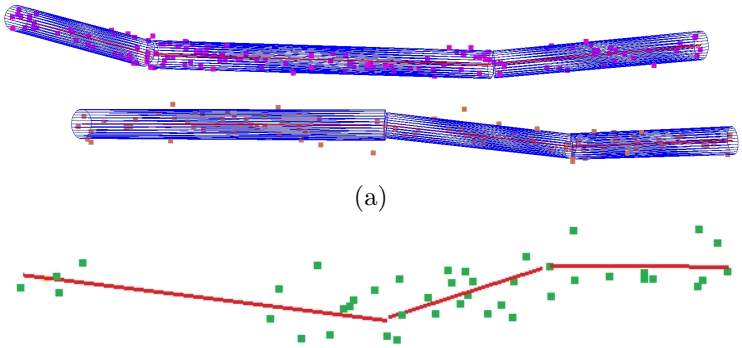
Regarding the merging step, it is necessary to specify the differential features quantifying segment similarity, as well as a stopping criterion for the Ncut segmentation. Moreover, to make use of the learning-based approach to feature weight determination in the Ncut similarity function (Sec. 3.1.3), a source of labeled segment pairs must be provided which yields both positive and negative examples, i.e. pairs of segments which belong to the same vs. different fallen stems. In the following, these required components of the merging process are introduced and characterized. Specifically, it is shown how to automatically obtain labeled samples for learning the similarity function and stopping criterion by means of a physical simulation which generates virtual scenes containing overlapping stems [Polewski et al., 2014, 2015b].

Features for merging

The generic differential features enumerated in Section 3.1.3 are augmented with the spatial overlap ratio $r_{i,j}$ introduced in Section 4.1.2. To maintain compatibility with the remaining features which are interpreted as a *distance*, the actual feature value for two segments s_i, s_j is defined as $1 - r_{i,j}$, resulting in a distance of zero for fully overlapping segments, and a distance of 1 for a completely disjoint pair. The neighborhood relation $\sim$ which induces the structure of the input graph and similarity matrix for merging is reused from the contextual classification step (Sec. 4.1.2).

Stopping criterion

As noted in Section 3.1.3, within the proposed framework the stopping criterion for the recursive two-way Ncut algorithm is viewed as a binary classifier which decides whether the scene contains exactly one vs. more than one target object. In the setting of fallen tree segmentation, the goal is to build a classifier which recognizes whether a cloud of points represents a single fallen tree. The model should express the intuition that a fallen tree consists of several connected cylindrical parts with approximately equal radii within the 10-50 cm range. To achieve this, the k-segment polyline skeleton of the point cloud is calculated. The skeleton can be regarded as a simplified representation of the point cloud, consisting of linear segments that coincide with the central axis of the object. The specific algorithm for calculating the polyline skeleton is given in the next section. Each skeleton segment is regarded as the axis of a cylinder, whose radius is estimated by taking the qth quantile of the distribution of orthogonal distances of points around the segment (Fig. 4.8a). To discourage connections of two unrelated point groups by a segment which is very sparsely populated with points (see Fig. 4.8b), the estimated linear occupancies of each skeleton part are also included into the model. The linear occupancy can be seen as an approximation of the percentage of the length of the segment which is covered by its projected surrounding points (see Sec. 4.1.1). The final set of features for classification, containing the sorted k radii and k linear occupancy ratios, is augmented by the standard deviation of the radii and the dimensions of the oriented bounding box of the input point cloud part (obtained by applying principal components



(a)

(b)

Figure 4.8: Visualization of tree skeletons. (a) cylinder-based representation of tree skeleton. (b) example of skeleton segment which connects unrelated point groups.

analysis to the covariance matrix of 3D coordinates). Moreover, thresholds are defined on the minimum linear occupancy ratio and on the maximum cylinder radius. If the minimum linear occupancy ratio among the k segments is below $c_{thr,2}$, or the maximum radius is above r_{thr}, the partitioning process is continued. Otherwise, the decision is made by the classifier.

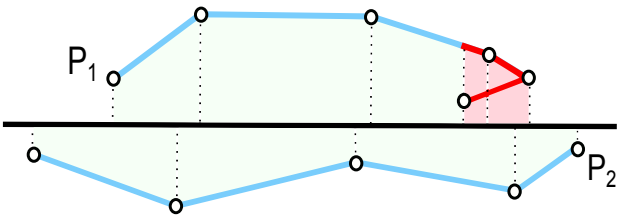


Figure 4.9: Two polylines P_1 and P_2. P_1 is not feasible, because its segments overlap near the right end. P_2 is feasible: its points can be uniquely mapped to a straight line.

Skeletonization

In this thesis, skeletonization of a set of points $Q = (q_1, \ldots, q_N)$ is understood as the task of finding a piecewise linear curve in 3D (the skeleton) which best fits the points in the sense of the sum of point-to-curve distances. For fallen tree detection, obtaining such a polyline for a point cloud representing a fallen stem is an essential step towards simplification and building a compact representation of the stem. The bendpoint coordinates of the calculated polylines form the basis for a precise comparison between detected and reference trees, and also allow the formulation of a fallen tree appearance model. The case when the skeleton consists of a single segment boils down to fitting a 3D line to the points Q. This is a well studied problem and can be accomplished using orthogonal distance regression (ODR) [Leng et al., 2007]. This technique relies on the fact that the centroid of the point set necessarily belongs to the ODR line and the line's direction corresponds to the eigenvector associated with the largest eigenvalue of the point cloud's covariance matrix $\sum_i (q_i - c)(q_i - c)^T / N$, where c is the centroid of Q. However, it is desirable for the method to handle stems that consist of multiple linear fragments which intersect at nonzero angles. Therefore, a piecewise linear curve (polyline), where each segment shares an endpoint with its predecessor in the chain, is an appropriate model for describing the fallen tree. The problem of fitting a polyline to a general set of points is quite computationally expensive (though solvable in polynomial time) even under a vertical distance criterion [Aronov et al., 1998]. Therefore, the class of valid polylines is restricted to those that do not bend back on

themselves or form zigzags (see Fig.4.9). More formally, the algorithm focuses only on polylines given by endpoints $p_1, \ldots, p_M$ for which there exists a line L such that the sequence of coordinates $t_1, \ldots, t_M$ of the points p_i projected onto L is strictly increasing. Such polylines are referred to as *feasible*. A dynamic algorithm is proposed for calculating an approximately optimal feasible polyline for the input set of points and a predefined number of segments k (Alg. 6). The optimized criterion is the sum of squared orthogonal distances of all points in the cloud to the polyline (the closest segment). The algorithm is based on the idea of reducing the original task of fitting the polyline to the problem of one-dimensional clustering and applying the classical dynamic programming method of Bellman [1973]. This reduction is done by sorting all the input points by their projected position on line L and considering each contiguous sequence of points as a potential cluster. Each cluster is then defined by its start end end points. Since the algorithm is searching for a clustering that will cover the entire input set of points, it is the case that for each end point, its immediate successor is the start point of the next cluster. Therefore, it is enough to find the $k-1$ end points that give rise to the clustering with the minimum sum of inter-cluster discrepancies. The quality measure of each individual cluster is based on the sum of squared orthogonal distances to the best-fit 3D line obtained via ODR. Note that all parts of the polyline except the first one have an additional constraint: they need to start at the endpoint of their immediate predecessor. The ODR procedure is slightly altered to account for this fact. Let p_p denote this constraining endpoint for the k-th skeleton part, $k > 1$. Instead of deriving the k-th line direction from the usual covariance matrix, an analogous matrix $B = \sum_i (q_i - p_p)(q_i - p_p)^T$ centered on p_p is used in this role. The line L which constrains the polyline coordinates and gives rise to the ordering of points is taken to be the ODR line through the entire point set (i.e. the one-segment skeleton). The total computational complexity is $\Theta(kN^3)$, which can be improved to $\Theta(N^3 + kN^2)$ by precomputing the line directions and total squared distances for all $\Theta(N^2)$ contiguous intervals of the input points' line position range.

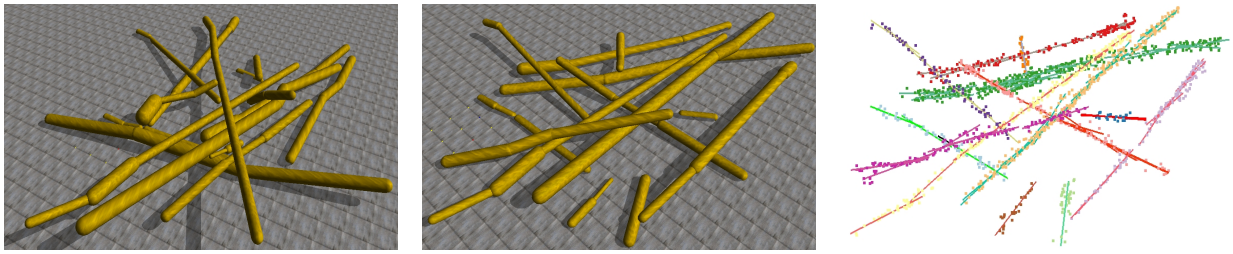
Simulating fallen tree scenarios

In the following, a virtual fallen stem generation pipeline based on physical simulation is introduced, inspired by a method for synthesizing virtual pedestrian appearance and shape samples in raster images described by Enzweiler & Gavrila [2008]. There are two goals of this procedure. The first one is to train a classifier to distinguish between pairs of segments which belong to the same stem versus pairs which belong to different stems. This classifier will be used as the Normalized Cut similarity function. The second goal is to train another classifier to become the segmentation stopping criterion as described in Section 4.1.3. The simulation requires a set of stem prototypes as input. They are given in the form of their individual point clouds. Each prototype is converted into a cylindrical representation (Fig.4.8a) by first computing its polyline skeleton (see previous section), and subsequently determining the maximum distance from the skeleton to any of its points. The cylinders define the physical behavior of the fallen stems in the simulation. Once the prototype cylinders are ready, the simulation proceeds iteratively by randomly picking a prototype, materializing it at a certain height above the ground plane and subjecting it to the force of gravity. Spatial occupancy and rigidity constraints applied to the cylinders combined with gravity and friction allow for quite complex interactions between the stem models, resulting in multiple overlapping stems lying on the ground in several layers (Fig. 4.10a, 4.10b). The simulation is carried out using the Open Dynamics Engine [Smith, 2007].

The translations and orientations defined by the final positions of the individual trees can then be applied to the point sets and segments associated with the corresponding tree prototypes (Fig. 4.10c). New positive examples (segments belonging to the same stem) are created by finding, for each tree copy, a randomized subset of segments whose projections onto the skeleton approximately cover most of the skeleton's line intervals. Specifically, the set of all generated

Algorithm 6 Fit polyline to set of points

function BestLineFit($p_1 \dots p_M$,p_{line})
  $U \leftarrow \begin{pmatrix} p_1^T - p_{line}^T \\ \vdots \\ p_M^T - p_{line}^T \end{pmatrix}$
  $B \leftarrow U^T U$
  $v \leftarrow$ *eigenvector of* B *paired with largest eigenvalue*
  $totalSqDist \leftarrow 0$
  for $i = 1..M$ **do**
    $d \leftarrow$ *distance from point* p_i *to line* $L(p_{line}, v)$
    $totalSqDist \leftarrow totalSqDist + d^2$
  return $(totalSqDist, v)$
function BestPolylineFit($q_1 \dots q_N$,k)
  $c_q \leftarrow Centroid(q_1 \dots q_N)$
  $(s_q, v_q) \leftarrow BestLineFit(q_1 \dots q_N, c_q)$
  $(p_1 \dots p_N) \leftarrow q_1 \dots q_N$ *sorted by projected line position on* $L(c_q, v_q)$
  $(t_1, t_N) \leftarrow$ *projected positions of* $p_1 \dots p_N$ *on* $L(c_q, v_q)$
  $C \leftarrow [-1]_{0..k \times 0..N}$ ▷ C stores the optimal partial skeleton costs
  $P \leftarrow [(0,0,0),(0,0,0)]_{0..k \times 0..N}$ ▷ P stores the optimal line coefficients
  $C[0,0] \leftarrow 0$
  for $i = 1..N$ **do**
    $c_i \leftarrow Centroid(p_1 \dots p_i)$
    $(s_i, v_i) \leftarrow BestLineFit(p_1 \dots p_i, c_i)$
    $C[1,i] \leftarrow s_i$
    $P[1,i] \leftarrow (c_i, v_i)$
  for $l = 2..k$ **do**
    for $i = 1..N$ **do**
      $C[l,i] \leftarrow C[l-1,i]$
      $P[l,i] \leftarrow P[l-1,i]$
      for $j = i..2$ **do**
        if $C[l-1,j-1] \geq 0$ **then**
          $p_{end} \leftarrow projectOntoLine(p_j, P[l-1,j-1])$
          $(s_j, v_j) \leftarrow BestLineFit(p_j \dots p_i, p_{end})$
          if $s_j + C[l-1,j-1] < C[l,j]$ **then**
            $C[l,j] \leftarrow s_j + C[l-1,j-1]$
            $P[l,j] \leftarrow (p_{end}, v_j)$
  return *sequence of breakpoints corresponding to optimal path from* $C[k,N]$ *to* $C[0,0]$, *reconstructed from* C *and* P



(a) (b) (c)

Figure 4.10: Simulating fallen stems. (a) and (b) depict randomly generated scenarios, (c) shows the ALS points transformed according to the rotations and translations of the stems in scenario (b).

segment candidates containing at least 1 point from the tree copy is considered. Then, segments from this set are randomly picked and projected onto the skeleton. The procedure is repeated until the unary sum over the projection intervals on the skeleton covers at least r_c of the total skeleton length. This can lead to new spatial configurations of segment pairs. Negative examples are obtained by taking segment pairs from two adjacent stems. The top-level simulation loop for learning the similarity function is given by Alg. 7. The training set S is initialized with positive and negative examples from a randomly generated stem scenario, and the initial classifier c is trained on S. An arbitrary classifier capable of providing a probability value along with the class label may be used. The algorithm then iteratively generates new stem scenarios, and classifies all newly created examples with c. For each classified element, its class probability is examined. A value close to 0.5 (distance below $pThr$) indicates a high uncertainty of the classifier's decision, which means that the example contains information about the decision boundary and causes it to be appended to the training set. At the end of each iteration, c is retrained on the augmented training set. This approach can be viewed as a simple implementation of the active learning paradigm, specifically the uncertainty sampling variation (see Sec. 2.1.3), where the classifier's uncertainty about a sample's label is taken as a measure of that sample's utility within the learning process. After executing a predefined number of iterations, the algorithm yields the probabilistic classifier c, which can be applied as the similarity function for Normalized Cut.

Algorithm 7 Learning segment pair classifier

```
function LearnClassifier(pThr)
    trainSet ← createSampleScenario()
    classifier ← trainClassifier(trainSet)
    for i = 1..N_iter do
        auxSet ← createSampleScenario()
        for o ∈ auxSet do
            p(o) ← getContinuousDecision(classifier,o)
            if p(o) > pThr ∧ p(o) < 1 − pThr then
                trainSet ← trainSet ∪ o
        classifier ← trainClassifier(trainSet)
    return classifier
```

For learning the segmentation stopping criterion, the procedure is somewhat simpler (Alg. 8). Random fallen tree scenarios are repeatedly generated and the entire Normalized Cut procedure is carried out on each generated training set. However, a special supervised criterion is used in order to stop the partitioning. The labels of all the segments (known by construction) in the

currently considered cluster are checked and the procedure is terminated only if all the labels are equal. At each step, the features of the point cloud (described in Sec. 4.1.3) associated with the skeleton are calculated and recorded as training data together with the decision to stop or continue partitioning. As a result, a labeled dataset is obtained which is subsequently used for training a classifier.

Algorithm 8 Learning Stopping Criterion

procedure LEARNSTOPPINGCRITERION
  for $i = 1..N_{iter}$ **do**
    trainSet $\leftarrow$ *createSampleScenario()*
    normalizedCut(trainSet, StopSupervised)
function STOPSUPERVISED(*segments,points,segLabels*)
  $l \leftarrow \bigcup_{s \in segments} segLabels[s]$
  $stop \leftarrow |l| = 1$
  recordExample(calculateSceneFeatures(points), stop)
  return *stop*

4.1.4 Scene ranking

The details of ranking segmentation results of fallen tree scenes are discussed in this section. This is the instantiation of the final step belonging to the linear structure detection framework (Sec. 3.1.4). The specific tree appearance models for assessing the quality of single fallen stems are presented, along with the definition of the scoring function which aggregates the single object evaluations to form a single score value assigned to the scene.

Partition into networks

To simplify computation, the entire input set of segment candidates is partitioned into separate networks, based on the connected components of the graph induced by the segment neighborhood relation $\sim$ (Sec. 4.1.2). Each network can be then processed independently of the others, which enables parallel calculations. An example of partitioning a set of segments into networks is depicted in Fig. 4.11. To eliminate networks which are not likely to contain actual fallen stems, a network-level classification is conducted. The used features are simple, top-level characteristics of each network: number of edges and nodes in the network's connected component, number of laser points belonging to its segments, the lengths of the points' oriented bounding box (obtained via principal component analysis), the planimetric area of the bounding box's projection onto the ground, as well as the mean stem point probability over all the laser points and the mean segment appearance probability P_{app} over all the segments within the network. A binary classification problem is formulated, with the positive class representing a grouping of probable fallen trees, and the negative label associated with random clusters of ground vegetation, DTM artifacts and other irrelevant objects. In contrast to the stem point and segment level classification, here the posterior class probability is not necessary - only a binary decision is required, admitting the use of any state-of-the-art classification method. Networks receiving a negative classification are removed from further processing.

Scene aggregate score

A segmentation result consists of labeled point subsets corresponding to the detected fallen stems. First, skeletonization of the point sets is performed, yielding a piecewise linear curve for each stem (Fig. 4.8b). The score $s(R)$ of a segmentation result (scene) $R = \{t_i\}, i = 1..n_r$ consisting of n_r

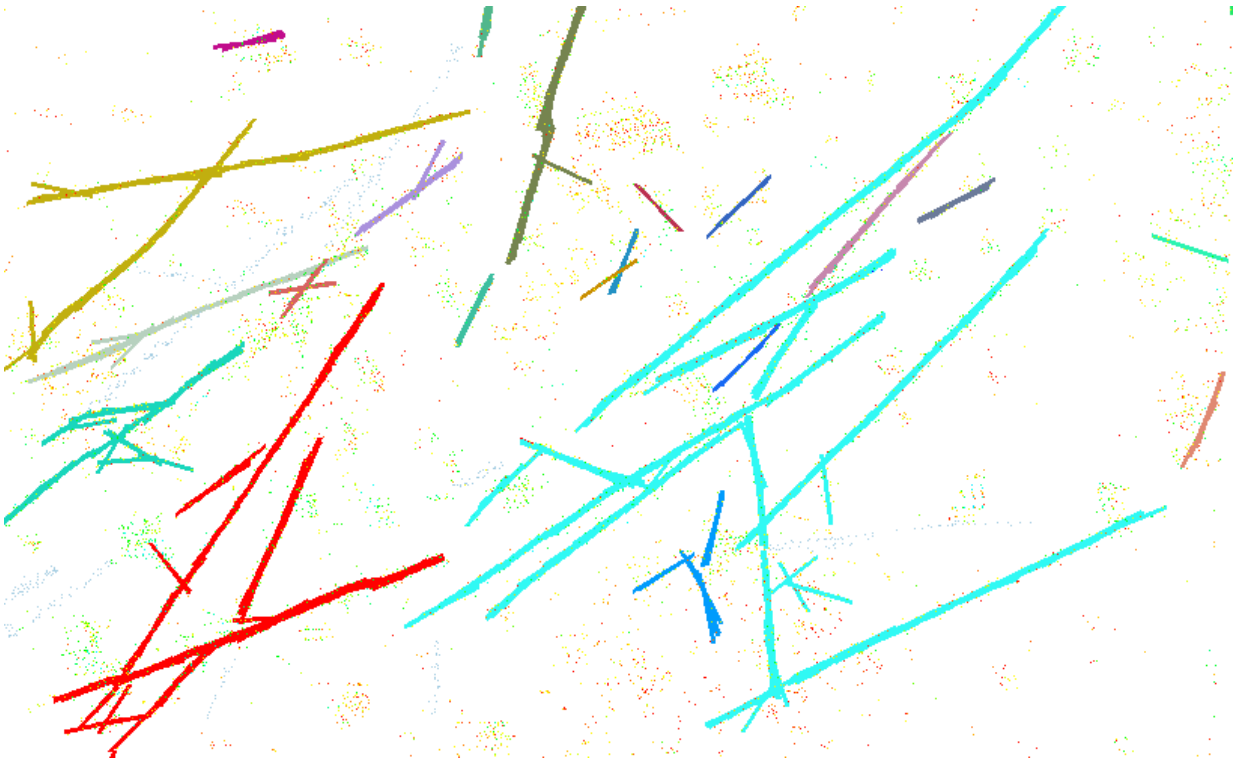


Figure 4.11: Set of segment candidates partitioned into disjoint networks, represented by different colors. Each network is independent of the others and can be processed separately.

detected trees t_i is defined as the sum of two terms: $s(R) = s_l(R) + \kappa s_c(R)$. The term s_l denotes the total detected tree length. It is calculated by iterating over all line segments l_i^j belonging to the skeletons of detected trees t_i. Each fallen tree part represented by line l_i^j is classified based on its appearance (see below) as either correct or wrong. In the former case, the summand s_l is increased by the length of l_i^j. The term s_c indicates the number of detected trees t_i with at least one member line segment l_i^j classified as correct. The coefficient κ expresses the tradeoff between assigning a high score to scenes with many (possibly short) trees, versus placing value on the total detected length regardless of the tree count. A default value of $\kappa = 1$ is assumed.

Tree appearance models

The scoring scheme utilizes the tree part appearance model, whose task is to classify a contiguous linear segment of the detected tree, corresponding to one part of its skeleton. The classification is binary: the positive label corresponds to proper parts of a fallen tree, and the negative class encompasses all other object types. The appearance is evaluated based on the properties of laser points and segment candidates around the target skeleton part. Specifically, the following core features are used: part length, number of associated segment candidates and laser points, mean stem point probability and segment probability (Sec. 4.1.4). Also, in analogy to the procedure described in Section 4.1.1, a linear occupancy histogram for the skeleton part is calculated on the line positions obtained from projecting the laser points onto the skeleton segment. A low or unevenly distributed coverage with large uncovered extents along the line are usually indicative of an erroneously detected tree. A similar procedure is performed at segment level, where candidate segments are projected onto the skeleton, defining covered intervals on the line. The resulting length coverage ratios at both point and segment level are the two final features. Finally, at the end of the processing pipeline an additional object-level classifier is applied to remove spurious

detected objects which are not likely to be true fallen trees. This entire tree model describes the neighborhood of the whole skeleton of a detected tree as opposed to the part model, outlined above, which focuses only on one of its segments. The tree model expands upon the part model by introducing the mean and standard deviation of part lengths, the angles between consecutive skeleton segments as well as the dimensions and volume of an oriented bounding box containing the tree's points. Once again, no posterior class probability output is required, only a binary decision, which admits the use of a wide range of classifiers in the role of tree appearance models.

4.2 TLS point clouds

In contrast to the case of ALS point clouds, the point density in TLS data is usually high enough to preserve detailed geometric shapes within the captured scene, at least in parts of the scene which are reasonably close to the scanning position and are free from occlusions. Therefore, the tree detection pipeline from TLS data makes use of cylinders as the intermediate representation for constructing primitives. Moreover, the cylinders which can be extracted from the dense point cloud data usually have a significantly lower degree of overlap compared to the segments generated from ALS data. Also, for the same forest scene the number of cylinders obtained from dense data can be 2-3 orders of magnitude lower compared to the sparse point cloud, since they are not generated exhaustively. This means that an explicit step of selecting representative primitives from an overcomplete pool can be omitted. The cylinder extraction step instantiates the cylindrical shape detection framework described in Section 3.3. It is preceded by a connected component segmentation which decomposes the scene into disjoint parts, to reduce computational complexity. In agreement with the general workflow depicted in Fig. 4.1, after primitive generation the cylinders are merged to form individual trees, and finally postprocessed to eliminate redundancy between potentially intersecting cylindrical parts.

4.2.1 Connected component segmentation

Although the cylinder extraction procedure could be applied to the entire set of high-probability stem points available after the DTM filtering and point classification steps, this could result in an overwhelming number of local maxima within the cylinder parameter space. Such a situation could lead to difficulties in locating all the local maxima and estimating the correct KDE bandwidth which would guarantee to not smooth out adjacent peaks within the estimated probability density. However, this problem can be mitigated by applying connected component segmentation on the set of high-probability (above $p_{thr,p}$) stem points, with connectivity defined by a spherical neighborhood with fixed radius r_n. When r_n is small enough, this procedure is expected to break down the scene into clusters that contain no more than 3-4 distinct cylinders each, thus simplifying the primitive extraction in the next step. The obtained clusters are then filtered based on a minimal member point count n_{min} and minimal length of the longest dimension of the cluster's oriented bounding box l_{min}. The remaining connected components usually correspond to fallen or standing tree stems and, occasionally, DTM artifacts (Fig. 4.12). Each cluster is processed individually by the cylinder detection framework.

4.2.2 Cylinder detection

The method described in Section 3.3 is applied to reconstruct the cylinders associated with each connected component. The parameter voting weights described in Sec. 3.3.5 are obtained from each point's posterior probability of belonging to a fallen stem. The application-specific filtering of results expresses some prior knowledge about the appearance and dimensions of fallen stems within TLS point clouds and is executed as follows. First, any cylinder C_i with too few inlier

Figure 4.12: Connected components of high-probability stem points (random colors). Elongated structures represent fallen trees, smaller clusters are parts of standing trunks.

points, i.e. $|I(C_i)| < n_{min}$, is removed. Also, a minimal cylinder length threshold l_{min} is set. Recalling that $\boldsymbol{d_i}$ denotes C_i's axis vector, the reference vector z_i can be interpreted as C_i's unit 'upward' direction, defined as the vector lying on the plane spanned by $\boldsymbol{d_i}$ and the world Z axis, perpendicular to $\boldsymbol{d_i}$ and having a positive Z coordinate (see Fig. 4.13). Then, the projected coordinate $a_{i,j}$ of an inlier point $p_j \in I(C_i)$ expresses the signed angular deviation of p_j's projection onto C_i from the 'top' of the cylinder (see Sec. 3.3.7). This results in a useful characterization of a point's position on the cylindrical surface, and can be used to express the intuition that points captured from a terrestrial perspective should generally lie on the top and side portions of the fallen stem, but not on the bottom (since it is not visible). Such a filter can remove certain DTM artifacts such as concavities in the ground which may exhibit a cylindrical shape. To formalize this intuition, the distribution of all angular coordinates $a_{i,j}$ of inlier points is analyzed, and the fraction of points having $a \in [-120°; 120°]$ is calculated. This angular range corresponds to points lying on the upper two-thirds of the cylindrical surface, i.e. the part normally visible from the TLS sensor's perspective. Cylinder hypotheses where the fraction of points in the valid range is below a threshold r_{top} are removed. Finally, the cylindrical surface is partitioned into uniform bins in the $l \times a$ space of projected coordinates. The surface cover ratio on this grid is calculated as the percentage of bins which contain at least one projected inlier point. Cylinders whose surface is covered to a degree of below r_{cov} are removed.

4.2.3 Merging

Similar to the case of ALS data, the Normalized Cut algorithm (Sec. 2.4) is used for merging the reconstructed cylinders into individual stems. However, this time the primitives (cylinders) are of much higher quality compared to the segments derived from ALS data. The cylinders already form large portions of the fallen stems, with considerably less overlap. Nevertheless, single stems may still be represented by multiple cylinders when occlusions during point cloud acquisition occur or the stems actually break down into disjoint, potentially non-collinear parts. Once again, the differential features introduced in earlier sections, augmented with the difference in radii, can be re-used for constructing the similarity function:

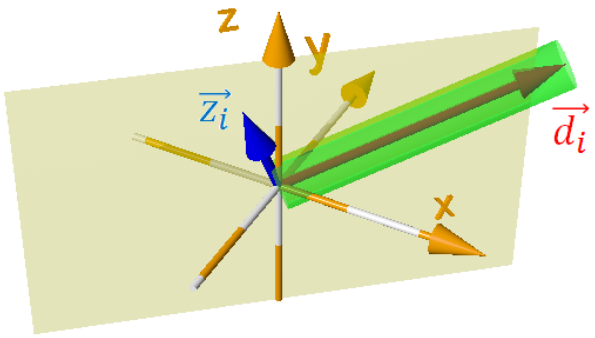


Figure 4.13: Cylinder i's reference vector z_i for computing the angular coordinate of projected inlier points.

- *difference in heading*: defined as the smaller of the two angles $\angle(\boldsymbol{d}_i, \boldsymbol{d}_j), \angle(-\boldsymbol{d}_i, \boldsymbol{d}_j)$, in analogy to d_{dir} from Section 3.1.2
- *mean axis distance*: the mean distance between the axes of C_i and C_j, computed by projecting equally spaced points from one axis onto the other and averaging the result, as described in Section 3.1.2 (d_{seg}) and depicted in Fig. 3.2
- *spatial overlap*: the ratio of the volume shared by cylinders C_i, C_j to the volume of the smaller cylinder, calculated as in Section 4.1.2 using Monte Carlo simulation
- *radius deviation*: the difference $|r_i - r_j|$

As before, each feature is associated with its weight σ_k which controls its influence within the multiplicative-exponential similarity function given by Eq. 2.34. Considering the aforementioned higher quality of primitives for merging, it is usually relatively easy in practice to find a combination of feature weights σ_k which yields good segmentation results. The same is true of an Ncut threshold value for the segmentation stopping criterion. Therefore, it is not necessary to learn the similarity function and/or stopping criterion from data in this case, and instead a coarse grid search and sensitivity analysis w.r.t. $Ncut_{thr}, \{\sigma_k\}$ suffices for this purpose.

4.2.4 Postprocessing

Each segmented tree obtained from the previous step consists of one or more potentially overlapping cylinders. This overlap could have a detrimental effect on the accuracy of estimating stem volume as a sum of the volumes of the constituent cylindrical parts. To remedy this, an algorithm is introduced for adjusting the reconstructed cylinder lengths in such a way that no two cylinders occupy the same space along the tree's axis. This algorithm is based on the ideas introduced in Sec. 4.1.3, related to the skeletonization of a set of 3D points representing a fallen stem. There, it was proposed to approximate the stem skeleton as a piecewise linear curve with vertices along a line L. By projecting the stem's member points onto L, a linear ordering of the points is obtained. This in turn makes the problem amenable to one-dimensional clustering methods for partitioning L into disjoint regions corresponding to input cylinders, similar to the original algorithm for skeletonization (Alg. 6). Specifically, consider a detected tree T consisting of the cylinders $(C_i)_{i=1..N_c}$ and let P_T be the set of all inlier points of (C_i). First, a line L is fitted, in the least-squares sense, to P_T and the projected coordinates for each point are obtained. Then, the points P_T are sorted according to their position on L. The problem of cylinder redundancy elimination can be reduced to assigning intervals of L to cylinders (C_i) such that (i) no two intervals overlap (ii) each point

Algorithm 9 Cylinder redundancy elimination

function CylinderDistance(*cyl*,*from*,*to*)
  return ▷ the sum of distances between cylinder *cyl*
    and all points with a line position ∈ [*from*;*to*]
function EliminateRedundancy(*points*,*cyls*)
  $L \leftarrow$ least-squares fitted line to *points*
  $p \leftarrow$ line positions of *points* projected onto L
  sort p in ascending order
  for i=1..$|p|$ **do**
    $S[1,i] \leftarrow \min_{c \in cyls}$ *CylinderDistance(c,0,i)*
  for k=2..n_c **do**
    for u=u_0..$|p|$ **do** ▷ $\{p[u_0] - p[1] \geq k \cdot l_{min}\}$
      $S[k,u] \leftarrow S[k-1,u]$
      for v=$v_0 \downarrow 1$ **do** ▷ $\{p[u] - p[v_0 - 1] \geq l_{min}\}$
        $t \leftarrow \min_{c \in cyls}$ *CylinderDistance(c,v,u)*
        $S[k,u] \leftarrow \min(S[k,u], t + S[k-1, v-1])$
  return sequence of non-overlapping cylinders along L based on $S[n_c, |p|]$

in P_T is covered by exactly 1 interval and (iii) the sum of point-to-cylinder distances over P_T is minimal. This problem can be expressed as a one-dimensional clustering task and is solvable using a dynamic algorithm [Bellman, 1973], as in the case of computing the polyline skeleton. The key property is that this task exhibits optimal substructure w.r.t. the number of chosen intervals. Let $S[k,u]$ denote the optimal sum of point-cylinder distances for line positions 0 to u attainable with exactly k intervals. Then, $S[k,u]$ can be recursively expressed in terms of the optimal solution for $k-1$ intervals: $S[k,u] = S[k-1, v-1] + \min_{c \in (C_i)}$ *CylinderDistance(c,v,u)* for some $v < u$. This gives rise to the dynamic algorithm presented in Listing 9. An additional constraint on the minimal interval length l_{min} is imposed to eliminate fragmentation. Based on the resulting values $S[\,\cdot\,,\,\cdot\,]$, it is possible to obtain the interval ends and associated cylinders corresponding to the minimal cost assignment. Also, since there is a well-defined ordering of the cylinder endpoints, gaps in the stem, i.e. segments of L with no projected point, can be identified and the missing parts completed by inserting 'hallucinated' cylinders connecting the endpoints on both sides of the gap. The sequence of chosen original and artificially inserted cylinders along the axis L is considered the final detected tree.

5 Detection of standing dead trees from aerial imagery and ALS data

This chapter is concerned with the detection and delineation of standing dead trees based on two complementary data sources: ALS point clouds and aerial multispectral raster imagery. While the former provides 3D structure information, the latter is especially useful for the detection of dead or damaged vegetation due to the presence of a near infrared (NIR) spectral band, where the decreased levels of chlorophyll pigments are clearly reflected. Specifically, three scenarios are considered here. First, the case is discussed when solely multispectral imagery without any 3D information is available. The method attempts to find clusters of pixels with a hue corresponding to dead tree crowns, based on values learned from training data. In the second step, an active contour segmentation with priors on shape and pixel colors is applied to the recovered clusters in order to find precise contours of individual dead trees. The next two scenarios assume the availability of ALS point clouds. Because of the vertical orientation of standing dead trees, any generic method for individual tree crown segmentation from point clouds can be applied to all trees within a plot and each segmented tree can then be classified as either dead or living. In the second scenario, the multispectral raster data are available along with the 3D point cloud data, which allows for the combination of 3D structure information with the predictive power of the NIR band. The proposed approach is to project the ALS points belonging to each individual segmented tree onto the image plane, using the standard collinearity equation and the camera orientation data associated with the aerial imagery. Then, a 2D bounding polygon of the projected points may be calculated, yielding a set of image pixels representing the specific tree. Features derived from the multispectral signatures of the relevant pixels serve as a basis for classification. This approach is suitable for dead trees which have retained enough of their crowns to produce a sufficient signature within the aerial image. The third and final scenario concerns the situation where this is not the case, i.e. only the dead stem is left standing, whereas the crown has already decomposed. In such case, the ground resolution of aerial imagery is usually not sufficient to capture the remaining tree stump, whose cross-section might be only represented by 1-2 pixels, rendering any image-based detection method infeasible. The proposed method therefore uses exclusively information from the point cloud, and focuses on developing a generalized version of the 3D Shape Context (see Sec. 2.6), resulting in a shape descriptor tailored to the appearance of standing dead trees which exhibits an increased discriminative capability with respect to this object class. It is demonstrated how to efficiently optimize the parameters of this descriptor for a particular training set using a genetic algorithm (GA).

5.1 Detection from aerial imagery using active contours

The method for detecting individual snags (standing dead trees) accepts single color infrared images as input. The number of spectral bands is not fixed, however the assumption is that the image contains bands such that the joint distributions (over all bands) of values of pixels belonging to dead trees versus pixels belonging to other objects are significantly different. A sample color infrared image of a forest scene containing standing dead trees is shown in Fig. 5.1, where the differences between the pixel colors of dead (gray) and living (red) vegetation are clearly visible. No image correspondence or 3D elevation information is necessary, but the image scale (pixel resolution at ground) is assumed to be known. Furthermore, a set of training samples must be available in the form of images with polygons delineating individual snags. The training data is the basis for deriving shape and intensity prior information. The entire algorithm can be conceptually divided into two major steps. The first step uses the intensity prior information to find regions in the image which are likely to contain snags. In the second step, a fine segmentation is iteratively applied locally at different points of the image which cover the high-likelihood snag regions. This fine segmentation uses both intensity and shape priors. The output of the algorithm is a set of Boolean pixel masks (or equivalently, contours) which specify the sets of pixels of the original image that define each individual detected snag. An overview of the detection pipeline is shown in Fig. 5.2. The details of the processing steps are given below.

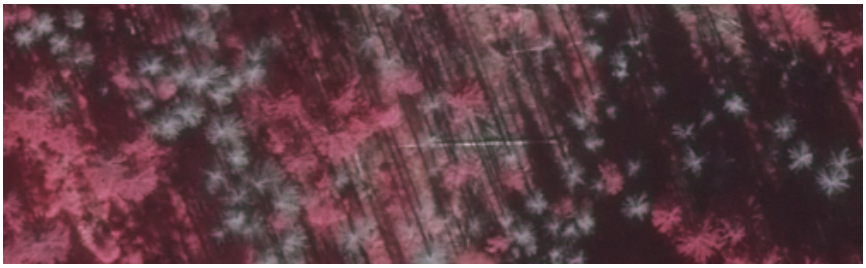


Figure 5.1: Multispectral image (NIR, red and green channels) of forest scene with dead trees. Dead vegetation appears in gray colors, contrasting with red hues for living vegetation.

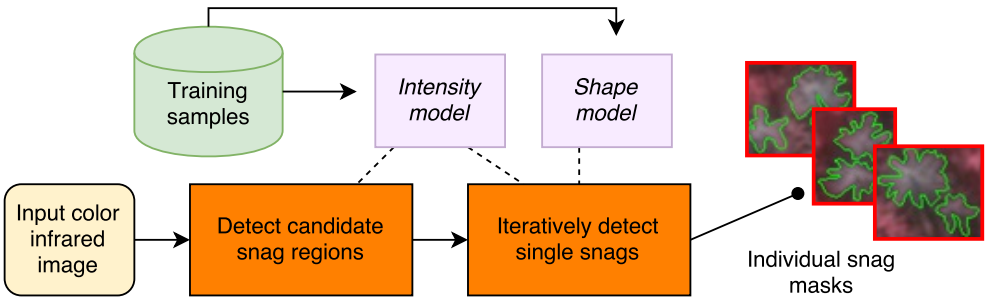


Figure 5.2: Overview of strategy for detecting snags from multispectral imagery

5.1.1 Snag region detection

This stage of the algorithm aims at finding all regions in the input image having a pixel intensity profile similar to that of dead trees. For this purpose, a clustering of the input image is performed using a Gaussian mixture model (GMM) with exactly 3 components, corresponding to the 3 types of areas commonly found in the target scenarios: (i) living vegetation, (ii) dead trees, and (iii) shadows (see Fig. 5.1). The GMM was chosen due to its ability to model interactions between the pixel features via the covariance matrix, and also because of a statistically sound way to introduce

prior information. The GMM is defined in a Bayesian setting, with priors on the component means and variances. Let $X = (x_1, x_2, \ldots, x_N)$ denote the d-dimensional feature vectors of the N image pixels, and β_i, μ_i, Σ_i, respectively, the mixture coefficient, mean and covariance matrix of component $i, i = 1..3$. The posterior probability of the mixture parameters conditioned on the data is proportional to the product of the data likelihood $\mathcal{L}$ and the prior parameter probability $\mathcal{P}_p$ (Eq. 5.1):

$$\mathcal{P}(\mu, \Sigma, \beta | X, \Theta) \propto \mathcal{L}(X|\mu, \Sigma, \beta)\mathcal{P}_p(\mu, \Sigma|\Theta) \tag{5.1}$$

In the above, a flat prior on the mixture coefficients is assumed. The vector $\Theta = (\theta_1, \theta_2, \theta_3)$ represents hyper-parameters of the per-component prior distributions. The data likelihood is the standard probability of a multivariate GMM (Eq. 5.2):

$$\begin{aligned} \mathcal{L}(X|\mu, \Sigma, \beta) &= \prod_{j=1}^{N} \sum_{k=1}^{3} \beta_k \mathcal{P}_k(x_j|\mu_k, \Sigma_k) \\ \mathcal{P}_k(x|\mu, \Sigma) &= |2\pi\Sigma|^{-\frac{1}{2}} exp\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\} \end{aligned} \tag{5.2}$$

Following Fraley & Raftery [2007], an inverse Wishart prior and a multivariate normal prior is applied respectively for the covariance matrices and for the component means (Eq. 5.3).

$$\begin{aligned} &\mathcal{P}(\Sigma|\theta) \propto |\Sigma|^{-(\frac{\nu_p}{2}+2)} exp\{-\frac{1}{2} trace(\Sigma^{-1}\Lambda_p^{-1})\} \\ &\mu_k|\Sigma_k, \theta \sim \mathcal{N}(\mu_{p,k}, \Sigma_k/\kappa_p) \end{aligned} \tag{5.3}$$

In the above, the prior probability model for the covariance matrix is common, whereas the mean priors are component-specific. The hyper-parameter vector θ_k consists of four entries: $\theta_k = (\mu_{p,k}, \Lambda_p, \nu_p, \kappa_p)$. The prior means $\mu_{p,k}$ are calculated based on samples from appropriate patches in the training images. The rest of the hyper-parameters receive the same values for all components. The shrinkage κ_p is set to the product of N and a small positive constant, while the degrees of freedom parameter ν_p assumes a value of $d + 2$. The scale Λ_p is taken to be the covariance matrix of all data points X. Note that the prior on the covariance matrices acts only as a Bayesian regularizer of the parameter estimation problem and does not actually introduce prior information depending on the training examples. The prior distribution imposed on the means is of greater importance, as it attracts the cluster centers towards partitions with regions that roughly correspond to the three semantically different area types. The optimal parameters μ, Σ, β maximizing the posterior probability (Eq. 5.1) are found via the Expectation-Maximization (EM) algorithm (see Sec. 2.1.6), using the prior means and aggregate covariance matrix as the starting point [Fraley & Raftery, 2007]. Despite the regularization, the covariances may degenerate into singular matrices under certain conditions. Specifically, if the image contains no pixels whose values lie close enough to the prior mean $\mu_{p,i}$, then component i degenerates to a singular covariance and zero mixing coefficient. However, this is not a problem, since this situation can be easily detected. If the degenerated component corresponds to the dead tree prior, this is taken as evidence that the image does not contain snags and the processing is stopped. Otherwise, the invalid components are simply discarded and the workflow proceeds normally. The parameters calculated using the EM algorithm are then used to assign each input image pixel to its maximum-probability component. As a result of this step, the set of pixels locations from the input image is obtained which were assigned to the component corresponding to the snag prior intensity mean. For an example, see the partition of Fig. 5.4b into the red, black, and gray regions. Note that since the presence or absence of a tree at a location will ultimately be decided by the fine segmentation, false positive pixels in the region step are not a serious problem. However, false negative pixels may lead to an irrecoverable error of missing a dead tree.

5.1.2 Individual tree delineation

To find the contours of individual dead trees within the relevant image parts, region-based level set segmentation is applied. The key idea behind the level set method is to represent the evolving object contour curve $C \in \Omega$, where $\Omega \subset R^2$ is the image domain, as the zero level set of an embedding function $\phi : \Omega \to R$. In such a setting, the evolution of C is done implicitly through the direct evolution of ϕ. Among the most important advantages of this modification, comparing to a representation of C via a set of control points, is the intrinsic handling of object topology changes (i.e. splitting into several parts) and avoiding the necessity to check for self-intersection of the contour and to perform elaborate re-gridding [Cremers et al., 2007]. Level set methods have been successfully applied to region-based segmentation schemes, wherein the optimized cost functional depends on the properties of entire regions into which the image is partitioned by the contour C. Region-based approaches constituted a major improvement over purely edge-based methods in that they are less sensitive to noise and possess less local minima in their associated energy functionals. Moreover, region-based approaches can be enriched by incorporating prior information about the appearance of target objects. In this thesis, the method proposed by Cremers & Rousson [2007] is adopted, due to its ability to utilize prior information in the form of non-parametric statistical models of both shape and pixel intensity. The original method, introduced in the context of segmenting single intensity channel medical images, is extended to account for multiple spectral bands present in the target scenario. In the remainder of this section, level set segmentation is described in detail, including the methods for incorporating prior knowledge. Also, the specific procedure for segmenting dead trees from multispectral imagery is presented.

General formulation

Let $\Omega \subset R^2$ be the image plane, $I : \Omega \to R^d$ a vector-valued image, and C an evolving contour in the image I. To find an optimal segmentation of I with respect to an energy functional $E(C)$, the contour C is evolved in the direction of the negative energy gradient [Cremers et al., 2007]. In the level set method, C is represented implicitly as the zero level set of an embedding function $\phi : \Omega \to R$. Then ϕ is evolved according to an appropriate partial differential equation dependent on the energy functional E (Eq. 5.4):

$$\frac{\partial \phi}{\partial t} = -\frac{\partial E}{\partial \phi} \tag{5.4}$$

A common choice for ϕ is the signed distance function, whose value for a point $p \in \Omega$ is the negative distance from p to the contour C if p is inside the contour, the positive distance from p to C if p is outside C, and 0 if p lies on the contour itself.

Relationship to region-based segmentation

Consider the problem of binary segmentation, i.e. partitioning the image I into two disjoint (not necessarily connected) regions Ω_1 and Ω_2. Using the Bayesian rule, the conditional probability of the partition given the image can be expressed as:

$$\mathcal{P}(\Omega_1, \Omega_2 | I) \propto \mathcal{P}(I | \Omega_1, \Omega_2) P(\Omega_1, \Omega_2) \tag{5.5}$$

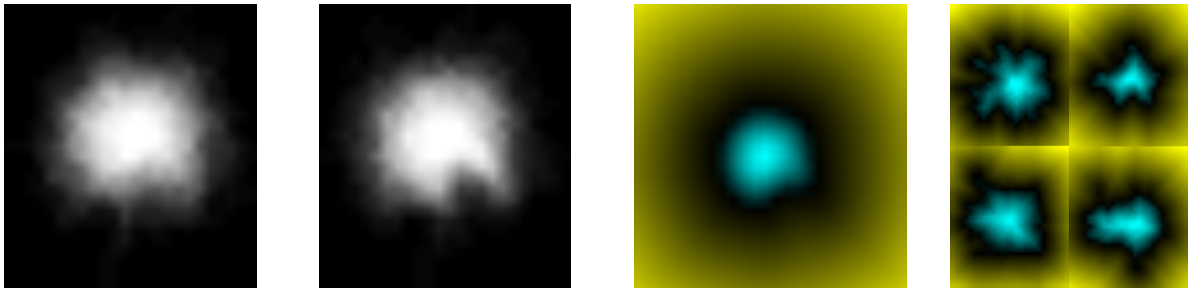
Moreover, since the embedding function ϕ uniquely defines the contour C and hence the partition Ω_1, Ω_2, we have that:

$$\begin{aligned} E(\phi) &= -log(\mathcal{P}(\phi | I)) = -log(\mathcal{P}(I | \phi)) - log(\mathcal{P}(\phi)) \\ &= E_{img} + E_{shp} \\ E_{img} &= -log(\mathcal{P}(I | \phi)), E_{shp} = -log(\mathcal{P}(\phi)) \end{aligned} \tag{5.6}$$

This forms a decomposition of the partition energy $E(\phi)$ into the sum of an image term E_{img} and a shape term E_{shp}. If it is further assumed that region labellings are uncorrelated, i.e. $\mathcal{P}(I|\Omega_1, \Omega_2) = \mathcal{P}(I|\Omega_1)\mathcal{P}(I|\Omega_2)$, and also that image values inside a region are realizations of independent and identically distributed random variables [Cremers et al., 2007], the image term can be written as:

$$\begin{aligned} E_{img}(\phi) &= -\int_{\Omega_1} log f_1(I(x))dx - \int_{\Omega_2} log f_2(I(x))dx \\ &= -\int_{\Omega} H[\phi(x)] log \frac{f_1(I(x))}{f_2(I(x))} dx + Z \end{aligned} \tag{5.7}$$

In Eq. 5.7, f_1 and f_2 represent, respectively, the probability density functions of the image values in region Ω_1 and Ω_2, $H(z)$ indicates the Heaviside function, while Z is a constant independent of ϕ. The models f_1, f_2 can either be specified a priori, or are assumed to follow a distribution from a known family but with unknown parameters, which are functions of the partition and must be co-estimated in the optimization process. Optimizing the functional in Eq. 5.6 with respect to ϕ is an infinite-dimensional problem which can be solved using calculus of variations.



(a) (b) (c) (d)

Figure 5.3: Deriving the eigenshape representation. (a), (b): mean polygon masks formed by summing all pixel values at corresponding locations in training polygons before and after alignment, respectively. (c): the mean signed distance function (SDF) of the training set. (d) SDFs of sample training shapes represented using 20 first eigenmodes of shape variation. In (c)-(d), the cyan and yellow colors represent the interior and exterior of the shape, respectively, with intensity proportional to the distance from the shape contour.

Eigenshape representation

Instead of directly optimizing the energy $E(\phi)$ in a variational framework, the approach described in the pioneering works of Leventon et al. [2000] and Tsai et al. [2003] is taken, where the signed distance function ϕ is constrained to a parametric representation based on the eigenmodes of a set of training shapes. Let $\phi_1, \phi_2, \ldots, \phi_M$ be M signed distance functions of training shapes sampled on a discrete grid. First, the training samples are aligned with respect to translation and rotation using the gradient descent method given in [Tsai et al., 2003] (Figs. 5.3a, 5.3b). The mean training shape is denoted by $\bar{\phi}$ (Fig. 5.3c). The shape variability matrix S is defined as:

$$S = \begin{pmatrix} vec(\phi_1 - \bar{\phi}) & vec(\phi_2 - \bar{\phi}) & \cdots & vec(\phi_M - \bar{\phi}) \end{pmatrix} \tag{5.8}$$

In Eq. 5.8, $vec(M)$ indicates an operator that creates a column vector out of a matrix M by stacking its columns. Subsequently the eigenvectors $\psi_1, \ldots, \psi_c$ of the matrix SS^T corresponding to its largest c eigenvalues are calculated. The parametric representation of the signed distance function can be then expressed as:

$$\phi_{\alpha,h,\theta}(x) = \bar{\phi}(R_\theta x + h) + \sum_{i=1}^{c} \alpha_i \psi_i (R_\theta x + h) \tag{5.9}$$

Fig. 5.3d depicts the eigenmode representation of example training shapes. The parameter vector α models shape deformations, while the rotation angle θ and translation offset h define the rigid transformation of the training shapes. Similarly to [Cremers & Rousson, 2007], scaling is omitted from the shape representation, because the scale of the target objects (dead trees) is directly represented by the training samples. The main benefit of representing ϕ in terms of a set of parameters is that the infinite-dimensional variational problem of optimizing the energy functional (Eq. 5.6) with respect to ϕ can be replaced by a finite-dimensional optimization problem with respect to the parameters, carried out in the constrained subspace spanned by the training examples:

$$E(\phi) = E(\alpha, h, \theta) = -log(\mathcal{P}(I|\alpha, h, \theta)) - log(\mathcal{P}(\alpha, h, \theta)) \tag{5.10}$$

Prior information

The goal is to derive a shape prior $\mathcal{P}(\phi) = \mathcal{P}(\alpha, h, \theta)$ (Eq. 5.6) which models the variability of the training polygons' shapes. The statistical nature of the model ensures that the evolved contours remain similar to, but not necessarily the same as, the training shapes. For this purpose, the nonparametric approach outlined in [Cremers & Rousson, 2007] is adopted. Note that only the shape variability parameters α are modeled, while a flat prior for the rigid transformation parameters θ, h is assumed, because all rotation angles and offsets are considered as equally likely to appear in the investigated scenario. In accordance with the original method, first the representations $\alpha_1, \ldots, \alpha_M$ of the M original training shapes $\phi_1, \ldots, \phi_M$ are calculated in terms of the eigenmodes $\psi_i, \ldots, \psi_c$. The probability of any shape parameter vector α can now be expressed using a kernel density estimate with respect to the training shapes:

$$\mathcal{P}(\alpha) = \frac{1}{M\sigma^c} \sum_{i=1}^{M} K(\frac{\alpha - \alpha_i}{\sigma}), K(u) = \frac{1}{2\pi^{c/2}} exp(-\frac{u^2}{2}) \tag{5.11}$$

The kernel bandwidth σ can be calculated using methods described in Sec. 2.3.1, or set to the average nearest neighbor distance within the training set to avoid computational overhead. The nonparametric shape prior has the advantage of not making too restrictive assumptions about the distribution of the shape parameters, which allows it to represent complex relationships between them more accurately compared to a parametric (e.g. Gaussian) model.

The intensity prior complements the shape model by encouraging the evolving shape to include patches of pixels with values frequently occurring in the training polygons, and simultaneously discouraging regions whose pixels values are uncharacteristic of the target class. In the dead tree setting, this amounts to fixing the intensity models f_1, f_2 (Eq. 5.7) for the foreground and background regions. In particular, Cremers & Rousson [2007] proposed to construct two independent kernel density estimators for this task. However, since their method was designed to accept images with a single intensity channel, in this thesis a different modeling technique is employed which is able to capture the inter-dependencies between the various channels present in the target images. To achieve this, it first noted that the image term E_{img} of the energy functional (Eq. 5.7) does not require the knowledge of the individual per-region probabilities $f_1(I(x)), f_2(I(x))$ for any point x, but only their ratio $r = f_1(I(x))/f_2(I(x))$. Note that $f_1(y)$ and $f_2(y)$ represent the region label-conditional probabilities of observing a feature value y. The ratio r can be represented by

inverted models which quantify the probability of point x having a label $l \in 1, 2$ given its feature value $I(x)$:

$$\begin{aligned} r(x) = \frac{f_1(I(x))}{f_2(I(x))} = \frac{\mathcal{P}(I(x)|l=1)}{\mathcal{P}(I(x)|l=2)} = \frac{\mathcal{P}(l=1|I(x))\mathcal{P}(I(x))}{\mathcal{P}(l=1)} \\ \times \frac{\mathcal{P}(l=2)}{\mathcal{P}(l=2|I(x))\mathcal{P}(I(x))} = \frac{\mathcal{P}(l=1|I(x))}{\mathcal{P}(l=2|I(x))}\frac{\mathcal{P}(l=2)}{\mathcal{P}(l=1)} \end{aligned} \tag{5.12}$$

Therefore, r is a quotient of the ratio of posterior region label probabilities and the ratio of the prior region label probabilities. Moreover, the logarithm of $r(x)$ can be written as $\log(r(x)) = \log\frac{\mathcal{P}(l=1|I(x))}{\mathcal{P}(l=2|I(x))} - D$, where D is a constant, because the prior probabilities do not depend on the point x. The re-expressing of r in terms of posterior label probabilities adds a D-multiple of ϕ's interior area to the energy E_{img}. Here, a uniform prior on the labels is assumed, resulting in $D = 0$ and hence no change to the energy. This means that any binary discriminative model of the boundary between the foreground and background region can be incorporated into E_{img}. In this setting, $\mathcal{P}(l = 2|I(x)) = 1 - \mathcal{P}(l = 1|I(x))$, so E_{img} can be rewritten as:

$$E_{img}(\phi) = -\int_\Omega H[\phi(x)] log \frac{\mathcal{P}(l=1|I(x))}{1-\mathcal{P}(l=1|I(x))} dx \tag{5.13}$$

The quantity $log\{\mathcal{P}(l = 1|I(x))/(1-\mathcal{P}(l = 1|I(x)))\}$ is reminiscent of the logit function (cf. Eq. 2.16). Indeed, if logistic regression is chosen as the discriminative model, then E_{img} boils down to:

$$E_{img}(\phi) = -\int_\Omega H[\phi(x)]\omega I(x) dx \tag{5.14}$$

In the above, the vector ω represents the trained coefficients of the logistic regression model. In order to retain the power of nonparametric approaches, the logistic regression model is kernelized on T training points $y_1, \ldots, y_T$ by transforming the image value vector $I(x)$ into a new feature vector $F(x) = [K(\frac{I(x)-I(y_1)}{\sigma_k}), \ldots, K(\frac{I(x)-I(y_T)}{\sigma_k})]$ using a positive definite kernel K (see Sec. 2.2.1). The bandwidth σ_k is chosen using cross validation. The final form of the logit is then $\omega F(x)$. The introduction of intensity priors causes the segmentation energy functional to depend on the pixel values $I(x)$ only through their foreground probabilities $f_1(I(x))$, which means that the algorithm is effectively working on a one-channel probability image (see Fig. 5.4c).

Final energy formulation

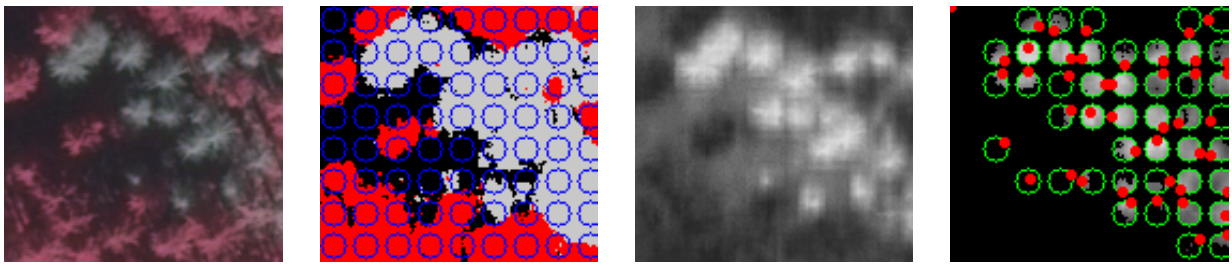
After defining both the shape and image terms in the energy functional (Eq. 5.6), the final form of the energy used in the proposed method can be written down [Polewski et al., 2015c]:

$$E(\alpha, h, \theta) = -\int_\Omega H[\phi_{\alpha,h,\theta}(x)]\omega F(x) dx - log \sum_{i=1}^{M} K(\frac{\alpha - \alpha_i}{\sigma}) \tag{5.15}$$

Partial differential equations are set up which govern the evolution of the function ϕ in the direction of the negative energy gradient:

$$\begin{aligned} \frac{\partial \alpha}{\partial t} &= -\int_\Omega \omega F(x)\delta[\phi(x)] \begin{pmatrix} \psi_1(x') \\ \vdots \\ \psi_c(x') \end{pmatrix} dx + \frac{1}{\sigma^2}\frac{\sum_{i=1}^M(\alpha_i - \alpha)L_i}{\sum_{i=1}^M L_i} \\ \frac{\partial h}{\partial t} &= -\int_\Omega \omega F(x)\delta[\phi(x)] \left(\frac{\partial \phi}{\partial x_1}, \frac{\partial \phi}{\partial x_2}\right)^T dx \\ \frac{\partial \theta}{\partial t} &= -\int_\Omega \omega F(x)\delta[\phi(x)] \left(\frac{\partial \phi}{\partial x_1}, \frac{\partial \phi}{\partial x_2}\right)^T \frac{\partial R_\theta}{\partial \theta} dx \end{aligned} \tag{5.16}$$

In the above, $x' = R_\theta x + h$, $x = (x_1, x_2)$, $L_i = exp(\frac{-|\alpha-\alpha_i|^2}{2\sigma^2})$, and $\frac{\partial R_\theta}{\partial \theta}$ is the component-wise derivative of the rotation matrix R_θ by the angle θ (see Tsai et al. [2003] for details). The term $\delta(z)$ denotes the Dirac delta function. In practice, a smooth approximation of δ is used, and the standard gradient descent method is applied to update the parameters α, h, θ with a small time step dt until convergence is reached.



(a) (b) (c) (d)

Figure 5.4: Selecting locations for initializing the level set segmentation. (a) Input image. (b) Result of candidate region detection covered by circles which indicate candidate starting locations. (c) Pixel-level dead tree probabilities of image points from intensity prior. (d) Intersection of candidate circles and dead tree cluster from candidate region detection. Red dots mark chosen starting locations for segmentation, grayscale values depict dead tree probability as in (c).

5.1.3 Iterative segmentation

The level set formulation described in Sec. 5.1.2 allows the segmentation process to converge to at most one object. The location of the final shape is dependent on the initial position of the contour. However, in the target scenario the input image is expected to contain many dead trees. Therefore, a procedure is developed to iteratively apply the segmentation at different locations of the image, while ensuring that the individual segmented shapes do not overlap. To achieve this, first the input image is covered with circles C_i of a diameter d approximately equal to that of the containing circle of the largest training polygon (representing a tree crown), by placing them on a regular grid (see Fig. 5.4). Then each of the circles C_i is considered as a potential location for the initial contour of the level set segmentation. Let C_S be the set of the circles C_i which have a non-empty pixel intersection I_i with the snag cluster obtained in the candidate region detection step (Sec. 5.1.1). Let $p_i = \arg\max_{p \in I_i} \mathcal{P}(l = \text{'dead tree'}|I(x))$ be the pixel in I_i which attains the highest dead tree class probability from the intensity prior (Sec. 5.1.2). The circles in $C_i \in C_S$ are ranked according to the probability of their best point p_i and and subsequently processed in the resulting order. For a circle C_i, the level-set segmentation is executed starting with the mean shape $\bar{\phi}$ centered at point p_i. If the segmentation converges to an empty contour, it is concluded that there is not enough evidence in the image region around p_i to support the existence of a dead tree. Otherwise, the polygon mask obtained from the converged shape s_i is added to the result set. Note that the image may contain strong local attractors to which the segmentation would converge from starting points within a large radius. To mitigate this, the image intensity values of all pixels belonging to s_i are set to a value which attains a near-zero probability in the prior intensity distribution. This prevents the formation of overlapping polygons at that location in subsequent iterations of the segmentation process.

5.2 Detection from aerial imagery combined with ALS point clouds

In this scenario, both multispectral imagery and ALS point clouds of the target plot are available. The core idea of the method is to take advantage of the detailed 3D information available in the point cloud to segment individual trees, and on the other hand make use of the strong discriminative capabilities of NIR spectral bands for dead vegetation detection. The entire classification workflow proceeds as follows. First, the point cloud is segmented in 3D to yield point clusters representing single trees. The points belonging to each tree are subsequently projected onto the image plane and a bounding polygon in 2D is calculated. Then, spectral features are extracted based on the intensities of pixels located inside the bounding polygon. These features form the basis for classification as either a dead or living tree.

5.2.1 Individual tree segmentation

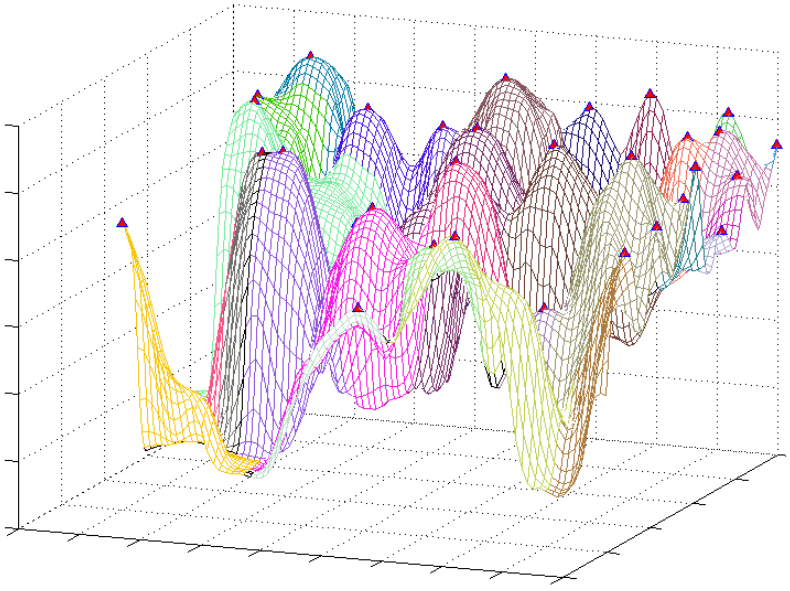
To obtain individual tree point clusters, the segmentation method proposed by Reitberger et al. [2009] is applied. This approach consists of two steps. Initially, a smoothed canopy height model (CHM) is generated from the input point cloud, and the watershed segmentation algorithm is used to detect local maxima in the CHM (see Fig. 5.5a). The maxima locations represent hypotheses about tree positions and are utilized as prior knowledge in the next step of the method. There, the point cloud is segmented in 3D using the aforementioned Normalized Cut algorithm (see Sec. 2.4), with the standard multiplicative exponential similarity function (Eq. 2.34). The segmentation is carried out on a voxelized version of the point cloud, and the differential features represent horizontal and vertical distance between voxel pairs, differences in laser reflection intensities, as well as maximum distances to the closest local CHM maximum found in the previous step, which encodes prior information about probable tree positions. A sample point cloud colored according to the single tree Ncut segmentation result is visualized in Fig. 5.5b.

5.2.2 Computing the bounding polygon

For each segmented tree, the set of available aerial images is examined to find the image whose center coordinates (expressed in the appropriate World coordinate system) are closest to the planimetric centroid of the tree's points. This helps reduce image deformation effects caused by perspective distortion, which become more prominent as the distance from the image center increases. All points belonging to the tree cluster are then projected onto the selected image by applying the standard collinearity equations [Luhmann et al., 2013] to the known exterior and interior camera orientations associated with the image acquisition. The 2D convex hull of the projected points is calculated, and it is considered the tree's bounding polygon on the image plane. Two tree segments along with their 2D bounding polygons are depicted in Fig. 5.6.

5.2.3 Feature extraction

The extraction of features for a tree segment is based on the image pixels which lie inside its bounding polygon as defined above. The features used in this scenario include the mean intensities of the NIR, read and green channels over all interior pixels, as well as the 6 independent elements of the spectral channel covariance matrix. This yields a total of 9 features, which are subsequently used for binary classification.



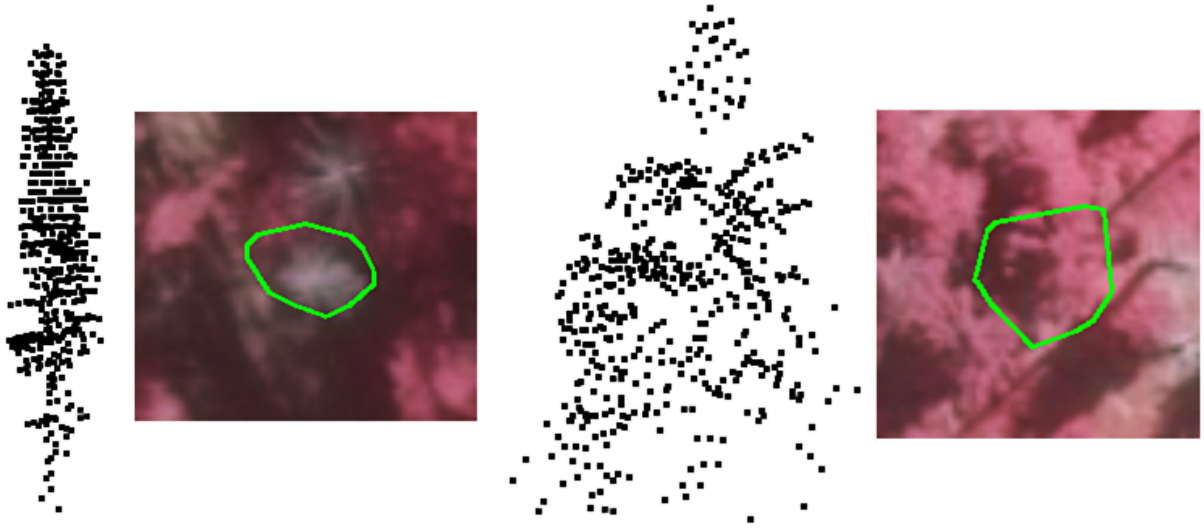
(a)

(b)

Figure 5.5: Results of watershed and Ncut segmentation of a forest plot. (a) Watershed segmentation of canopy height model, with triangular markers showing local maxima. (b) Ncut segmentation, with points colored based on assignment to single trees (unassigned points in red).

5.3 Detection from ALS point clouds using Free Shape Contexts

In the final part of this chapter, the case of detecting standing dead trees which have lost large portions of their crowns is considered. Due to the aforementioned fact that such trees are expected to possess only a very limited signature in any nadir-view imagery (as a result of small cross-section area), the proposed method concentrates of 3D point cloud data to classify single trees and does not utilize aerial imagery. As in Section 5.2, the scene is first segmented



(a) (b)

Figure 5.6: The 2D bounding polygons (in green) of a (a) dead and (b) living tree, obtained by calculating the convex hull of corresponding ALS points projected onto image.

into clusters representing individual trees using the approach outlined in Sec. 5.2.1. To classify each cluster as either dead or living, a new type of 3D shape descriptor, termed the Free Shape Context (FSC) is introduced to generalize the standard 3D Shape Contexts (SC) described in Secs. 2.6 and 4.1.2. The FSC descriptors are tailored to the task of detecting general elongated objects which possess a well-defined axis from point clouds. The associated features provide a comprehensive volumetric description of the target object's point cloud. This makes FSC useful in scenarios where the point cloud is sparse and/or noisy, which would make the construction of more detailed surface descriptors such as PFH (Sec. 2.6) problematic due to inaccurate surface normal estimation. The FSC descriptor uses the cylindrical SC as a building block. In particular, it can be thought of as a sequence of cylindrical volumes around a common axis. These volumes are further subdivided along 3 dimensions (axial, radial, angular) to produce a spatial occupancy histogram of points within the scene, similar to the original SC. The dimensions (radii, heights) and subdivision parameters of each cylinder are independent from the other descriptor parts. This enables the FSC to assign different levels of detail and importance to intervals along the object axis, in order to construct a representation that allows the best discrimination between the object class of interest and irrelevant objects. This flexibility comes at a price of greatly increased computational complexity, since not only the dimensions of the cylindrical volumes, but also the entire *structure* of the FSC (number of member cylinders, their arrangement along the axis) needs to be determined. Therefore, exhaustive approaches such as grid search are usually infeasible for this purpose. Instead, a genetic algorithm (GA) is proposed as a tool for optimizing the FSC parameters and structure. The rest of this section is dedicated to explaining the details of calculating the object axis, providing a precise definition of the Free Shape Contexts, as well as describing the optimization process based on the genetic algorithm.

5.3.1 Calculating object axis

In this step, for each candidate object a principal axis is found which best fits the corresponding set of points. Since the tree clustering step is likely to produce imperfect partitions, the presence of clutter and/or parts of other objects is expected within the point cloud. For this reason, an estimation method from the Sample Consensus (SAC) family (see Sec. 2.5) is applied to calculate

the axis. Specifically, the M-estimator SAC variation (Eq. 2.38) is utilized, based on a 3D line model, cylindrical neighborhood with radius r_{sac} and point-to-line perpendicular distance as the metric. Additionally, a constraint on the maximum angular deviation α_{dev} of the candidate line models from the world Z coordinate axis is introduced to account for the phenomenon of gravitropism in trees. A sample result of the calculated axes for several point clouds is depicted by Fig. 5.7.

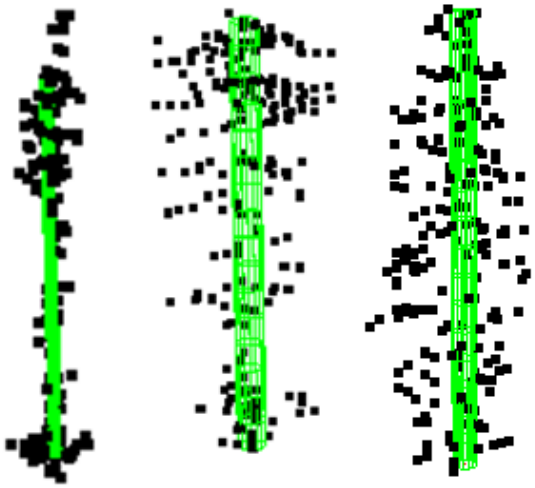


Figure 5.7: Axes of point clouds determined by Sample Consensus.

5.3.2 Free shape contexts

The standard cylindrical Shape Context (Fig. 4.5b), which was utilized for detecting fallen stem segments in Sec. 4.1.2, has a uniform structure. However, for a specific detection task, it is unlikely that discriminative information is evenly distributed along the object axis. In particular, some parts may be heavily affected by clutter, which renders them useless as a source of distinctive features. Furthermore, for the parts which are relevant, the optimal neighborhood may significantly vary in size. Based on these two intuitions, a new shape descriptor, the Free Shape Context (FSC), is proposed [Polewski et al., 2015d]. The FSC is essentially a stack of cylindrical volumes arranged around the axis. This can be viewed as a sequence of adjacent standard SCs which are divided along the radial dimension, i.e. potentially contain embedded cylinders of smaller radii (see Fig. 5.8). The lengths and radii of the constituent SCs are unrelated and may be adjusted independently. Additionally, every cylinder has an activity state which controls whether the associated histogram bin is allowed to emit a feature to the final descriptor. Inactive cylinders can be regarded as buffer elements - they only define the structure of the FSC without affecting feature values. The FSC is parameterized by the ordered sequence of the member SCs' parameters. Each such SC is in turn described by its length and sequence of cylinder radii along with their corresponding active states. Due to the difficulty of obtaining a meaningful 0° reference angle for standing trees, the angular subdivision is not explored in this work. However, for other applications where a useful 0° angle can be defined, the third partitioning dimension could be easily incorporated into the FSC.

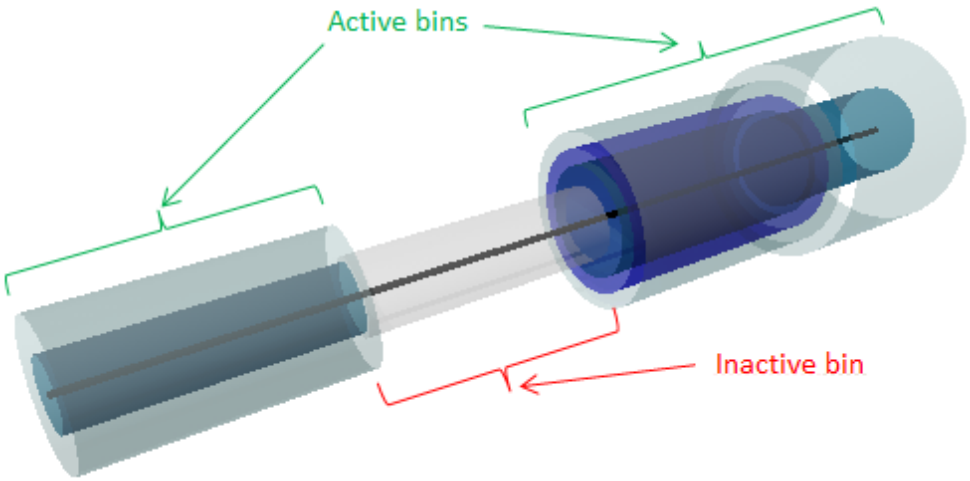


Figure 5.8: Example of a Free Shape Context. Gray cylinder marked with red bracket is inactive - it will not contribute to the shape histogram.

Parameter space size

The FSC's increased number of degrees of freedom results in a considerably larger parameter space. To analyze the parameter space, some notation is introduced. Let N_R, N_L be the number of quantized radius and length configurations, respectively, that can be examined with a reasonable computational effort. We then have that for an FSC consisting of a single SC with a single cylinder, the number of configurations is $2N_RN_L$. The factor 2 arises from the two possible activity states. If a new cylinder is added to the SC, the new radius can be set to any other of the N_R values, assigned an active or passive state and combined with any previous configuration, resulting in $2^2\binom{N_R}{2}N_L$ options. Following this line of reasoning, the number of configurations for an FSC consisting of one SC with i cylinders is: $C_{1,i} = 2^i\binom{N_R}{i}N_L$. Let n_r denote the maximal number of radial bins. The total number of FSC configurations consisting of k SCs can be expressed using $C_{1,i}$: $C_k = (C_{1,1} + \ldots + C_{1,n_r})^k$. This expression is dominated by the term $\{2^{n_r}\binom{N_R}{n_r}N_L\}^k$. It can be easily verified that even for a coarse quantization of the radii and lengths and a small number of SCs, the computational complexity renders any exact or even grid search approach infeasible.

Approximate optimization

It is possible to alleviate the computational burden by making two simplifying assumptions. The first assumption states that the problem of finding the most discriminative FSC exhibits the optimal substructure property. In practice this means that if the best FSC f of length L contains its last shape context s from positions D to L along the axis, then the FSC obtained by removing s from f must be the most discriminative FSC of length D. The second assumption recasts this requirement in the context of cylinder radii within a single SC. The optimal substructure premise gives rise to a dynamic programming strategy for calculating the optimal FSC. Let $W_{k,j}$ denote the best score of a k-component FSC of (quantized) length j, and $P_{k,j}$ the corresponding FSC parameters. Also, let $V_{k,j}(P, l)$ be the smallest error of an FSC formed by combining the parameters P with a new SC consisting of k cylinders with a total (external) radius j and length l. Let $S_{k,j}^{P,l}$ denote this optimal SC, and c_r - a cylinder with radius r. The minimum cost can then be calculated:

$$W_{k,j} = \min\{W_{k,j-1}, \min_{l,u,v} V_{u,v}(P_{k-1,j-l}, l)\} \tag{5.17}$$

$$V_{k,j}(P, l) = \min\{V_{k,j-1}(P, l), \min_{r} E(P \oplus [S_{k-1,j-r}^{P,l} \cup c_j])\}$$

In the above, $\oplus$ denotes appending an SC at the end of an FSC, and $E(\cdot)$ is the evaluation function (e.g. classification error). Of course, the simplifying assumptions made here need not hold in practice, so this procedure should be regarded as a coarse approximation and computationally feasible replacement for grid search.

5.3.3 Optimizing FSC with genetic algorithm

Due to the size of the parameter space of the FSC, it is unlikely that any approach which attempts to sample parameters in equal intervals will attain a near-optimal solution. The parameter space is usually highly heterogeneous, with small isolated regions of high-quality solutions. Therefore, it appears more beneficial to apply an optimization strategy capable of dedicating more effort to the exploration of promising regions of the solution space at the expense of omitting its weaker parts, thereby making a better use of the allocated resources. One such method is the Genetic Algorithm, a biologically inspired meta-heuristic which has been successfully applied to the optimization of multiple problems in science and engineering. As a representative of evolutionary computation methods, Genetic Algorithms [Goldberg, 1989] are an optimization technique which attempts to mimic an evolutionary process. In the language of the biological metaphor, solutions to the problem of interest are referred to as *individuals* and are encoded using a *genetic representation*. The value of the optimized function f associated with a solution is called the *fitness function* of the individual. The genetic algorithm acts on a *population* of individuals by iteratively selecting the best elements based on their fitness and subsequently recombining them to create a new generation of individuals. The recombination usually consists of two parts: mutation and crossover. Mutation is a unary operator which randomly alters small parts of the target individual's genome. Crossover is an operator acting on a pair of individuals ('parents') with the purpose of randomly exchanging parts of their genomes. The iteration is repeated until a convergence criterion is met. It is expected that in the course of evolution, the individuals will gradually improve their fitness until a (local or global) extremum of the fitness function is attained.

FSC genotype representation

A variable-length, structured genetic representation $G(c)$ of the Free Shape Context c is used. At the top level, the genome is an ordered sequence of the constituent SCs' sub-genomes: $G(c) = (G(sc_1), \ldots, G(sc_N))$. Each sub-genome $G(sc_k) = (l_k, ([r_{k,i}, a_{k,i}])_{i=1..n_k})$ is composed of the length l and list of the n_k cylinder radii r and active states a. The distinction is made between *structural parameters*, which describe the number of SCs, the number of radii within each SC as well as the active state of each cylinder, and *size parameters*, which encode the cylinder lengths and radii.

Mutation

For the structural parameters, a simple mutation scheme is utilized which randomly adds and removes SCs from the FSC, adds and removes cylinders from a single SC, or flips the active state. All of these events occur with a small probability $p_{m,1}$. For the size parameters, a relative mutation strategy is used. A random number κ is drawn from a zero-mean normal distribution with standard deviation $\sigma_m \in (0; 1)$, and the target parameter p is updated according to: $p \leftarrow p(1+\kappa)$. This mutation takes place with a probability $p_{m,2} > p_{m,1}$. The mutation rates for the structural and size parameters are distinct because the evolutionary process should spend sufficient effort examining the size parameters for a given structure. This is motivated by the fact that the number of structural configurations is small compared to the number of length/radius configurations.

Crossover

The crossover operator is implemented by a randomized strategy which picks, with uniform probability, one of three procedures. The first two, one-point and two-point cross-over, take place only at the top level and involve swapping SCs at corresponding positions in the genome. The crossover points (one or two) are picked at random. Finally, the Fuzzy Recombination (FR) crossover takes place between corresponding SCs at each position in the parents' genomes, with probability p_{FR}. In this procedure, the lengths and radii of the SCs' cylinders are subjected to the FR operator, which aims at integrating the effects of crossover and mutation. For two values p_1, p_2 and a width parameter $0 \leq d_{FR} \leq 0.5$, FR defines a bi-triangular probability distribution on the output value, with modes located at p_1 and p_2 (Fig. 5.9). Large values of d_{FR} lead to introducing more diversity into the population, whereas small values benefit the protection of good existing solutions.

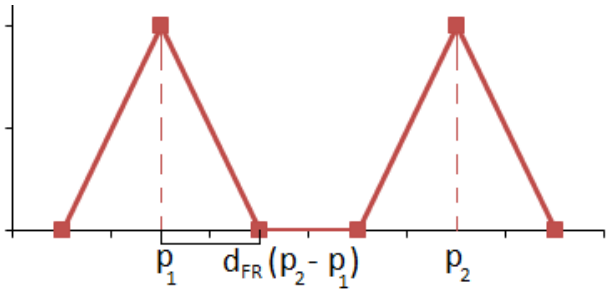


Figure 5.9: Bi-triangular output distribution induced by Fuzzy Recombination operator.

Fitness function

The evolution of the Free Shape Contexts is driven by their overall classification performance on a set of labeled objects (point clouds). In particular, the labeled data set is not partitioned into training and test sets. Instead, a cross-validation scheme is applied for calculating the classification error. It is important that the error measure be deterministic in the sense of always obtaining the same error value upon repeated calculation for the same individual. Otherwise, the evolution could be promoting random constellations of training vs. test set partitions which lead to lower errors at the expense of true discriminative performance. For this reason, leave-one-out cross-validation is used, which is free from any randomness.

6 Experiments

This chapter provides insight into the various experiments which were designed and carried out with the purpose of quantifying the influence of the proposed methodological innovations introduced in this thesis on the final quality of detection and reconstruction of standing and fallen dead trees. When possible, the experiments followed a modular design, which enabled to determine the contribution of a single element of the processing pipeline on the result. First, Section 6.1 describes the data acquisition campaigns, which served to collect airborne and terrestrial laser scanning point clouds as well as aerial imagery. Subsequent sections characterize the experimental design, along with the parameter settings and the evaluation strategy used in each case. Also, the process of obtaining ground truth/reference data as a basis for quantitative evaluation is detailed. Specifically, Section 6.2 deals with fallen trees, whereas the focus of Section 6.3 is on standing dead trees.

6.1 Material

All dead tree mapping methods presented in this thesis were tested on plots from the Bavarian Forest National Park (49°3′19″ N, 13°12′9″ E), which is situated in South-Eastern Germany along the border to the Czech Republic, within the federal state of Bavaria (see Fig. 6.1). The studies were carried out in the mountain mixed forests zone consisting mostly of Norway spruce (*Picea abies*) and European beech (*Fagus sylvatica*). From 1988 to 2010, a total of 5800 ha of the Norway spruce stands died off because of a bark beetle (*Ips typographus*) infestation [Lausch et al., 2013]. The dead trees were retained in the area for the purposes of scientific study.

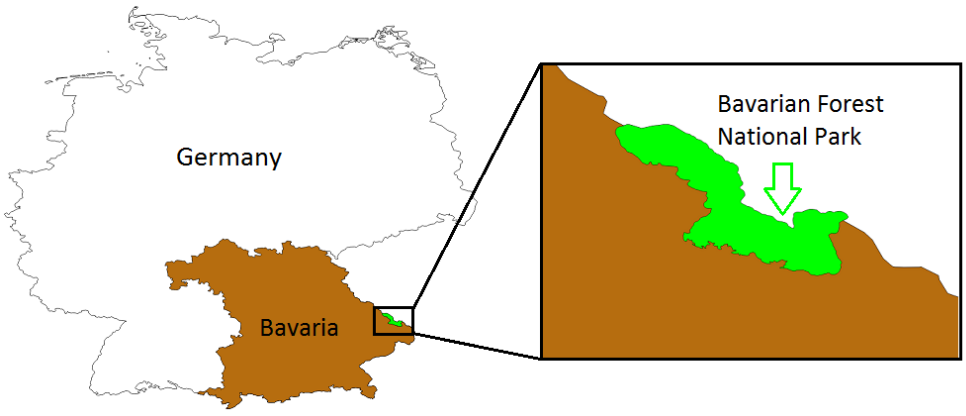


Figure 6.1: Location of Bavarian Forest National Park in Germany and federal state of Bavaria.

6.1.1 ALS - fallen trees

The airborne full waveform data were acquired using a Riegl LMS-Q560 scanner in April 2011 in a leaf-off condition with a nominal point density of 30 points/m^2. The vertical sampling distance was 15 cm, the pulse width at half maximum reached 4 ns and the laser wavelength was 1550 nm. The flying altitude of 400 m resulted in a footprint size of 20 cm. The collected full waveforms were decomposed according to a mixture-of-Gaussians model [Reitberger et al., 2008], resulting in a 3D point cloud. Five plots, referred to as Plot 1 - Plot 5, were used as a basis for evaluating the fallen tree reconstruction pipeline from Sec. 4.1, with additional two plots T1 and T2 used for training the point-level and segment-level classifiers. The plot characteristics are summarized in Table 6.1, whereas Fig.6.2a shows the relative locations of the plots on a color infrared orthophotograph of the relevant part of the Bavarian Forest National Park. The point clouds of two plots, representing the beech-dominated and spruce-dominated scenarios, are depicted in Figs. 6.2b,6.2c.

Property	Plot 1	Plot 2	Plot 3	Plot 4	Plot 5
Size [ha]	0.41	0.64	0.49	0.35	0.53
Trees/ha	196	247	261	351	397
Fallen stems /ha	126	108	88	130	98
Dominant species	beech	beech	beech	spruce	spruce
Overstory cover [%]	34	50	45	66	66

Table 6.1: Properties of ALS plots for fallen tree detection

Property	Plot A	Plot B	Plot C
Size [ha]	1.8	0.8	0.18
Living trees per ha	81	339	144
Dominant species	beech	spruce	spruce
Point density per m^2	800	2400	44000
Num. scanning positions	17	10	12

Table 6.2: Properties of TLS test plots for fallen tree detection

6.1.2 TLS - fallen trees

To test the fallen tree detection method for dense data (Sec. 4.2), three test plots were scanned. They are denoted Plot A - Plot C to avoid confusion with the ALS plots. The laser data were acquired using a Riegl LMS-Z390i terrestrial scanner in June-August 2009 (Plots A and B) as well as October 2011 (Plot C). The laser scanner was mounted on a tripod, resulting in an

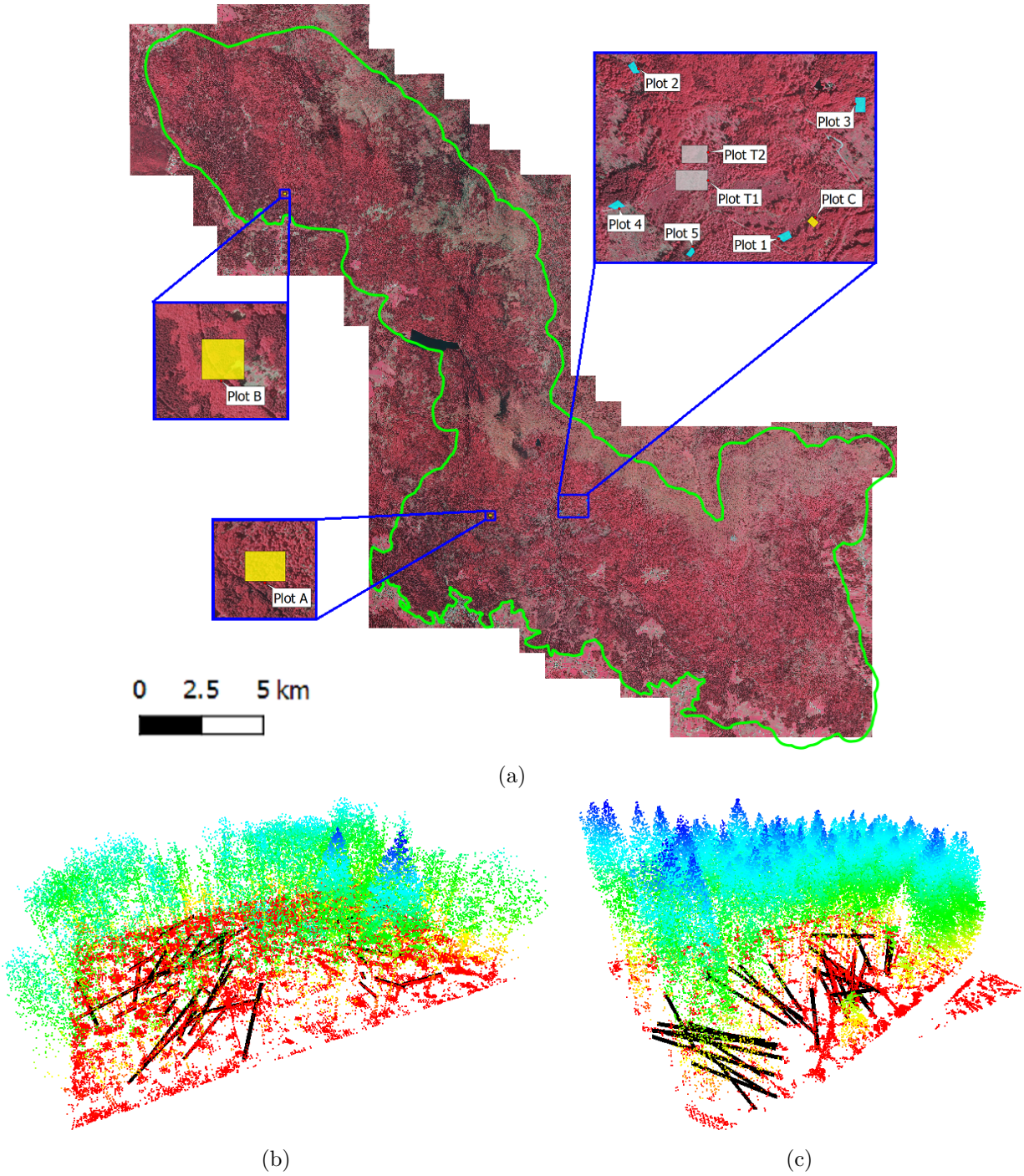
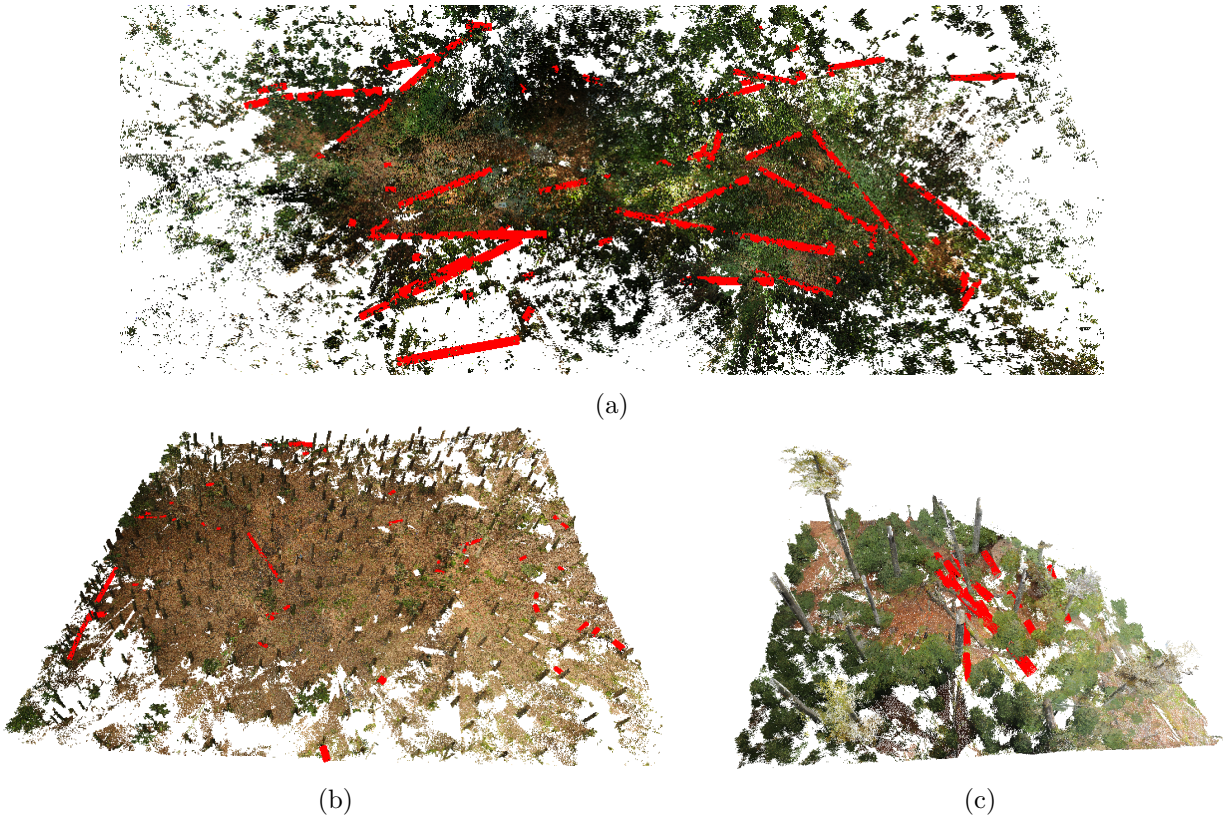


Figure 6.2: (a) Color infrared orthophotograph of Bavarian Forest National Park (park area in green), with marked test plots for detecting fallen trees. Plots 1-5 as well as T1,T2 were scanned using ALS, while Plots A,B,C were captured with TLS. (b)-(c) ALS point clouds of (b) Plot 2 (beech-dominated) and (c) Plot 4 (spruce-dominated), colored by above-ground height. Points lower than 20 cm above the ground have been removed, revealing fallen stems and ground vegetation. Black bars represent fallen stems measured during field inventory.

above-ground height of 1.4-1.6 m. Additional RGB information was collected using an external calibrated Nikon D300 single lens reflex camera with a resolution of 12.3 M pixels, mounted on top of the scanner. The scanning accuracy was 6 mm with an average repeatability of 2 mm, measurement rate of 8000 points/s and beam divergence equal to 0.3 mrad. The co-registration

of point clouds from different scan positions was marker-based, with an accuracy of up to 5 mm. The properties of the test plots, estimated based on the point clouds, are given in Table 6.2, whereas the point clouds themselves are depicted in Fig. 6.3. Also, the locations of Plots A-C within the Bavarian Forest National Park are shown in Fig. 6.2a.



(a)

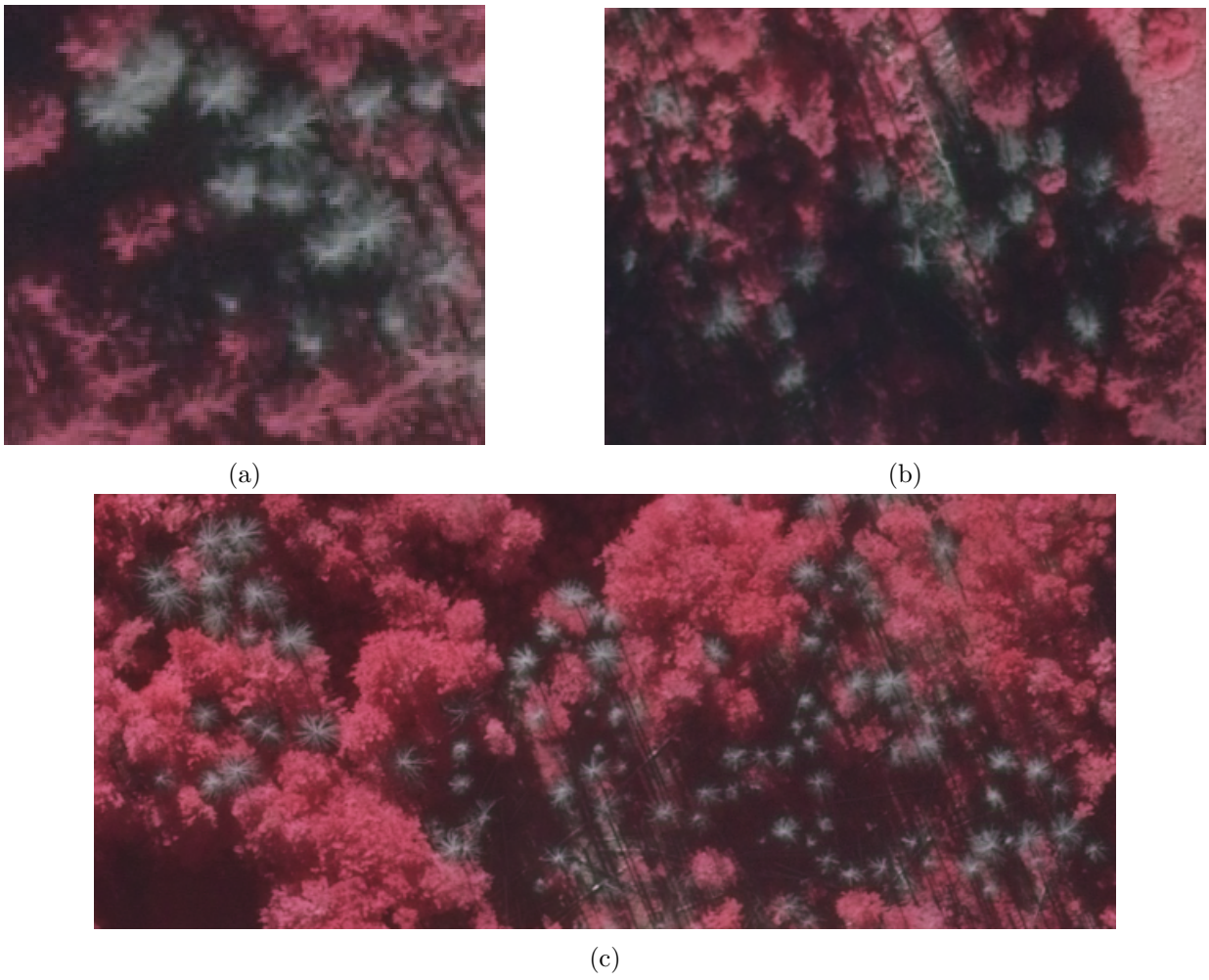
(b) (c)

Figure 6.3: TLS point clouds of Plots A (a), B (b) and C (c), cropped to show areas with fallen trees. The reference lying stems are marked with red bars. The canopy layer in (a) and (b) was removed for better visualization of ground.

6.1.3 ALS & aerial imagery - standing dead trees

The experiments concerning the detection of standing dead trees were based on ALS and aerial imagery data, both acquired in the summer of 2012 (i.e. leaf-on condition) over the entire area of the Bavarian Forest National Park, with a temporal offset of 1 month. Color infrared images were collected in August 2012 using a DMC camera. The mean above-ground flight height was 1900 m, corresponding to a pixel resolution of 20 cm on the ground. The images contain 3 spectral bands: near infrared, red and green. The orthorectified version of the aerial images is shown in Fig. 6.2a. The airborne full waveform ALS data were acquired using a Riegl LMS-680i scanner in July 2012 with a nominal point density of 30-40 points/m^2. The pulse rate was 266 kHz. The flying altitude of 650 m resulted in a footprint size of 32 cm. The collected full waveforms were decomposed according to a mixture-of-Gaussians model [Reitberger et al., 2008] to obtain a 3D point cloud. For the experiment related to dead tree crown delineation (Sec. 6.3.1), three test plots, denoted Plot I, Plot II and Plot III, were selected as regions within the collected aerial imagery. Their respective areas are 618 m^2, 2426 m^2, and 7153 m^2. These plots were chosen to contain diverse scenarios such as shadows, open ground, clusters of nearby dead trees as well as isolated snags surrounded by living vegetation. The color infrared images of the plots are shown in Fig. 6.4. For other experiments concerning standing dead trees, no single test area was defined.

The tree polygons were already fixed and their creation was not part of the experiment, so the polygons chosen for classification were scattered throughout different parts of the National Park.



(a) (b)

(c)

Figure 6.4: CIR images of Plots I - III (a)-(c) for testing active contour-based crown delineation.

6.2 Fallen trees

This section characterizes all experiments carried out for the fallen tree detection methods. First, the evaluation criterion, shared by the ALS and TLS scenarios, is explained. The specific experiments, reference data and parameter settings are given separately for each point cloud type.

6.2.1 Evaluation criterion

For both ALS and TLS data, a vectorized version of the results is available. In particular, a skeleton representation of the stems can easily be obtained, consisting of a sequence of 3D lines representing the stem central axes. This line-based description has the advantage of being more easily comparable to similarly vectorized reference data obtained from field measurements, as opposed to more complex comparison schemes operating directly on the 3D point cloud. Therefore, the applied evaluation criterion is based on the skeleton representation. The evaluation procedure accepts two sets of of stem skeletons (3D line sequences) representing the detected and reference trees. It outputs quantitative information showing to what degree each reference tree corresponds to a detection result. For this purpose, 3 criteria are defined which attempt to narrow down, to

the extent allowed by the accuracy of the reference data, the set of detection results which may be matched to a reference stem. Let R be a reference tree given by the sequence of endpoints of its line segments: $R = \{(r_{1,1}, r_{1,2}), \ldots, (r_{K,1}, r_{K,2})\}$. By analogy, we can define a detected tree $D = \{(d_{1,1}, d_{1,2}), \ldots, (d_{L,1}, d_{L,2})\}$. The first two criteria concern the segment level, i.e. matching a single pair of segments (d_i, r_j). First, the angular deviation between the normalized direction vectors determined by $d_{i,2} - d_{i,1}$ as well as $r_{j,2} - r_{j,1}$ is checked, and the matching is rejected if it exceeds d_{ang}. The next step involves orthogonally projecting d_i onto r_j (see Fig. 6.5). This yields the line position interval $[t_{start}; t_{end}]$ that specifies which part of r_j is covered by the projection. The average distance between d_i and r_j on this interval is then calculated. This leads to the second criterion: the average distance must lie below a threshold value d_{proj}. The last criterion is applied at the object level: only such matchings between R and D are allowed where at least p_{cov} of the length of D can be projected onto some $r_i \in R$ such that the average distance and angular deviation criteria are fulfilled. Comparing each reference tree with each detected tree yields a set of potential, possibly conflicting, matchings M. Note that for each matching, the intervals of each reference segment which are covered by an associated detected segment can be exactly calculated. Two versions of the evaluation procedure have been developed based on the target data. For ALS point clouds, the reference data were obtained from field works (see Sec. 6.2.2). To account for some discrepancies between the positions measured during the field inventory and the ALS point coordinates, a value of $d_{proj} = 0.55m$ was used. In order to compensate for this setting, a strict angular deviation threshold of $d_{ang} = 5°$ and coverage threshold of $p_{cov} = 0.7$ was set. Also, each detected stem was allowed to be matched to at most one reference tree, but a reference tree could be matched to multiple detected trees provided that the line position intervals covered by each matched tree did not overlap. Regarding TLS data, the reference stems were derived from the point clouds, and therefore the mean projected distance d_{proj} was set to a far stricter value of 10 cm. This already constrained the matched tree pair count quite strongly, which allowed a more lenient setting of the remaining criteria: $d_{ang} = 10°, p_{cov} = 0.1$. The higher angular deviation was chosen to handle very short stems (below 1 m) whose reconstructed cylinder orientations deviated from the reference direction due to occlusions or low point density. Moreover, for the dense data scenario the restriction of each detected stem being matched at most once was removed, and instead non-conflicting n-to-m matchings were permitted. In formal terms, define a tree-level matching $M^{ij} \rightarrow (R^i, D^j)$ as the set of all matching pairs (r_k^i, d_l^j). Two tree-level matches M^{ij}, M^{kl} are considered non-conflicting if there is no reference segment r with a projection interval covered both in M^{ij} and M^{kl}, and also no detected segment d exists with this property. In other words, matchings are non-conflicting if there is no overlap between their covered intervals of both reference *and* detected stems. In practice, conflicting assignments were rare, because the matchings were constrained quite well by the three criteria. For both the ALS and TLS case, the conflicts were resolved in a greedy fashion, always picking the assignment with the longest covered interval of the reference tree.

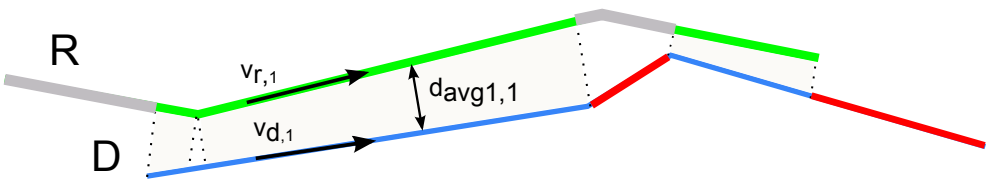


Figure 6.5: Projecting a detected fallen tree D onto a reference tree R. Parts of D could not be projected or were excluded due to excess angular deviation and average distance (drawn in red). The pieces of R covered and not covered by the detection result are marked green and gray, respectively.

Quality measures

The matching between detected and reference trees obtained using the aforementioned procedure forms the basis for calculating several metrics which describe the quality of the obtained detection result. The first metric is *correctness*, which expresses the number of detected trees that were successfully linked to reference trees as a fraction of the total number of detected trees. The next metric, *completeness*, is defined as the ratio of the number of reference trees which have at least one associated detected tree to the total number of reference trees. However, it is desirable to leverage the detailed coverage information provided by the tree matching algorithm, and therefore a modified version of this metric is introduced: *completeness at level p%*, which is defined analogously but with the condition that only reference stems whose length is covered to at least p% by detected segments are counted in the numerator. Finally, the *total length completeness* is specified as the ratio of the sum of the lengths of all covered intervals of the reference trees to the sum of lengths of the reference trees.

6.2.2 ALS point clouds

Reference data

Ground truth data for the Plots 1-5 was obtained using field measurements. In each of the 5 areas, at least 30 stems lying on the ground with a diameter above 10 cm were measured. For every stem, the position of its endpoints along with the diameters at both endpoints was recorded by tacheometric means augmented by differential GPS measurements with an accuracy of 2 cm in the local coordinate system. The relative accuracy between the laser points and reference data was influenced by the use of different parameters for transforming the ALS and reference coordinates into the Gauss-Krüger system, which resulted in minor differences in position and heading of the fallen stems. If a tree consisted of more than one part, each linear segment was processed independently. The semantic labels which group individual segments into trees were later assigned based on inspection of the relative segment positions and orientations using visualization software. The measurement campaign was carried out in July 2014, which constitutes a temporal offset of 3 years and 3 months with respect to the ALS campaign. Visual comparison of the reference stem positions with the ALS point clouds revealed that a significant portion of the recorded stems is not at all represented in the point clouds (particularly for the coniferous plots). This is associated with the temporal offset, but also with a significant (66%) overstory cover. There were also cases where a fallen stem was strongly represented in the ALS point cloud, but missing from the reference data, because not all stems in each plot were measured. Both the unrepresented reference stems and the mismatched stem points were removed from the analysis. The properties of the filtered reference data are given in Table 6.3.

Property	Plot 1	Plot 2	Plot 3	Plot 4	Plot 5
Num. fallen trees	33	37	23	35	16
Mean diam. at center [m]	0.33 ± 0.07	0.33 ± 0.08	0.34 ± 0.11	0.25 ± 0.05	0.23 ± 0.04
Mean length [m]	19.0 ± 6.9	14.3 ± 7.8	19.1 ± 7.8	18.0 ± 5.6	16.6 ± 7.0

Table 6.3: Properties of reference data for fallen tree detection from ALS. Mean diameters and lengths are given ± standard deviation.

Parameter settings

In the DTM generation and filtering step, the DTM grid size was $d_{DTM} = 10$ cm, while the height thresholds were set to $h_{min} = 10$ cm, $h_{max} = 150$ cm, and the smoothing coefficient $\alpha =$ equalled 7. The low and high thresholds were selected to reflect the typical range over the ground where most fallen trees are found. The lower bound, 10 cm, lies below the smallest expected fallen tree diameter to avoid removing relevant stem points due to DTM overestimation while filtering out the ground points. The parameter values for candidate generation were: $p_{thr,p} = 0.5$, $r_{seg} = 0.3$ m, $l_{seg} = 3$ m, $c_{thr,1} = 30$ %. The cylinder radius is based on the largest expected radius of a fallen tree, augmented with a 5-10 cm buffer. Similarly, the segment length approximates the minimum expected length of a fallen stem in the target area. The neighborhood parameters in the segment merging step were $l_{max} = 10$ m and $r_{max} = 2.4$ m. The radius is relatively large to account for potential differences in heading among the segments of the same stem, which are expected to appear due to the sparsity of the point cloud. Finally, in the skeleton-based stopping criterion for Normalized Cut (Sec. 3.1.3), values of $k = 3$, $q = 0.8$, $c_{thr,2} = 50$ % and $r_{thr} = 50$ cm were used. The maximum number of stem parts was set to 3, to handle stems which fractured into multiple fragments. All parameter values were reused for all test plots, except the DTM smoothing factor for Plot 3 which needed to be reduced to account for a road passing through the middle of the plot.

Training point-level classifier

To learn the stem point probability model, the two additional plots T1 and T2, distinct from the test plots, were used. The two plots were processed with the DTM filter, and subsequently a total of 37000 laser points were manually marked by visual interpretation with the label *stem point* or *other point*. The classifier used was kernelized logistic regression (KLR) with L2 regularization. The two metaparameters, Gaussian kernel bandwidth and regularization coefficient, were determined through grid search on an exponential grid, using 5-fold cross-validation and Cohen's kappa coefficient as the error measure.

Training CRF components

In order to derive training data for the probabilistic models forming the implicitly constrained CRF (Sec. 4.1.2), also the two plots T1 and T2 were used. First, a number of segment candidates in both plots was generated as described in Sec. 4.1.1. Then, 584 of them were manually labeled as either *true stem segments* or *other objects*. This data was directly used for training the segment appearance model P_{app} (Sec. 4.1.2). The difference in the distribution of the Shape Context features for stem segments versus other objects is illustrated in Fig. 6.7. The histograms show a repeating pattern every 18 (3 radial times 6 angular) bins, which suggests that the feature distribution is similar in all 10 intervals along the segment's axis. Moreover, the segments were used to delineate 180 fallen stems by visual inspection, obtaining the stems' individual laser point subsets. For each stem, the skeleton of its points was calculated (Sec. 4.1.3). The set S_{skel} of segment candidates which lie close to any skeleton part was then found, where the proximity is quantified through a maximum angular deviation in direction of 6° and maximum average segment distance (Sec. 3.2) of 7 cm. The elements of S_{skel} were used as the basis for training the centrality and collinearity models as follows. In case of the unary segment centrality model (Sec. 4.1.2), these segments were taken as the positive examples for training P_{ctr}, whereas the negative examples consisted of candidates not meeting the two aforementioned criteria. For both unary probability models P_{ctr} and P_{app}, kernelized logistic regression was applied, similar to pointwise classification. To train the pairwise collinearity model (Sec. 3.1.2), all segment pairs $s, t \in S_{skel} : s \sim t$ which

belong to the neighborhood relation $\sim$ (Sec. 3.1.2) were used for constructing the kernel density estimator P_{kde}. The learned KDE model is visualized in Fig. 6.6.

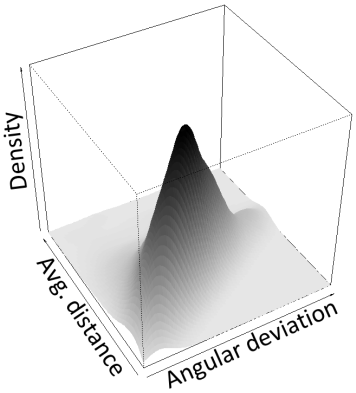


Figure 6.6: Learned KDE model of segment collinearity. Axes represent two differential features of segment pairs.

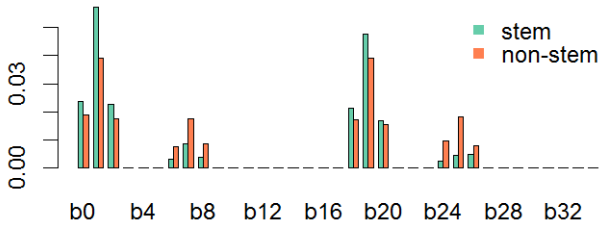


Figure 6.7: Differences in statistical distribution of Shape Context features for stem segment probability calculation.

Evaluation of sensitivity to parameters

In the first experiment, the performance of the entire fallen tree detection workflow was assessed as a function of two important parameters of the algorithm: the minimum segment support threshold s_{thr} (Sec. 4.1.1) and the DTM grid width d_{DTM}. The goal was to find out how reliably the fallen stems can be detected under the conditions outlined in Table 6.1. In particular, the existence of a trend in the performance for coniferous vs. deciduous plots as well as the dependence on overstory cover or stem diameter were investigated. To eliminate the influence of the multiple energy balance coefficients γ, β within the CRF formulation (Eq. 3.11), a simplified version of the processing pipeline was used where the pairwise terms were inactive ($\forall_k \beta_k = 0$). The segment selection procedure boiled down to setting a threshold value p_{thr} on the segment appearance probability P_{app} (Sec. 4.1.2) and picking segments with P_{app} exceeding the threshold. Since this strategy led to many redundant segments whose quantity was impractical for directly conducting the merging step, a pruning procedure was carried out to reduce the number of chosen segment candidates. Specifically, for the set of initially selected segments S, an instance of the classic set cover problem was defined on the point level, with the goal of finding a minimal subset $S' \subset S$ such that all points belonging to the cylindrical volume around any $s \in S$ are also covered by at least one $s' \in S'$. This problem is known to be NP-hard, therefore an approximate randomized algorithm, the Meta-RaPS heuristic [Lan et al., 2007], was applied. The processing pipeline was thus executed from the start up to and including the segment merging step, with the Ncut similarity function and stopping criterion learned from simulation as described in Sec. 4.1.3. The computations were conducted on a sequence of equidistant probability threshold values p_{thr} over

the interval [0.2;0.8], giving rise to a Receiver Operating Characteristic (ROC) curve. Due to the non-deterministic nature of Meta-RaPS, each point on the ROC curve was based on the best of 5 random restarts. The curves are displayed at several different levels of completeness to illustrate what part of the trees (in terms of length coverage) is detected. The experiment was repeated for following values of the parameter s_{thr}: 10, 15 and 20 points. The investigated DTM grid sizes d_{DTM} were 10, 50 and 100 cm.

Evaluation of learned stopping criterion and similarity function

This experiment focused on the effects of learning the Normalized Cut similarity function and the stopping criterion (see Sec. 4.1.3) on the final detection result for each plot. For this purpose, the processing pipeline steps were executed from the initial DTM filtering up to and including the simplified segment selection using P_{app} and set cover as described in the previous section. Next, 4 variations of the merging step on the same set of segments were performed. These 4 variations correspond to the 4 possible situations when the similarity function and stopping criterion are either obtained from learning via the simulation, or from a coarse grid search. In case of the similarity function, the grid search is on the space of 3 feature weight categories discretized on the interval [0.2;1] with steps of width 0.2, resulting in a total of $5^3 = 125$ configurations. For the stopping criterion, the grid search boils down to examining 20 Ncut threshold values from 0.01 to 0.20 with step 0.01. Note that in case when both the similarity function and the Ncut threshold are taken from the grid search, due to an overwhelming computational effort (2500 possible configurations) the search was reduced to the Ncut threshold while reusing the best weights from the configuration where the stopping criterion was the learned one. Since the input set of segments was the same for all variations and the clustering algorithm is deterministic, the resulting segmentations depended only on the feature weights and stopping criterion and hence no configuration was favored over the others. Note, however, that the grid search configurations had an unfair advantage over the learning-based ones, because the former were optimized on the test data, whereas the latter were only trained once on independent data. As in the previous experiment, a ROC curve was obtained by varying the probability threshold p_{thr} for segment-based classification and re-running the experiment.

Evaluation of benefit due to CRF

The purpose of the next experiment was to quantify the improvement in the final dead tree detection quality attainable when the CRF-based contextual classification method (Sec. 3.1.2) with pairwise interactions is used for segment selection instead of the simplified procedure based on thresholding and set cover described above. Specifically, for each test plot two series of computations were performed. First, the entire processing pipeline with icCRF was executed on an exponential grid of CRF component coefficients $\beta(0), \beta(1)$ (Sec. 3.1.2) corresponding to the pairwise collinearity and spatial overlap potentials, with coefficients assuming values from the set $\{10^{-5}, 10^{-4}, \ldots, 10^{5}\}$. The coefficients $\gamma(0), \gamma(1)$ related to the unary appearance and centrality potentials were fixed at 1. For every coefficient combination, the icCRF was solved at a sequence of λ values of the constraint term, increasing exponentially by a factor of 1.2. Each set of selected segment sets was processed further with the merging step. The quality metrics (correctness, total length completeness) were computed for all results. In this experiment, no attempt was made to identify the single best result, so the ranking step was not executed. Instead, a Pareto-optimal (non-dominated) front in the 2D space of the metrics was computed, obtaining a ROC curve. To limit the number of processed configurations, the test scenarios were not split into networks (Sec. 4.1.4). In the second part of the experiment, the baseline ROC curves were obtained for the case when the simplified, set-cover based segment selection method was used in place of icCRF.

Like in the previous two experiments, this was done by varying the minimum appearance probability threshold p_{thr} for selecting segment candidates. Since the input data, evaluation criteria and the remaining parts of the processing pipeline (excluding segment selection) were the same for both scenarios, any changes in the quality of final tree detection results should be attributed only to the icCRF-based segment selection method.

Evaluation of fully automated processing

The goal of the final experiment was to investigate the performance of the proposed method in a real-life scenario where the optimal CRF coefficients are not known. For this purpose, the processing pipeline was executed on a set of coefficient values like in the previous experiment, this time applying the partitioning into segment networks (Sec. 4.1.4). Once again, the unary potential coefficients were fixed at $\gamma(0), \gamma(1) = 1$, while for the pairwise coefficients only $\beta(0)$ (collinearity term) was varied, with $\beta(1)$ set to $10\beta(0)$ to reduce computation times. For the same reason, all segment candidates having P_{ctr} below 0.3 were removed from consideration. The scene ranking method (Sec. 3.1.4) was used for obtaining the final result for each plot as the highest-scoring segmented scene. To adhere to the fully automatic processing constraints, the tree/network appearance models for scene ranking were trained based on a cross-validation scheme. When processing a test plot, the remaining four plots were used to generate training data for the appearance models. This was done by executing the entire processing pipeline for all considered coefficient configurations, and matching the detected trees with reference measurements using the evaluation criteria outlined in Section 6.2.1. Positive examples were then derived from objects successfully matched to reference trees, and negative examples - from the remainder. The random forest (RF) classifier was applied due to its high performance and ease of tuning metaparameters (number of grown trees, number of variables sampled at each tree node). The number of labeled networks per plot ranged from 18 to 48, whereas the number of labeled tree parts and entire trees per plot varied from 7600 to 11600 and 3200 to 5300, respectively. Finally, to verify that any change in the final result quality is due to the icCRF segment selection as opposed to arising from the partitioning into networks or the introduction of the segment centrality model, a baseline experiment was conducted. The baseline consisted of applying the previous non-pairwise segment selection method (based on set cover) for a variety of minimal segment probabilities P_{app}, P_{ctr} on all segment networks. The merge results were evaluated by the same scene ranking appearance models as for the main experiment (icCRF-based), which enables a quantification of the part of the difference in performance directly attributable to the segment selection.

6.2.3 TLS point clouds

Reference data

The reference stem positions were derived manually from the TLS point clouds by visual interpretation. Specifically, point clusters belonging to single stems were delineated, followed by applying RANSAC cylinder fitting to each individual cluster and adjusting various parameters of the fitting procedure until a visually consistent result was obtained. Only objects with a length above 0.5 m exhibiting a cylindrical shape were considered. Quite frequently only parts of the entire stem surface were captured due to occlusions and the choice of scanning locations, therefore the cylinder radius estimation from RANSAC may be associated with significant uncertainty. This is also the reason why a significant number of the measured stems is shorter than the expected in reality. Consequently, no comparisons on the basis of the approximated radii were performed, and mainly the cylinder axes were used in the role of reference data. Also, it should be noted that for Plot B, the high tree density (4 times that of Plot A and 2.5 times that of Plot C) resulted in significant occlusions and substantially reduced the point density near the ground. As a result,

the determination of reference trees in that plot is the most subjective among the three. Table 6.4 summarizes the characteristics of the derived reference data. The reference cylinders created for Plot A are displayed in Fig. 6.8.

Figure 6.8: All reference tree cylinders derived for Plot A. The dark blue cylinders represent hypotheses from RANSAC, other colors show point clouds of individual trees.

Property	Plot A	Plot B	Plot C
Num. fallen trees	61	47	29
Median stem diameter [cm]	47 ± 11	25 ± 3	27 ± 8
Median stem length [m]	3.2 ± 2.4	1.2 ± 0.4	1.5 ± 0.8
Median points per stem	1357	285	4047

Table 6.4: Properties of reference data for fallen tree detection from TLS. Median diameters and lengths are given ± median absolute deviation.

Parameter settings

The DTM filtering interval was set to h_{min}=5 cm, h_{max}=1 m with a DTM grid size of 10 cm. The voxelization width d_{vox} equaled 1 cm. The minimal cylinder length and point count used in the connected component segmentation as well as in the cylinder pruning step were l_{min}=0.5 m, n_{min} = 30. Regarding the distance threshold for connected component segmentation d_{max}, a value of 10 cm was applied in case of Plots A & B, while for Plot C the threshold was decreased to 5 cm

due to the higher point density. For calculating surface normals a spherical neighborhood with radius $r = 10$ cm was used. The same value was re-used as the neighborhood size for aggregating the cylinder axis and position votes (Secs. 3.3.2, 3.3.3) with the spatial median. The minimal distance d_{min} between surface normals voting for cylinder positions was fixed at 3 cm, whereas the allowed range for cylinder radius values was defined as r_{min}=5 cm, r_{max}=50 cm. A value of 2 cm was utilized for the distance threshold d_{in} which determined the cylinders' inlier points. The feature weights σ_i for Normalized Cut were obtained empirically by a grid search on a subset of cylinders generated for Plot A. They remained fixed throughout all experiments.

Training point-level classifier

To train the classifier, a number of both stem points and points representing irrelevant objects (ground vegetation, DTM artifacts) were labeled within the point clouds of each test plot, based on visual interpretation of the data. The total number of labeled points was 740k, 214k and 520k respectively for Plot A, B and C, with points distributed equally among the two classes (stem/other). The random forest classifier was applied due to its minimal requirements for tuning metaparameters, although any other binary classifier with a probabilistic output could be used in its place. The neighborhood radius for feature calculation r_n was set to 0.12 and 0.14 respectively for the covariance and FPFH features. These values were determined using cross-validation with Cohen's kappa coefficient as the error measure.

Evaluation of performance - comparison with SAC

This experiment aimed to compare the quality of generated cylinder primitives obtained from the proposed statistical detection framework (Sec. 3.3) versus a SAC baseline. To that end, the entire pipeline was executed for two settings of the cylinder detection relative local maximum parameter $k_{kde} \in \{0.4, 0.7\}$. Each k_{kde} setting was investigated in two configurations: (i) uniformly-weighted voting and (ii) voting with weights derived from stem point probabilities (see Sec. 3.3.5). The cylinder generation step was subsequently replaced with a RANSAC cylinder fitting procedure which had access to the same input data (point cloud and surface normals). The chosen version of RANSAC required two samples of points with corresponding normals to determine the shape parameters [Rusu & Cousins, 2011]. Its inlier criterion was based on a linear combination of Euclidean distance of the point to the considered cylindrical surface and angular deviation between the theoretical and actual surface normals at that point. Furthermore, the RANSAC procedure was altered to also make use of point-level probabilities: instead of using the usual inlier count, the quality of a model is assessed based on the sum of fallen stem probabilities of all inlier points. The altered pipeline was re-run using three levels of the angular deviation's relative influence on the combined inlier distance: 0, 0.1 and 0.3. Note that since SAC-type approaches yield only the best-fitting shape, it was necessary to proceed in an iterative fashion, removing inliers around each found cylinder and re-running RANSAC until the best candidate failed to meet minimal size criteria. All other processing steps were left unchanged.

Evaluation of point classification transferability

In the other two experiments conducted for TLS, the point-level classifier which provides stem point probabilities was, for each processed plot, re-trained using point samples from that plot. In this experiment, it was investigated whether the classifier can generalize well across different test plots. For this purpose, for each plot a classifier trained on sample points from the other two plots was applied to obtain the point probabilities, which were then used as a basis for further processing. Here, both the uniform and weighted parameter voting variations were considered.

Evaluation of surface normal influence

Finally, the influence of the method of surface normal calculation on the final performance was examined. The processing pipeline was altered to use the standard local plane fitting method for estimating surface normals, with the rest of the steps unchanged. The detection quality was compared to the case when the normals are calculated by fitting quadric surfaces as described in Sec. 3.3.1. This experiment was based on uniformly-weighted parameter voting.

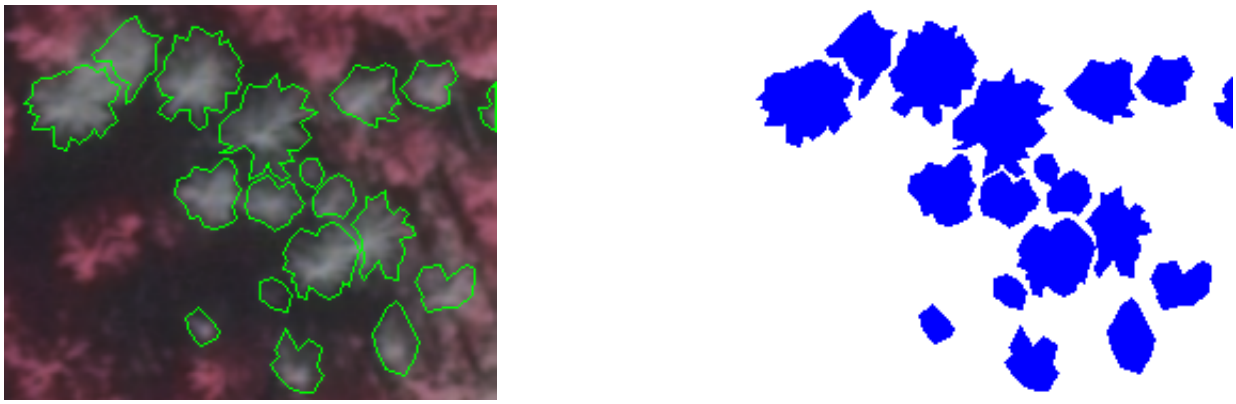
6.3 Standing dead trees

Here, the experimental design for the case of standing dead trees is outlined. First, the scenario from Section 5.1 is treated, where only aerial imagery data is available. Since the task involves delineating tree crowns by defining polygons, a specialized measure of polygon overlap is used to gauge the quality of the results. The next two subsections deal with scenarios where the set of tree polygons is obtained a priori and the task is that of standard classification. The emphasis is placed on the evaluation of feature extraction and learning techniques, which allows the use of classic metrics for object detection. Specifically, the second part of this section is concerned with evaluating the Free Shape Context descriptors in the role of features for detecting standing dead stems from ALS data (as described in Sec. 5.3), whereas the final part focuses on assessing the combined active and semi-supervised learning framework introduced in Sec. 3.2 in the context of dead tree detection from both aerial imagery and ALS data.

6.3.1 Aerial imagery - active contours

Reference data

The reference data for validating the results was obtained by manually marking the contours of the individual dead trees on the three test images. The contours were subsequently converted to polygon masks in order to enable a pixel-based comparison (see Fig. 6.9). The number of delineated tree crowns was 18, 23 and 83 respectively for Plots 1,2, and 3.



(a) (b)

Figure 6.9: Reference data for dead tree crowns created using manual delineation in image. (a) Contours of marked polygons. (b) Polygon masks created from contours.

Evaluation criterion

The polygon mask representation (Fig. 6.9b) is utilized to perform a precise comparison of the detected and reference objects at the pixel level. A reference snag o_R and a detected snag o_D are

considered as potentially matched if their polygon masks have a non-zero pixel intersection. Note that due to over- or under-segmentation, there may generally exist a many-to-many relationship between detected and reference objects. To better describe the quality of a matching (o_R, o_D), it is equipped with two quality metrics. The first metric is the Euclidean distance d_{ctr} between the centers of gravity of the reference and detected polygons, which are calculated as the average planimetric coordinates of all pixels belonging to the respective polygon. The second metric is the Dice coefficient DSC defined on two sets of pixels A, B as: $DSC(A,B) = \frac{2|A \cap B|}{|A|+|B|}$. The Dice coefficient is normalized on the interval $[0;1]$ and measures the similarity of two sets, with a value of 0 indicating no overlap and a value of 1 indicating set equality. Two strategies are employed to evaluate the quality of the entire detection result: a strict 1-1 correspondence strategy and a lenient N-M correspondence strategy. In the first case, a 1-1 matching is sought between reference and detected trees. To that end, a bipartite graph $G = ((V_R, V_D), E)$ is constructed with vertices in V_R and V_D corresponding to reference and detected snags, respectively. Each edge $e \in E$ connects a reference and detected vertex. The edge weight, which can be regarded as the assignment cost, is set to the centroid distance between the corresponding polygons if the polygons intersect, and $+\infty$ otherwise. The minimum cost assignment is found using the classic Kuhn-Munkres algorithm. The 1-1 strategy incorporates information about under/over-segmentation as well as reference snags which were completely missed and detected snags which did not overlap with any reference polygon. The second strategy allows an N-M relationship between reference and detected snags. It considers all potentially matched object pairs (in the sense of pixel overlap) as matched. This strategy is only informative in terms of entirely missed reference and detected dead trees and does not quantify over/under-segmentation issues. However, it is still meaningful to apply it in the target scenario, because both the reference and detected polygons are disjoint by construction. For both strategies, the correctness and completeness measures are calculated on the resulting assignment, which are defined as the ratio of the number of matched pairs to the number of, respectively, detected and reference polygons. Finally, the average per-pixel correctness and completeness is introduced. For the pair (o_R, o_D) they are defined, respectively, as the ratio of the pixel intersection size between o_R, o_D to the number of pixels in the polygon of o_D and of o_R.

Training shape & intensity priors

As the basis for the experiment, a total of 96 training polygons corresponding to dead trees in were marked in CIR images distinct from the test plots. Polygons representing diverse lighting conditions and camera perspectives were selected in order to obtain a reasonable coverage of the target objects' appearance. The set of training polygons was then aligned and used as a basis for deriving the eigenshape representation (see Sec. 5.1.2). The number of used eigenmodes of shape variation was $c = 20$, which represented a total of 98.4% of the variance associated with the original shape variability matrix S. Also several regions containing only shadows and only living vegetation were marked in the training images, with a total of ca. 20000 pixels. The mean pixel intensities of shadow, living vegetation and snag regions were used as the mean priors in the candidate region detection step (Sec. 5.1.1). The labeled dead tree and background polygons were re-used for training a kernelized logistic model for use as the intensity prior (Sec. 5.1.2). A total of $T = 3000$ pixel values were utilized as the KLR training points y_i, with the kernel bandwidth selected using 10-fold cross-validation. The overall accuracy of the KLR model obtained from cross-validation was 0.81.

Evaluation of dead tree crown delineation

The experiment was carried out on Plots I-III by first executing the coarse segmentation step (Sec. 5.1.1) to find image regions likely to contain dead trees, and then following the iterative fine

segmentation scheme as outlined in Sec. 5.1.3 over the selected regions. The radius of the circles which constituted the initial tree crown approximations was derived from the size of the mean training shape (see above) and had a value of 1.2 m.

6.3.2 ALS point cloud - Free Shape Contexts

Reference data

To create a data set for assessing the performance of each feature family, first the ALS point cloud was segmented into individual trees using the method of Reitberger et al. [2009]. This method has already been previously investigated and evaluated, therefore its evaluation was not the object of this study. A total of 285 objects, corresponding to the segmented trees, were then chosen and labeled to serve as the basis for this experiment, of which 56% were living trees and 44% were dead trunks. The labeling was done through visual interpretation of the 3D point cloud. The average point counts in the segmented point clouds of the objects were 340 and 208 respectively for living trees and dead trunks. The selection criterion was associated with ensuring a possibly broad coverage of both classes' appearance ranges within the ALS point cloud.

Evaluation criterion

The quality metric for the detection results was the classic measure of overall classification accuracy, obtained from leave-one-out cross-validation on the dataset.

Evaluation of performance - Free Shape Contexts

The goal of the experiment was to compare the performances of the proposed Free Shape Contexts, standard (uniform structure) Shape Contexts, and features based on eigenvalues of the covariance matrix (referred to as CE). To understand what part of the potential gain in classification accuracy is associated with the features as opposed to being merely a result of applying a better optimization strategy (genetic algorithm vs. grid search), each feature family was tested in two configurations. The first variation uses parameters optimized by grid search (in case of FSC the dynamic algorithm described in Sec. 5.3.2), while the second configuration is based on GA parameter optimization. In all GA-based experiments, the same population size (1000) and generation count (150) as well as selection mechanism (size 5 tournament selection) were applied. Because the GA optimization result is non-deterministic, the experiments were repeated 15 times from random initializations. For FSC, the maximum allowed number of member SCs was 12, each with up to 5 radial bins. The meta-parameter settings (Sec. 5.3.3) were: $p_{m,1} = 0.03$, $p_{m,2} = 0.15$, $\sigma_m = 0.1$, $p_{FR} = 0.6$. In this experiment, the FR width was fixed at a value of $d_{FR} = 0.05$. This value was chosen empirically from a small set of discrete levels based on the results of preliminary experiments with smaller population sizes and generation counts. For the evolution of standard SCs, a vector of 5 real numbers was used, representing the parameters r_0, r_B, l, n_l, r_{max} as the SC's genotype, with two-point crossover and Gaussian mutation (such as the size parameter mutation in Sec. 5.3.3). Finally, for the CE features, a simple variation of the Bag-of-Features model [Toldo et al., 2009] was applied. For each point within the input point cloud, the values of the 8 covariance eigenvalue-based features (Sec. 2.6), defined in [Weinmann et al., 2015a], are calculated. The domain of each feature i is subsequently partitioned into b_i equidistant bins. Each bin defines a 'visual word' for the associated attribute. The descriptor of the entire scene consists of the decorrelated, concatenated histograms of visual word counts for each feature over all points within the scene. The parameters of this model are the numbers of feature partitions b_i as well as the common neighborhood radius for calculating the covariance matrix around each point. Note that a value of 0 partitions causes the feature to be omitted from analysis, which enables a simultaneous feature

selection. A similar strategy for genetic representation and recombination is applied as was the case with standard SC evolution. The grid search/approximate algorithm baseline calculations were carried out with a number of inspected configurations greater than or equal to the number of individuals processed in a single GA execution (1000 * 150 = 150000).

6.3.3 Aerial imagery & ALS - active/semi-supervised learning

Data preparation and feature extraction

To evaluate the semi-supervised and combined active/semi-supervised learning methods introduced in Sec. 3.2, two of the standing dead tree classification problems were considered: (1) detection of dead trees with crowns from ALS and CIR imagery (Sec. 5.2), and (2) the dead stem detection from ALS (Sec. 5.3). Both of these tasks operate on the individual tree clusters obtained from 3D segmentation with Normalized Cut. The preparation of data for this experiment started with selecting two independent square ares of size 1 x 1 km^2 within the ALS point cloud of the Bavarian Forest National Part (one for each test problem), and performing the Ncut segmentation. This yielded 42000 and 37000 segmented individual trees and their associated bounding polygons respectively for Problem 1 and Problem 2. Then, each object's features were extracted to serve as a basis for classification. In case of Problem 1, the 9 spectral features based on the means and covariances of pixel intensities within the object's bounding polygon were used, as described in Sec. 5.2. For Problem 2, the Free Shape Contexts were employed (Sec. 5.3.2). As opposed to the investigations of the previous section (Sec. 6.3.2), here the FSC structure and parameters remained fixed throughout the experiment, because the goal was not to assess the FSCs' performance as a descriptor, but rather to evaluate the learning mechanism itself. The utilized FSC consisted of 13 features.

Reference data

As a basis for the performance evaluation, four labeled datasets per problem were created. Of the four, two were constructed using independent runs of the greedy pre-selection procedure based on low-rank matrix approximation (Sec. 3.2.1) and contained 1000 objects each. They will be referred to as G_A^i and G_B^i, where i denotes the problem index (1 or 2). Furthermore, two additional datasets per problem, denoted R_A^i, R_B^i were obtained by random sampling, with object counts of 1000 and 2000, respectively. The labeling was carried out based on visual interpretation of the images and point clouds associated with each segmented cluster. Table 6.5 lists the target class (dead tree/standing trunk) percentages in the used datasets.

	Dataset			
	G_A^i	G_B^i	R_A^i	R_B^i
Problem 1	21.4	22.2	7.6	7.2
Problem 2	19.9	19.3	15.7	16.1

Table 6.5: Target class (dead tree) percentages for test problems in datasets.

Parameter settings

In all experiments, a binary logistic regression classifier was used. The coefficient λ controlling the balance between the labeled log-likelihood and unlabeled entropy (Eq. 3.35) was set to 1.23, where a value of 1 means that both terms have the same influence. Regarding the greedy matrix

approximation procedure (Sec. 3.2.1), a standard Gaussian kernel was applied, with the bandwidth parameter σ set to the mean inter-sample distance. All experiments were carried out for both problems. The results were not very sensitive to the choice of λ and σ, for values of $\lambda \in [0.5; 5]$ as well as σ between the nearest-neighbor distance and the mean inter-sample distance.

Evaluation criterion

All evaluations in this experiment were done by constructing an overall classification accuracy curve as a function of training set size. The reported overall accuracy was in all cases calculated on the portion of the data not included in the training set.

Evaluation of effects of pre-selection

Active learning is a selective sampling strategy which is by design biased with respect to the actual class occurrence ratios. By applying the pre-selection procedure (Sec. 3.2.1), the class balance is potentially altered even further away from the natural priors. It is known that such a shift in the prior probabilities between the train and test sets may have adverse effects for the classifier [Lewis & Catlett, 1994]. This situation, known as *covariate shift*, occurs when the class posterior probability $P(y|x)$ is the same in both datasets, but the distributions of the covariates $P(x)$ diverge. In this experiment, the effect of pre-selecting the training examples through Alg. 1 on active learning performance is investigated. Specifically, (i) standard Expected Error Reduction and (ii) random sampling (RS) are executed first using G_A^i and then using R_A^i as the training pool. The test set in both cases is the enlarged random pool R_B^i, whose class distribution is representative of the entire data pool. In each active learning run, the classifier is given an initial training set of 8 random elements, and then the algorithm iteratively selects objects until the training set size attains 100. The results are averaged over 300 random restarts.

Evaluation of semi-supervised learning

In this experiment, the performance of the entropy-regularized classifiers (Sec. 3.2.3) is compared to the pure log-likelihood baseline. The objective is to determine the gain due to regularization and whether regularizing with Renyi entropy can yield superior results w.r.t. Shannon entropy. The investigated quantity is the accuracy of classifiers trained on labeled datasets of increasing sizes (from 6 to 50) and using the rest of the data as the unlabeled pool for regularization. This is carried out for the baseline LR model, Shannon ER, and Renyi ER for various values of the entropy parameter α (Sec. 3.2.3), with G_A^i as the base dataset. The reported overall classification accuracy is calculated on the remaining unlabeled examples in the data pool, and averaged over 3000 random restarts.

Evaluation of combined active and semi-supervised learning

The final experiment examines the active learning performance of the proposed Expected Error Reduction with Entropy Regularization method (Sec. 3.2.5). In particular, results are compared for ER-EER with various Renyi alpha values, Shannon regularization, standard EER and random sampling. The algorithm starts with an initial training set of 4 positive and 4 negative examples and is given dataset G_A^i as the unlabeled pool. The active learning is performed for all methods up to a training sample count of 100, with dataset G_B^i used for testing. The results are averaged over 900 random initializations of the training set. In this experiment, in order to avoid biasing the results by class imbalance, the counts of positive and negative examples were made equal in each pool by removing excess negative examples, so that the magnitudes of the overall accuracies best reflect differences between the learning methods.

7 Results and discussion

This chapter is concerned with presenting the qualitative and quantitative results of the various experiments listed in Chapter 6. Based on the numerical results, the strengths and weaknesses of the methods introduced in this thesis are discussed, in comparison to baseline and competing approaches when available. Also, possible enhancements are proposed which could alleviate the deficiencies associated with the current methods in future work. The structure of this chapter mirrors that of the experiment description (Chapter 6), first focusing on fallen trees (Sec. 7.1), and then moving on to the case of standing dead trees (Sec. 7.2).

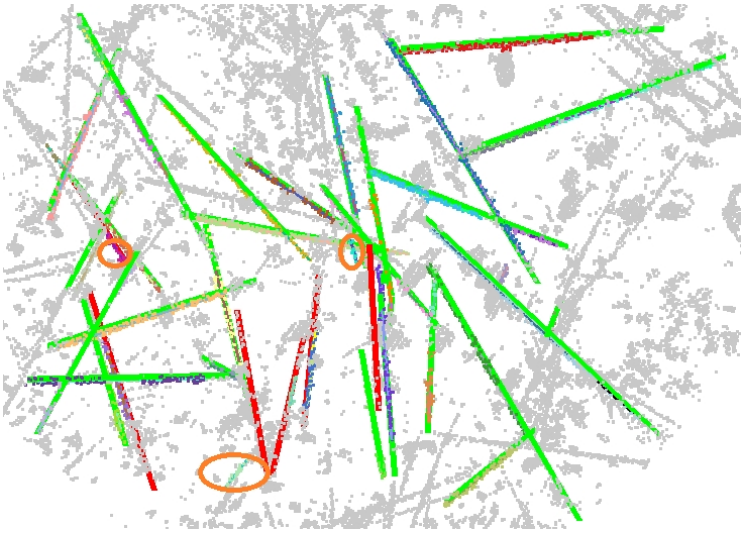
7.1 Fallen trees

7.1.1 ALS point clouds

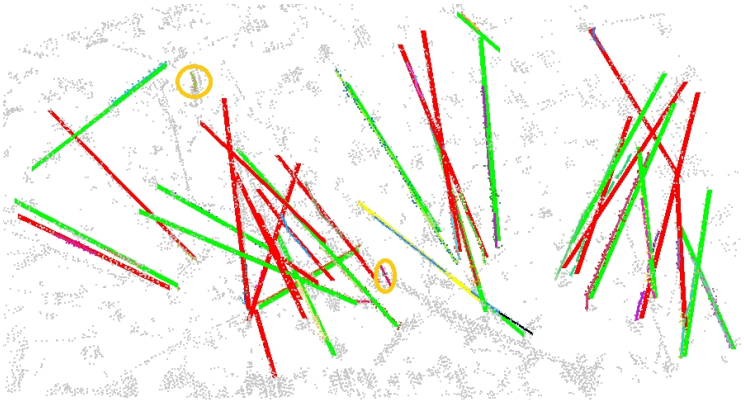
Evaluation of sensitivity to parameters

Figure 7.1 shows examples of detection results for one deciduous and one coniferous test plot. The detection performance of the entire pipeline is depicted by Figs. 7.2 and 7.3 for the deciduous (1-3) and coniferous (4-5) plots, respectively. The three deciduous datasets share the characteristic that at a detection threshold of 40% of the tree's length, over 80% trees can be detected with an accuracy of ca. 0.8. For Plots 1 and 3, one can further improve the 40% completeness to over 0.9 by trading for some correctness. The corresponding ROC curve for Plot 2 remains flat, regardless of how much correctness is traded off. At the 60% detection threshold, some more differences between the 3 beech plots become apparent. The 0.7 completeness level is attained fastest on Plot 1, with a correctness of 0.9. In case of Plots 2 and 3, a completeness of 0.65 is reached around the 0.81 correctness value. The superiority of the performance on Plot 1 asserts itself also when observing the correctness levels for which a 0.8 completeness is achieved. This value is 0.72 for Plot 1, 0.61 for Plot 3, and 0.33 for Plot 2. Also, once again on Plot 1 the ROC curve continues upwards and attains a completeness of 0.88, whereas on the other two deciduous plots, a value of 0.81 is never exceeded. The results in Fig. 7.2 also show the relationship between the minimum segment support and the detection quality. Consider the shape of the 60% ROC curves. For Plot 3, if segments of minimum 10 points are allowed, the curve peaks at about 0.8 completeness, whereas for a lower bound of 20 points, not even 0.6 is attained. In case of Plot 2, for 10-15 points a maximum completeness of ca. 0.8 is observed, while for 20 points this drops to 0.7. In contrast, Plot 1 is barely affected by the different support levels, with a completeness peak at 0.88 even for 20 points. At the 80% detection threshold, a sharp drop in completeness is seen. For Plots 1-2, only about 33% of the stems can be recognized with a good accuracy (>0.75). For the price of rather poor accuracy (<0.4), the completeness may be improved to 0.55-0.6. On Plot 3, the peak completeness of 0.48 is achieved for a quite high accuracy (0.81), but it cannot be improved further. At the 80% threshold, the difference between performances for various minimum support levels becomes visible for all 3 plots.

For coniferous plots, the attained segmentation quality is significantly lower. Even with a 20% detection threshold, the method achieves 0.69 completeness for an accuracy of 0.71 on Plot 4 and 0.75 with accuracy 0.81 on Plot 5. Increasing the detection threshold to 40%, one observes that a completeness of ca. 0.44 can be obtained with good accuracy (exceeding 0.9). The 0.6 completeness threshold is cleared at 0.6 accuracy. At the 60% detection level, only about 20% stems can be reliably (>0.9 accuracy) found. At about 0.6 correctness, a completeness of 30% can be reached. The percentage of found trees for the 80% threshold lies below 10%.



(a)



(b)

Figure 7.1: Sample fallen tree detection result for (a) Plot 1 and (b) Plot 4. The solid green bars indicate reference trees covered by a detected tree to at least 50% (a) and 40% (b) of length. Red bars indicate reference trees unmatched or matched to less than the threshold percentage. Points not belonging to a detected stem and points removed from analysis (see Sec. 6.2.2) are marked gray. Orange ellipses outline false alarms.

For Plots 2 and 4, a completeness of 1 is never attained. This means that there was a subset of respectively ca. 10% and 20% of the trees that could not be even partially detected. This is in part a consequence of the evaluation strategy and the definition of the stopping criterion (Sec. 7.1.1 provides details). The differences in performance between Plot 1 and Plots 2-3 suggest that for Plots 2 and 3, the points are simply not present in the ALS point cloud, which in turn indicates a correlation with the degree of overstory cover (Table 6.1). Indeed, Plot 1 is significantly less covered (34%) than Plots 2 (50%) and 3 (45%). Such an explanation is also strengthened

by the character of the aforementioned relationship between the minimum segment support and detection quality, where for Plots 2-3 the results deteriorated when setting a support threshold of 20 points, while remaining unaffected for Plot 1. This confirms that some parts of the stems are represented very weakly in the point cloud and cannot be reliably detected with segments having a minimum of 20 points. Note that in the context of the ALS data, segments with 10 points have an area point density of less than 6 pts/m^2, which results in rather feeble patterns that can be confused with random artifacts from the DTM. Modifying the minimum segment support parameter can thus be regarded as yet another way of effectuating a completeness/correctness tradeoff. The lower average stem length (14 vs. 19 meters) may also have contributed to the performance deterioration on Plot 2.

The task of detecting fallen trees in the two coniferous plots proved to be even more challenging than it was the case for the beech-dominated stands, which can be attributed to three main reasons. First, the average diameter of the stems was 8-10 cm smaller for Plots 4-5. Moreover, the fact that the ALS campaign was carried out in a leaf-off condition was irrelevant for the coniferous plots, which means that the canopy cover was denser (see Table 6.1). Also, there were 60-100% more trees per unit area. All these reasons contributed to complicating the detection problem by adding occlusions, making the point cloud more sparse near the ground and reducing the stem footprints in the cloud. A further reason may be that the segment classifier was trained on a plot with significantly less overstory cover, and thus it may not have learned to recognize occluded or sparse variations of the segments' appearance.

Sensitivity to DTM grid size

Table 7.1 summarizes the sensitivity analysis results of the detection performance with respect to the DTM grid width. For all plots, the correctness was best at the finest grid size (10 cm). In 4 out of 5 cases, the best completeness was also attained with the smallest grid. This is to be expected, because a coarser grid makes it more difficult to distinguish between ground and fallen stem points. When too many ground points are preserved, this may lead to the formation of random linear patterns, which negatively impacts correctness. On the other hand, removing stem points may decrease completeness. Another problem with increasing the grid size is that because the grid height is taken from the lowest point contained in the cell, a greater grid width means that the height is a local minimum of a larger ground patch, which leads to an underestimation of the DTM. As a consequence, the intuitive height thresholds for point filtering (see Sec. 6.2.2) are inadequate, as they leave most ground points unfiltered. For this study, visual inspection was used for each plot to find a new threshold value at which most ground points are removed while retaining the fallen stem points.

Comparison to existing methods

Blanchard et al. [2011] report a detection completeness of 73% and state that overclassification occurred in areas with large numbers of logs clustered in close proximity and in areas with shrub and tree canopy, but they do not provide a specific correctness value. Since their test area was chosen to have a small overstory cover, here it is compared to the test plot with the smallest cover (30%). Their evaluation strategy includes partially matched stems, therefore the results at the 50% tree length detection threshold are used for the comparison: 81% completeness and 86% correctness was attained on Plot 1. Muecke et al. [2013] conducted their study on a single test plot with dominating deciduous species. They also allowed partially detected stems to be counted in the evaluation, provided that the decay state was moderate or high. Their reported correctness and completeness of 89.9% and 75.6% is compared to own results at the 50% detection threshold for the 3 deciduous plots: (86%,81%), (84%,78%) and (83%,78%) respectively for Plot 1, 2 and

Grid width [m]	Plot 1	Plot 2	Plot 3	Plot 4	Plot 5
Correctness					
0.1	0.90	0.82	0.83	0.92	0.81
0.5	0.87	0.68	0.66	0.86	0.46
1.0	0.80	0.61	0.68	0.74	0.55
Completeness					
0.1	0.82	0.76	0.74	0.51	0.63
0.5	0.67	0.70	0.70	0.51	0.44
1.0	0.61	0.68	0.52	0.57	0.56

Table 7.1: Results of sensitivity analysis for stem detection from ALS with respect to DTM grid width. The completeness is given at the 50% threshold for Plots 1-3 and at the 30% threshold for plots 4-5.

3. Note that their test scenario contained mostly isolated stems and was somewhat simpler than the ones used in the current study. Lindberg et al. [2013] as well as Nyström et al. [2014] applied their methods on a challenging test area more similar to the own test plots. In the former study, a correctness and completeness of 32% and 41% was attained, while the latter contribution declares values of 64% and 38%. However, it is difficult to directly compare the results due to a different stem matching method and the deficiencies in the measured field data.

Evaluation of learned stopping criterion and similarity function

Next, the effect of learning the two components of the Ncut algorithm on its performance (Fig. 7.4) is examined. The ROC curves correspond to a detection threshold of 50% for deciduous and 30% for coniferous plots. For Plot 1, the 4 methods perform quite similarly in the false positive rate (FPR) interval [0;0.14], with a maximum difference of 3 percentage points. Around the FPR value 0.25, some divergence appears. The method with learned feature weights and the Ncut threshold from grid search (fL+sGS) is superior to its contenders on the FPR interval [0.25-0.45] by up to 6 and 3 percentage points (pp) in completeness and correctness, respectively. The method were both components were learned (fL+sL) is able to keep within 3 percentage points of and at times outperform the remaining two methods on the same interval. The second half of the FPR axis [0.5;0.9] is dominated by the method with the learned stopping criterion and feature weights from grid search (fGS+sL). Note that for only one point on the ROC curve, the method where both the feature weights and the Ncut thresholds are obtained via grid search (fGS+sGS) produces the best result. For Plot 2, the fully learned method performs the worst among all 4 variations on the high-correctness range of the ROC curve. However, the best performance is once again achieved not by the full grid search method, but rather by a combination with partial learning (fGS+sL). The fL+sL method can recover around an FPR of 0.28, keeping within several percentage points of the best combination. For Plot 3, all methods except the exclusive grid search combination perform similarly up to 0.2 FPR. Then, once again fGS+sL exhibits a slight advantage over the other methods. At about 50% correctness, the entirely learned method (fL+sL) lags by 9 pp. On the first coniferous plot (4), the dominant combination for the FPR interval [0;0.45] is fL+sGS. In most of this interval, the methods perform similarly, except the FPR value 0.2, where the two sL methods trail behind by ca. 6 pp. Finally, for Plot 5 the fully trained method fL+sL turns out superior up to an FPR of 0.4. The full grid search method lags behind significantly.

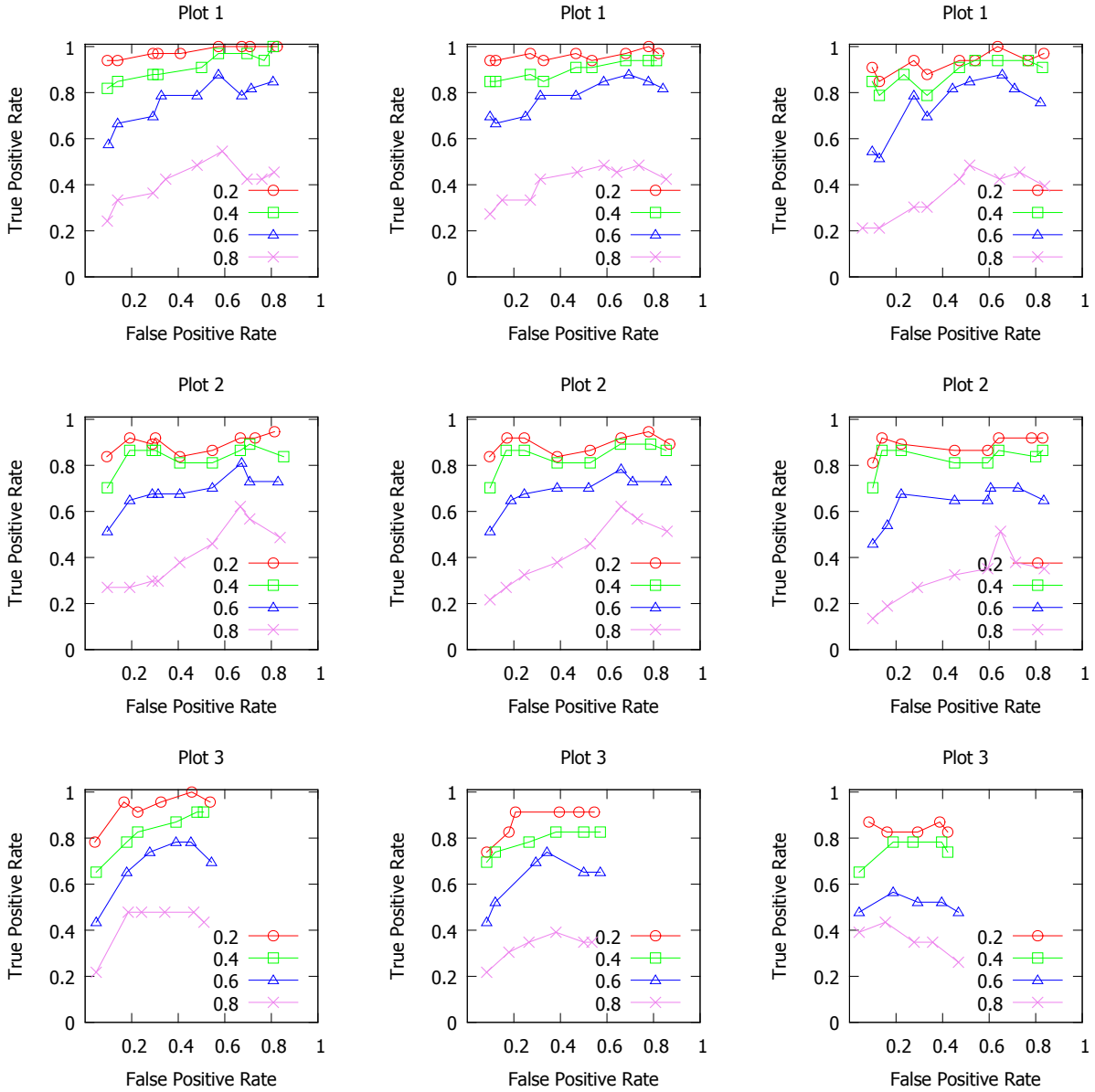


Figure 7.2: ROC curve of fallen stem detection for deciduous plots. Rows 1-3 correspond to Plots 1-3. Columns (a), (b) and (c) show results for minimum segment support of 10, 15 and 20 points, respectively. Each diagram contains 4 ROC curves which correspond to various thresholds on the percent of detected length above which the stem is considered a true positive.

On the whole, learning the Ncut components turned out to be beneficial on all plots, however the fully learned combination was not always the optimal choice. Plot 2 exemplifies a potential problem with the method fL+sL. Its inferior performance is possibly connected to a scenario in the point cloud where two stems are lying in close proximity, with a moderate angular deviation in their orientations (Fig. 7.5). The skeleton based stopping criterion terminates the merging and considers the two parts as one stem, although they are labeled as 2 separate trees in the reference data, which causes the stem to fail the matching conditions (Sec. 6.2.1). This inability to distinguish between a single tree consisting of several parts and several nearly collinear trees

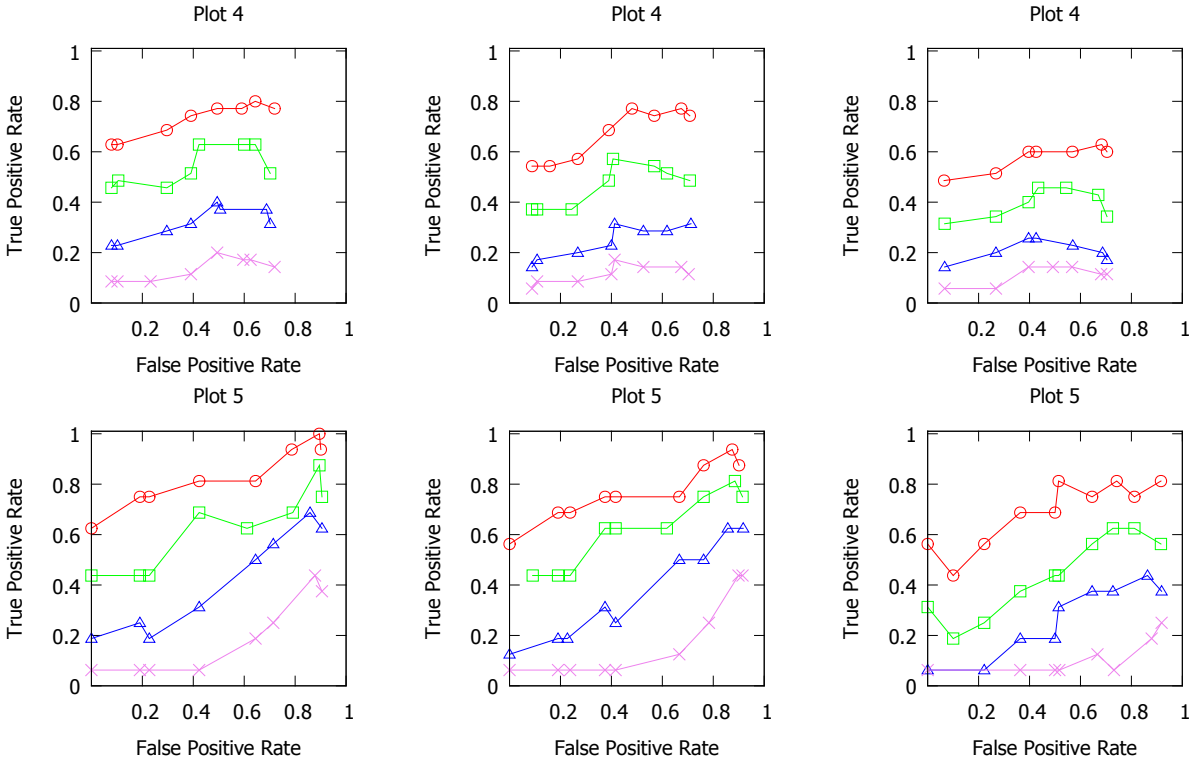


Figure 7.3: ROC curve of fallen stem detection for coniferous plots. Rows 1 and 2 correspond to Plots 4 and 5. Columns (a), (b) and (c) show results for minimum segment support of 10, 15 and 20 points, respectively. Each diagram contains 4 ROC curves which correspond to various thresholds on the percent of detected length above which the stem is considered a true positive. Legend see Fig. 7.2.

can be considered a weakness of the method, although such scene interpretation is subjective even for a human expert.

Evaluation of benefit due to CRF

Figure 7.6 presents the ROC curves describing the tree detection quality for two types of segment selection: the contextual CRF-based approach (icCRF, ◯) and the set cover baseline method (SC, □). For all test plots, the CRF-based version yields new non-dominated points on the ROC curves, offering better tradeoffs between the false positive rate and the total detected tree length. In case of Plot 1, for a false positive rate (FPR) of 0.05, the length completeness increases from 0.62 to 0.68. Also, the full completeness of 0.73 is attained at an FPR of 0.6 compared to 0.78. Plot 2 shows significant gains in accuracy, nearly doubling the completeness at an FPR of 0.07 from 0.32 to 0.63. The latter completeness is achieved by SC only at an error rate of 0.25, which constitutes a shift of 18 percentage points (pp) in favor of icCRF. Moreover, SC's final completeness of 0.68 is once again achieved significantly sooner, with an FPR of 0.41 versus the baseline value of 0.67. Regarding Plot 3, the benefit of using icCRF asserts itself mostly within the error interval [0;0.1]. A result with no false positives and a completeness of 0.34 could be obtained, whereas in case of SC the lowest FPR of 0.05 corresponded to 0.23 completeness. Note that when using icCRF, this value once again can be more than doubled, reaching 0.47 for a lower FPR of 0.04. The gain becomes less pronounced as the error rate increases, with

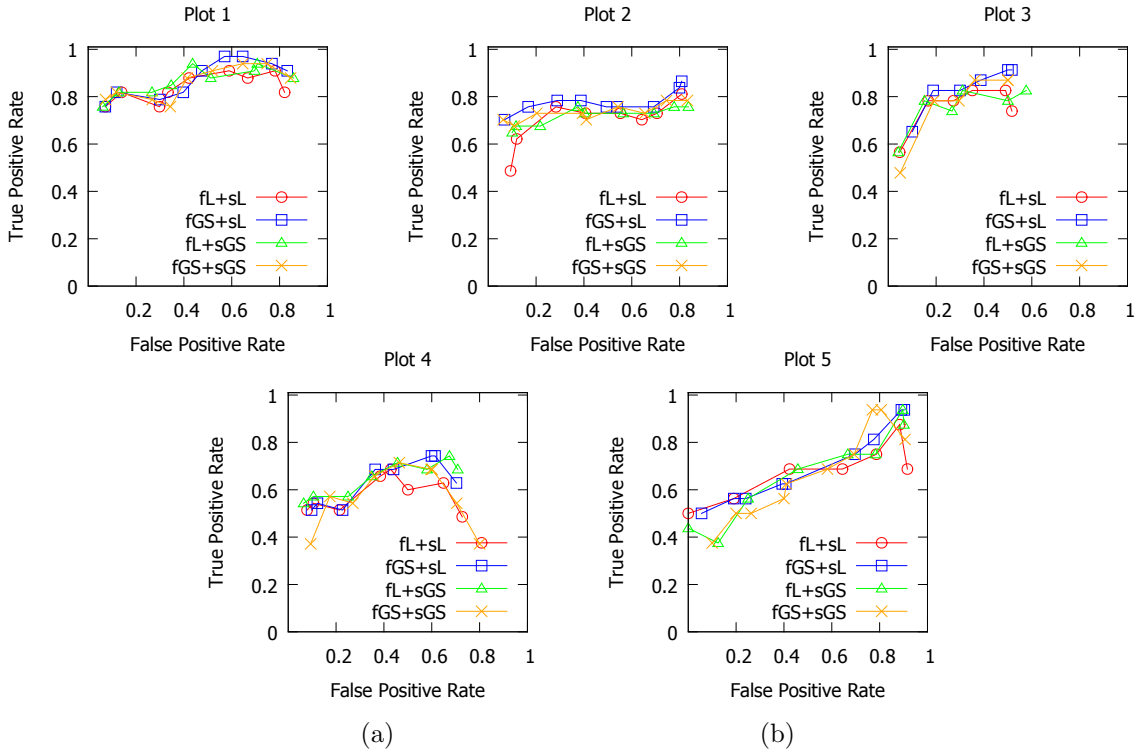


Figure 7.4: ROC curves depicting the effects of learning the Ncut similarity function and stopping criterion. The f and s prefixes refer to feature weights and stopping criterion, respectively. The l and GS suffixes indicate that the algorithm component was learned or obtained via grid search, respectively.

Figure 7.5: Difficult case for skeleton-based stopping criterion. Two red bars mark positions of reference trees. Instead, they were considered a single tree.

0.56 completeness at 0.08 error (icCRF) versus 0.45 at 0.09 (SC), to finally vanish at 0.13. This time, SC's final completeness of 0.68 cannot be reached, stopping at 0.64. In comparison to the deciduous plots (1-3), the coniferous test areas exhibit a wider FPR interval where the application of icCRF is beneficial. On Plot 4, a completely new segment of the ROC curve for error rates between [0;0.11] is obtained, with a completeness of 0.35 at 0.11. No results at all were available in this interval when SC was used as the selection method, and a similar completeness of 0.34 could only be achieved at 0.18 FPR, an offset of 7 pp. The icCRF ROC curve also dominates the baseline for error rates within [0.28;0.43], however the best completeness of 0.43 is achieved for an FPR value worse by 6 pp with respect to SC. Plot 5 shows a wide false positive rate interval positively affected by the new selection method. Increased completeness is observed at FPR from 0.07 to 0.78. The improvement starts at 5 pp, increases to 6 pp at 0.33 FPR, maintains its value up to error rate 0.67, after which point it is gradually reduced to 3 pp at 0.78. For this plot, SC's highest completeness of 0.64 is also not achieved by icCRF, whose peak completeness lies at 0.57.

The results for all plots indicate that the icCRF segmentation method is able to provide the greatest benefit within the error range 0-0.2, significantly increasing (in some cases doubling) the detected length completeness. On the other hand, for large false positive rates above 0.4, the original set cover approach may yield superior performance. A possible explanation could be

related to both methods' characteristics. The set cover approach selects the segment candidates in such a way that every laser point belonging to any segment with appearance probability P_{app} must be covered by a member of the chosen set *at least once*. This method does not include a concept of segment collinearity, neither does it try to prevent the selection of overlapping candidates. This is exemplified by Fig. 7.7a, where many of the chosen segments strongly overlap while having divergent directions. This can unnecessarily complicate the merging process, which relies, to a large extent, on the collinearity of nearby segments to categorize them into the same tree. As a result, false positives, usually of short length, arise more often. Conversely, this lack of spatial constraints apparently leads to a higher detection completeness ('paid for' with a higher error rate), allowing to find additional fallen stems which lie very close to each other. In contrast, the proposed icCRF method makes explicit use of both a collinearity and an overlap model for adjacent candidates, which combine into a prior that favors well-aligned linear structures (Fig. 7.7b). The potential for merging error is thereby reduced, at the cost of not detecting the most unusual nearby tree configurations. Finally, it is probable that the icCRF results shown here could be further improved by performing a more thorough search of the energy coefficient space (terms α, β, γ in Eq. 3.11), which was restricted to a sparse exponential grid in this study.

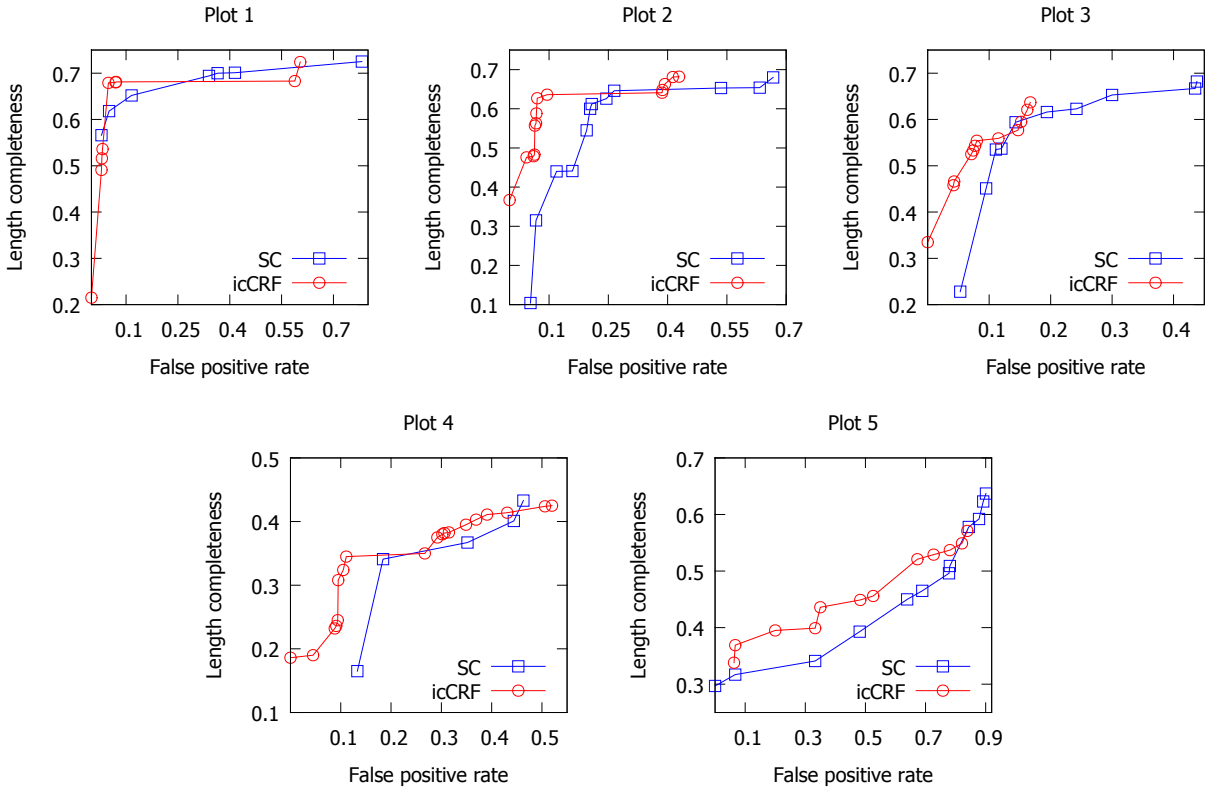
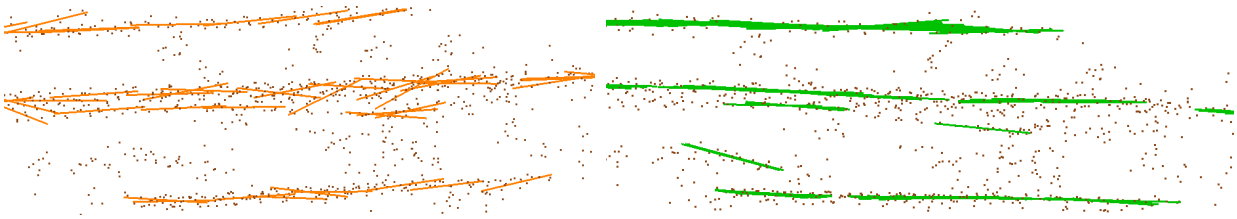


Figure 7.6: ROC curves showing performance of fallen tree detection algorithm for two versions of segment candidate selection: the proposed CRF-based approach (icCRF) and the baseline set cover-based method (SC). Top row displays curves for beech-dominated, bottom row for spruce-dominated plots.

Evaluation of fully automated processing

This section discusses the results of the experiment which mimics the real-life application of the fallen tree detection method to previously unseen data, where nothing is known about the optimal energy coefficients. Table 7.2 summarizes the detection quality metrics for the baseline set cover



(a) (b)

Figure 7.7: The same fallen tree scenario processed with two segment selection methods: (a) set cover-based and (b) CRF-based. (a) contains many overlapping segments with diverging directions due to the lack of collinearity and overlap modeling, (b) shows more ordered linear structures due to learned prior.

segment selection method (SC) as well as the contextual classification approach (icCRF). For Plot 1, when icCRF is used we observe an increase in length completeness of 4 pp, while maintaining the correctness (true positive rate). In case of Plot 2, both metrics benefit from icCRF, with correctness improving by 5 pp and completeness by 9 pp. Similarly, both quality measures can be increased by 5 pp on Plot 3. For the first of the coniferous test sites (Plot 4), the principal gain is in correctness (7 pp), accompanied by an enhancement in completeness of 2 pp. Plot 5 appears to be an anomaly in that the correctness obtained with icCRF significantly *decreases* (by 11 pp) with respect to the baseline.

Selection method	Plot 1	Plot 2	Plot 3	Plot 4	Plot 5
icCRF					
Correctness	0.9	0.97	0.96	0.9	0.47
Completeness	0.69	0.71	0.58	0.34	0.47
SC					
Correctness	0.9	0.92	0.91	0.83	0.58
Completeness	0.65	0.62	0.53	0.32	0.46

Table 7.2: Fallen tree detection quality metrics from ALS for the fully automated processing experiment. *SC* denotes the baseline set cover segment selection, whereas *icCRF* is the proposed CRF-based method.

The results associated with the first four plots show that utilizing icCRF as the segment selection method may lead to gains in detection correctness of up to 7 pp, and in completeness up to 9 pp. These improvements can be attributed to the segment selection step because the rest of the processing pipeline was identical in both parts of the experiment. In particular, both methods had access to the same initial pool of segment candidates, the selected segments were merged using identical parameters, and the resulting stems evaluated based on the same tree appearance models. As for Plot 5, additional investigations into the method's performance were conducted to determine the cause of the relatively poor correctness compared to the other test areas. It was determined that this performance gap should be primarily attributed to the learned appearance models used for ranking and pruning the segmented trees (Sec. 4.1.4). This is supported by additional results which show that assuming perfect network and tree classification models and a perfect scene ranking, the icCRF-based variation outperformed SC by 4 pp in correctness and 4.6

pp in completeness. Apparently, the dead tree classifier trained on data from the four remaining plots was too lenient in assigning the positive class to segmented objects, resulting in a high false positive rate. It was possible to nearly double the correctness for both SC and icCRF respectively to 0.88, 0.81 at no or little cost in completeness (no change: 0.46, drop from 0.47 to 0.42) simply by raising the tree classifier's positive label probability threshold above the automatically learned value. On the other hand, it is interesting to note that on 4 out of 5 plots, this imperfect ranking/classification scheme combined with partitioning the segment candidates into networks was able to outperform the ROC curves obtained when the entire plot was processed with one single parameter set (Fig. 7.6). This suggests that it is beneficial to split the segments into independent groups and search for each group's optimal parameters separately. Based on the results from all test plots, the new CRF-based segment classification method yields noticeably higher-quality segments for the merging step compared to the baseline context-independent set cover approach. This corroborates the initial hypothesis and intuition that introducing context in the classification process by explicit modeling of neighboring segments' spatial interactions is beneficial for the overall detection accuracy.

7.1.2 TLS point clouds

Evaluation of performance - comparison with SAC

The numerical results of the comparison between the introduced statistical cylinder detection framework and RANSAC methods are presented in Fig. 7.8(a). Each of the three test plots presents a different challenge. Plot A is dominated by longer stems which are scattered over a relatively large area of 1.8 ha. Here, two versions of the voting method with a stricter value $k_{kde} = 0.7$ of the KDE local maxima acceptance criterion dominated the ROC curve for the region of high accuracy (0.75-1.0). The uniformly weighted voting achieved a detected tree length completeness of 0.75 for an error rate of 0.16. Its closest competitor among the RANSAC methods was able to attain a completeness value of 0.66 at the same error rate, while the false positive rate (FPR) at a comparable completeness of 0.74 deteriorated to 0.36. The probability-weighted voting method was slightly inferior to the uniform version on the initial part of the ROC curve, however it was able to achieve the best completeness of 0.77 on the dataset, for an acceptable error rate of 0.24. It is noteworthy that for the aforementioned point (0.16, 0.75) on the ROC curve, the average length of trees entirely missed by the voting method was 1.6 m, compared to the mean length of 10.3 m for trees partially or fully detected. The version of the voting framework with $k_{kde} = 0.4$ scored mostly higher than the SAC procedures, except for the error rate of 0.27 where the completeness was worse. The difference in completeness compared to the best SAC variation deteriorated to 2.5 pp at 0.71 correctness. In contrast to Plot A, Plot C's structure leads to a different challenge: most reference stems are concentrated on a small area, clustered around a sequence of large fallen trunks. Plot C's ROC curve is dominated by the voting framework with the more lenient setting of $k_{kde} = 0.4$ for the entire correctness range of 0.7-1.0. Indeed, the uniformly weighted version of the voting approach is the only competitor to achieve a zero error result, with a corresponding completeness of 0.68. The closest runner-up from the SAC group attains an error rate of 0.08 and completeness 0.61. Interestingly, for this plot the introduction of probability weighting of votes had the greatest impact on the detection quality: a completeness of 0.78 was achieved for an FPR of 0.18, which constitutes an improvement of 9 pp compared to SAC, but also over 6 pp w.r.t. voting with uniform weights. In case of Plot B the comparison results are significantly different than in the other two plots. First, all methods show a relatively mediocre performance, never attaining a length completeness beyond 0.53 within the error rate range of 0-0.4. All methods are within 4 pp of each other, except one SAC variation which outperformed the best voting-based method by up to 7 pp.

Based on the results, it appears that detection of short stems (below 2 m) is particularly challenging. This confirms the intuition that the voting-based method works more reliably for longer stems, since the votes are based on a higher number of points. For Plot A, the fact that the stricter setting of $k_{kde} = 0.7$ outperformed the lenient one (0.4) could be associated with the aforementioned distribution of stems. In that plot, the fallen stems were overlapping only moderately, which in practice meant that each cluster obtained via connected component segmentation contained only one true stem. Therefore, the relatively high local maxima acceptance threshold of 0.7 was appropriate in most cases. Plot C has characteristics which lead to the inversion of this behavior. Because the stems were concentrated on a small area in close proximity, connected component segmentation often failed to isolate individual stems into separate components, even with the reduced neighbor distance threshold of 5 cm. This explains the inverted behavior of the algorithm w.r.t. the threshold k_{kde}, with the lenient setting dominant this time. Since many trees usually belong to a single connected component, considering only the strongest peaks in the voting space leads to missing a large portion of the trees which are less strongly represented in the point cloud. The necessity to manually select the KDE maxima cutoff threshold k_{kde} is a certain weakness of the proposed method. A possible solution is to replace the simple connected component segmentation step with a more geometrically aware clustering method, such as the graph-cut based approach proposed by Golovinskiy & Funkhouser [2009] for segmenting point clouds into foreground and background components. This method allows the incorporation of a tailored background prior into the segmentation process, which could make it possible to obtain more regular clusters. This in turn would make the k_{kde} less significant, since the underlying KDE would probably possess considerably less local maxima. Regarding the issue of weighting the parameter votes by their associated point probabilities, it seems that the weighting is mostly useful for non-homogeneous connected components which consist of parts of multiple stems, possibly contaminated with non-pertinent points (ground vegetation etc.). This is consistent with the intuition that the weights should be nearly uniform (and hence have no influence) if the considered point cluster already contains only relevant points without noise. As for the poor performance on Plot B, part of the reason may be attributed to the fact that Plot B contains, on average, the shortest trees, with a median length of only 1.2 m compared to the 3.2 m of Plot A. However, Plot C's median tree length of 1.5 m is only slightly higher, and the detection performance is vastly superior. Therefore, a more plausible explanation seems to be based on the difference in stem point densities. Plot B contains over 4 and nearly 2.5 times as many living trees per unit area as respectively Plots A and C, which leads to more occlusions during the point cloud acquisition process (Fig. 6.3b;Tab. 6.2). Consequently, in Plot B the median number of points per stem is nearly 5 times lower than in Plot A and over 14 times lower compared to Plot C, causing any approaches relying on geometric properties of the stem surfaces to break down. The introduction of weights into the parameter voting also did not make any significant difference for of Plot B. The SAC variation with no surface normal influence performed the poorest on all plots, therefore it was removed from the performance graphs in order to avoid clutter. The results from this section lead to the conclusion that when sufficiently high-quality geometric information about the object surfaces is available, the proposed voting framework can make better use of it than SAC methods (as exemplified by Plots A & C). On the other hand, when the number of points per stem drops below a certain value, there is no benefit to using the proposed method compared to SAC. This threshold value most probably lies between 1400 and 300 points per fallen stem (see Table 6.4).

Relationship to other methods

From a degree-of-automation perspective, the presented approach for detecting cylindrical structures occupies the middle ground between two existing paradigms. On the one hand, compared to SAC-based procedures this method has the advantage of not requiring iterative execution and the

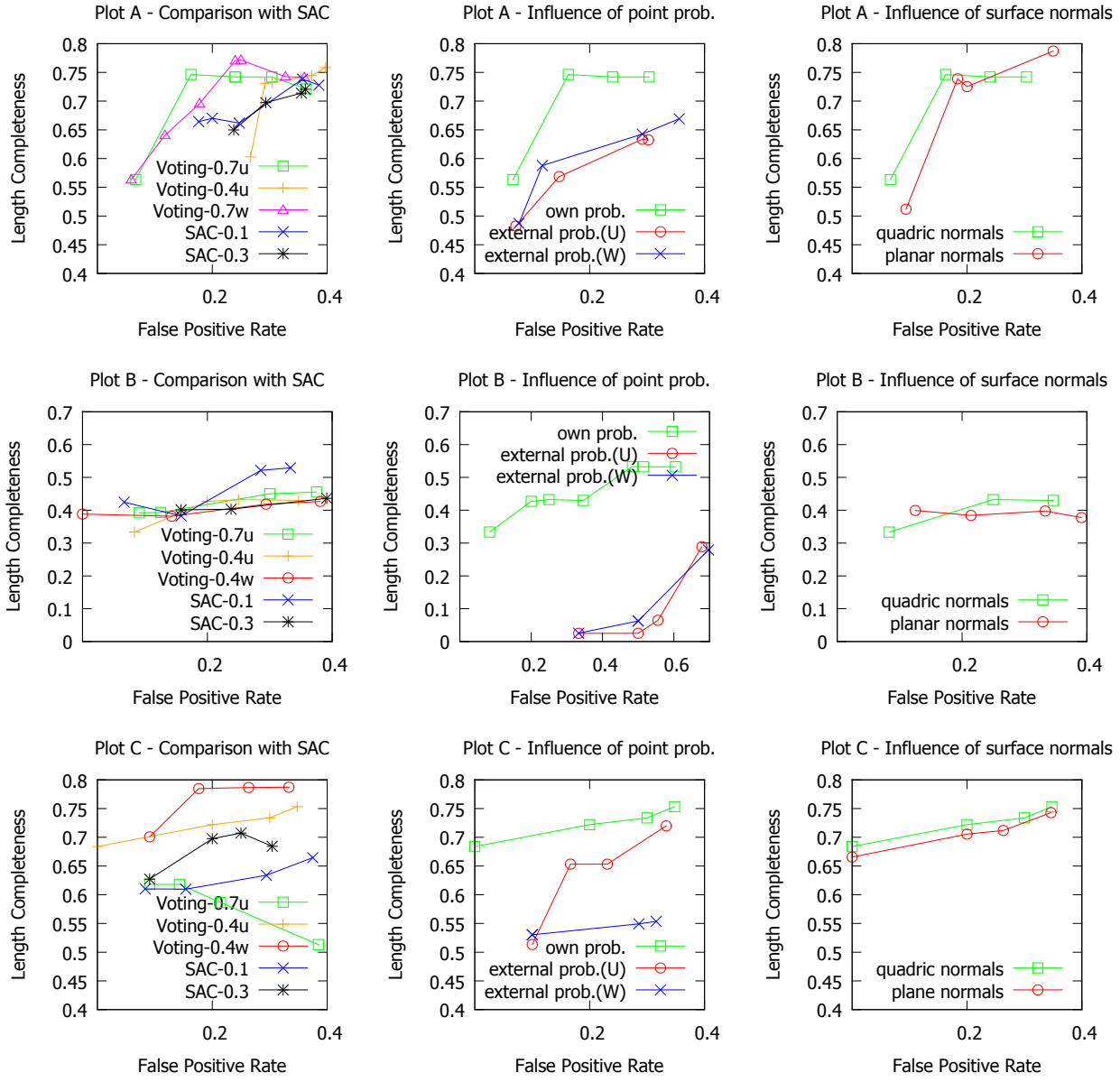


Figure 7.8: ROC curves showing performance of fallen stem detection pipeline from TLS. Rows 1-3 correspond to Plots A-C. Column (a) shows the comparison of the proposed statistical cylinder voting framework (at two levels of parameter $k_{kde} \in \{0.4, 0.7\}$, and two voting types: uniform and weighted) with RANSAC fitting (at two levels of surface normal influence $\{0.1, 0.3\}$). Column (b) demonstrates the influence of training the stem point classifier on external plots compared to using the target plot's own points. In column (c), the effect of utilizing two types of normal estimation methods (quadric fitting vs. local planar approximation) is presented. The suffixes 'u' and 'w' correspond respectively to uniform and weighted voting.

associated, potentially cumbersome elimination of inliers after each cylindrical shape has been recovered. Therefore, the proposed voting scheme produces deterministic results, which might not be true for iterative SAC. However, as a direct result of the parameter voting, a set of potentially overlapping cylinders is obtained which needs to be disambiguated. Also, the choice of the threshold k_{kde} controlling which cylinders will be retained can be somewhat difficult in practice,

for certain complicated scenarios. Such problems could, perhaps, be alleviated by casting the voting method in the framework of energy-based geometric model fitting due to Isack & Boykov [2012]. There, the authors proposed to formulate the shape fitting as a point labeling problem, with labels corresponding to randomly sampled shape candidates, in the spirit of SAC. The spatial support of the sampled models (or, equivalently, the assignment of labels to points) is estimated by minimizing an energy which combines a point-to-model distance criterion with a smoothness term. An additional regularization term in the energy penalizing nonempty model count ensures that the number of shapes does not need to be specified a priori. It is hypothesized that since the voting scheme described in this thesis produced higher-quality cylindrical primitives, the framework of Isack & Boykov [2012] could benefit from utilizing this method as a model generator instead of SAC. In such a setting, the k_{kde} threshold would not need to be explicitly set, since the model pruning would already be carried out as part of the energy minimization.

Evaluation of point classification transferability

The point level classifier's generalization capabilities and their influence on the performance of the entire processing pipeline are considered here. As it is visible in Fig. 7.8(b), using training data from external plots has a highly detrimental effect on the results for all test plots, regardless of whether the votes are weighted or not. It is only the magnitude of the deterioration that varies. In Plot A, for FPR 0.07 we observe a decrease of 8 pp in completeness, whereas at an FPR value of 0.16 the completeness gap reaches 18 pp. The versions based on external probabilities never exceed a completeness value of 0.67, compared to 0.75 obtained with the plot's internal probabilities. In case of Plot B, the externally trained classifier failed to produce usable stem point probabilities, leading to a complete breakdown of the method in the FPR interval of 0-0.4 and partially recovering only for poor FPR rates exceeding 0.67. For Plot C, vast differences in completeness nearing 20 pp are observed in the low FPR ranges up to 0.2, with the gap gradually closing to reach 3 pp near FPR 0.33 when considering the uniformly weighted voting scheme. For the probability weighted variation, the completeness did not recover, never breaking the 0.55 level. This shows that applying weights to the parameter votes can also decrease the performance if the point-level probabilities are of poor quality.

All of these results suggest that the point classification step has critical importance for successful detection of the fallen stems, and on the other hand that the utilized features did not generalize well enough across the studied test plots. It should be noted that part of these discrepancies is probably associated with the significantly different characteristics of the plots, especially regarding the average point counts per stem (see Tab. 6.4).

Evaluation of surface normal influence

The effect of choosing the surface normal estimation method is depicted by the curves in Fig. 7.8(c). On Plot A, the quadric normal based version is superior to its plane based counterpart for high correctness values (1-0.8), achieving an advantage of up to 5 pp completeness at a false positive rate of 0.07. However, it is the planar estimation variant which attains the highest overall completeness of 0.79 for FPR 0.35, which in turn constitutes an improvement of 4 pp over the quadric version's best result of 0.75. For Plot B, the quadric approach gains an advantage of up to 4 pp between FPR 0.2 and 0.4, while the opposite situation takes place at an FPR of 0.08. Plot C exhibits the most uniform behavior: the quadric version holds a steady lead of ca. 2 pp over the entire FPR range.

Overall, the results of the surface normal comparison are inconclusive in that they do not reveal a clear winner. Therefore, it seems that the intuition of quadric fitting based surface normals leading to higher quality detection results is not true. Moreover, the quadric surface

estimation requires significantly more computational effort: the eigenvalues of a 6x6 matrix must be determined for each point. Considering that the eigenvalue calculation has a computational complexity of $O(n^3)$ and that for plane fitting only a 3x3 matrix has to be processed, the quadric version requires 8 times the effort.

7.2 Standing dead trees

7.2.1 Aerial imagery - active contours

Evaluation of dead tree crown delineation

Table 7.3 summarizes the performance of the dead tree delineation method on all test plots. The reference and detected contours are outlined in Fig. 7.9. In Plot I, most trees are clustered in the central area and the main challenge consists of separating the individual snags. For this plot, all reference trees can be matched to detected snags, with a false positive rate of 0.25. The algorithm does a reasonable job of delineating the individual crowns, attaining an average Dice coefficient value of 0.57 and center-of-gravity displacement of 5.5 pixels. The significant difference between the correctness in the 1-1 and N-M matching strategies reveals an over-segmentation. Plot II contains a similar number of snags as Plot I, however they are distributed across a larger area. This plot tests the ability to detect isolated snags surrounded by living vegetation or standing in shadowed areas. The algorithm is able to delineate the individual snags quite reliably, achieving a completeness of 0.96 with a correctness of 0.71. Compared to Plot I, the mean Dice coefficient is 12 percentage points higher at 0.69 and the mean center-of-gravity displacement more than a pixel lower at 4.4 px. Plot III, the final test area, is quite challenging due to the high variability of the snag crown diameters (Fig. 6.4c), presence of shadowed areas, open ground patches and fallen dead trees. Nevertheless, the algorithm is able to attain a completeness value of 0.81 with a correctness of 0.77. A drop in pixel-level completeness to 0.5 (see Table 7.3) can be observed in comparison to the other plots.

Metric	Plot 1	Plot 2	Plot3
Dice Coefficient	0.57	0.69	0.62
Center-of-gravity distance [px]	5.5	4.4	3.5
Per-pixel correctness	0.59	0.76	0.89
Per-pixel completeness	0.61	0.67	0.50
Per-pixel correctness (N-M)	0.76	0.58	0.78
Per-pixel completeness (N-M)	0.85	0.62	0.52
Object correctness (1-1)	0.75	0.71	0.77
Object completeness (1-1)	1	0.96	0.81
Object correctness (N-M)	0.96	0.77	0.85
Object completeness (N-M)	1	1	0.82

Table 7.3: Evaluation of dead tree crown detection results. Rows 1-4: average polygon matching quality metrics for 1-1 matching. Rows 5-6: average per-pixel correctness and completeness for N-M matching. Rows 7-10: object-level correctness and completeness for 1-1 and N-M matching.

For Plot I, the false positive polygons mostly lie in dark areas at the perimeter of larger trees. However, note that the dead tree pixel probability within these polygons is relatively high

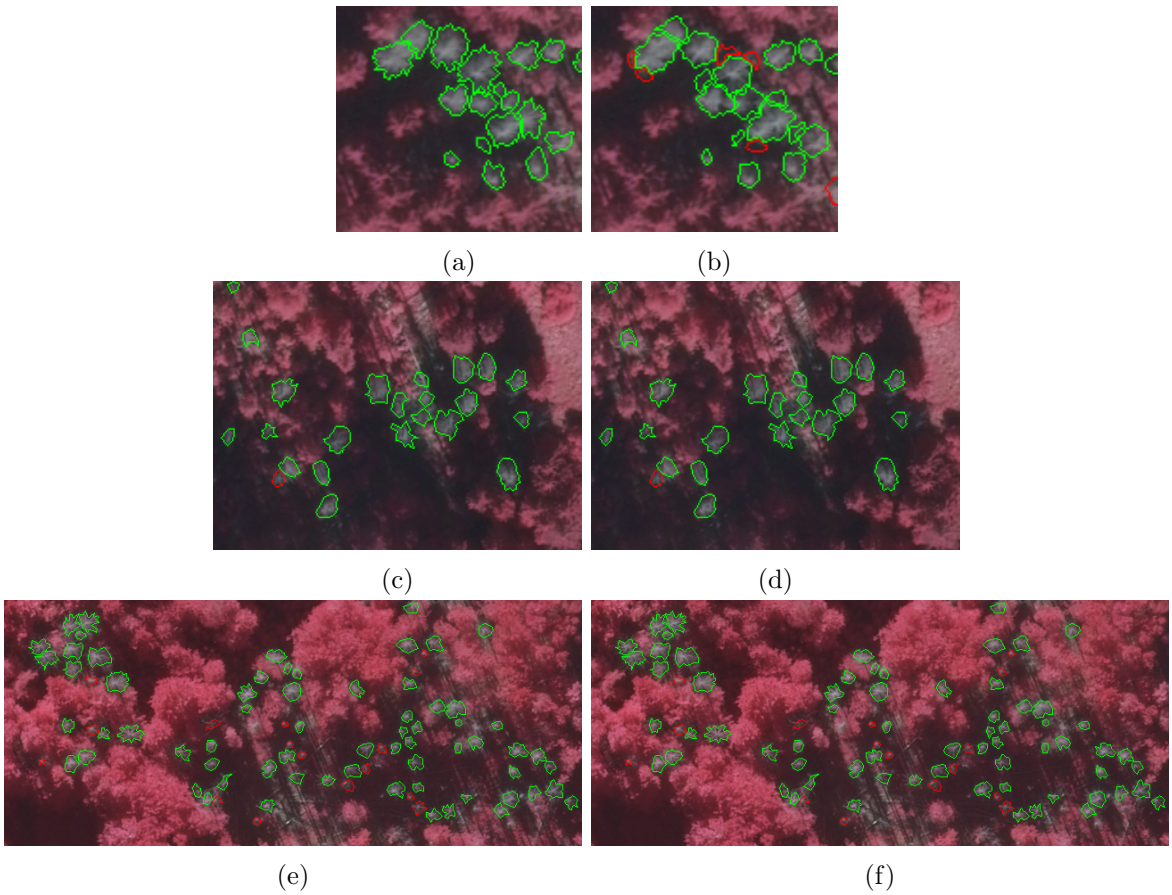
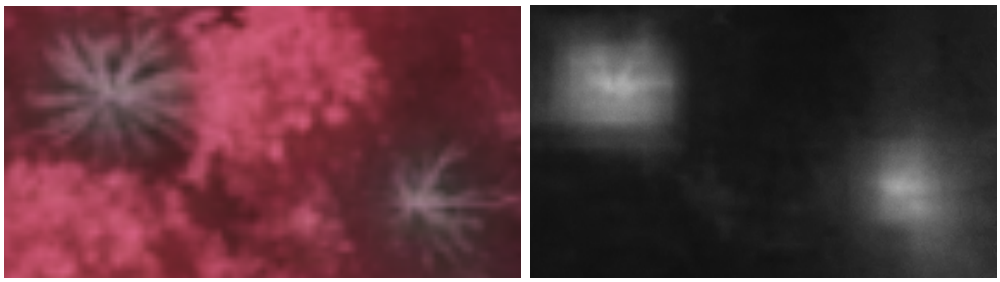


Figure 7.9: Reference and detected polygons. (a), (c), (e): reference polygons respectively of Plot 1,2, and 3. (b), (d), (f): detected polygons respectively in Plot 1,2, and 3. Red contours denote unmatched, green denote matched polygons using the 1-1 matching strategy. Note the different image scales - (a),(b): 26 m x 24 m; (c),(d): 58 m x 42 m; (e),(f): 130 m x 55 m.

(cf. Fig. 5.4c). The segmentation energy functional only interacts with the pixel intensities through the prior distribution function, and therefore any inaccuracies in the discriminative intensity model will be directly transferred to the segmentation result. As stated in Sec. 6.3.1, the model's pixel classification accuracy was 0.81, which indicates that the snag and background pixels cannot be clearly separated using the supplied 3 spectral channels, perhaps due to diverse lighting conditions and varying hues of the target objects. Since any generative or discriminative model may be used in the role of the intensity prior (Sec. 5.1.2), it is hypothesized that this part of the method could be improved upon by exploring a richer feature space, e.g. augmented by further spectral bands or by transformations thereof such as the NDVI, or by applying more powerful learning methods such as probabilistic graphical models. Part of the false positives detected in Plot II are associated with small open ground patches visible in the image that additionally contain fallen trees. Once again this deficiency may be attributed to the intensity classifier, because neither ground or fallen tree pixels were included in the training set, and their colors in the image are closer to the snag class than to the remaining background classes. These problems outline a disadvantage of this purely image-based approach versus methods which have access to a surface model or a 3D point cloud, where such open ground patches could be easily detected using simple height thresholding schemes. As expected, the quality of the detected polygons is higher in this plot compared to Plot I, because finding the boundary of single crowns is an easier and less ambiguous task than splitting a cluster of several crowns. The inferior performance on Plot III can be partially explained by

the presence of snags with elaborate branch structures visible in the image. The pixel classifier does not take any neighborhood properties of a pixel into account, and therefore any fine details are usually lost when calculating the snag probabilities (Fig. 7.10). This makes it impossible for the method to recover the fine branch structure, and instead only the core area of the crown is delineated. Since it is unlikely that a perfect point-level snag classifier can be obtained, it seems more promising to alter the segmentation energy functional to include a term which evolves the intensity model together with the shape using contrast information present in the original image (such as the classical Chan-Vese model, Chan & Vese [2001]) while maintaining a balance with the intensity prior. This is expected to come at an additional computational cost.



(a) (b)

Figure 7.10: Loss of tree crown contour detail due to transformation into probability image. (a) branch structure visible in original image, (b) probability image (intensity proportional to dead tree probability), most of fine branches blurred or not visible.

7.2.2 ALS point cloud - Free Shape Contexts

Evaluation of performance - Free Shape Contexts

Table 7.4 summarizes the overall classification results for all 6 tested combinations of feature families and optimization algorithms. The computational effort is measured in units equal to the number of evaluations of individuals in a single GA run, i.e. 150000. In case of the two SC-based feature types, the genetic algorithm is able to outperform the grid search/dynamic algorithm by up to 4.6 percentage points (pp), using less parameter configuration evaluations. Also, in case of the Free Shape Contexts, the dynamic algorithm is not able to find a solution comparable to one obtained through GA despite a tenfold increase in the used computational resources. Interestingly, the GA fails as an optimizer of CE parameters, its best found solution lagging by 2.8 pp behind the grid search result. The proposed Free Shape Context-based features delivered the best performance. For average quality of the GA-optimized solutions, FSC outperforms SC by 4 pp and CE by 7.1 pp, whereas in case of the best found solution, the respective gains are 3.5 pp and 4.2 pp. Regarding the feature counts associated with each descriptor, the best FSC feature set had a cardinality of 10, equalling the corresponding count for CE and lying significantly below the SC-related count of 30. To investigate the statistical significance of the performance differences, the one-tailed binomial test was used. The outcome of the labeling of N objects by a classifier with accuracy p can be regarded as a random variable with a binomial distribution $B(N, p)$. In particular, for the pair FSC-CE, it is tested if the result of 240 successfully classified objects is likely to have arisen from a distribution $B(285, 0.8)$. The obtained p-value is 0.04, so the difference is significant at a level of 0.05. Similarly, for the pair FSC-SC, the p-value lies at 0.07. In contrast, the difference between SC and CE is considerably less significant with a p-value of 0.42.

Descriptor	Effort	Overall accuracy [%]		
		max.	μ	σ
Shape Context				
Grid Search	1.07	76.1	76.1	0.0
GA	1	80.7	78.7	1.3
Covariance Eigenvalue BoF				
Grid Search	1.09	80.0	80.0	0.0
GA	1	77.2	75.6	1.0
Free Shape Context				
Dyn. Search	10.44	80.0	80.0	0.0
GA	1	84.2	82.7	0.8

Table 7.4: Evaluation results for standing dead stem detection using the considered features w.r.t. classification accuracy. Workload reflects the number of evaluated parameter sets (1 = number of evaluations in a single GA run).

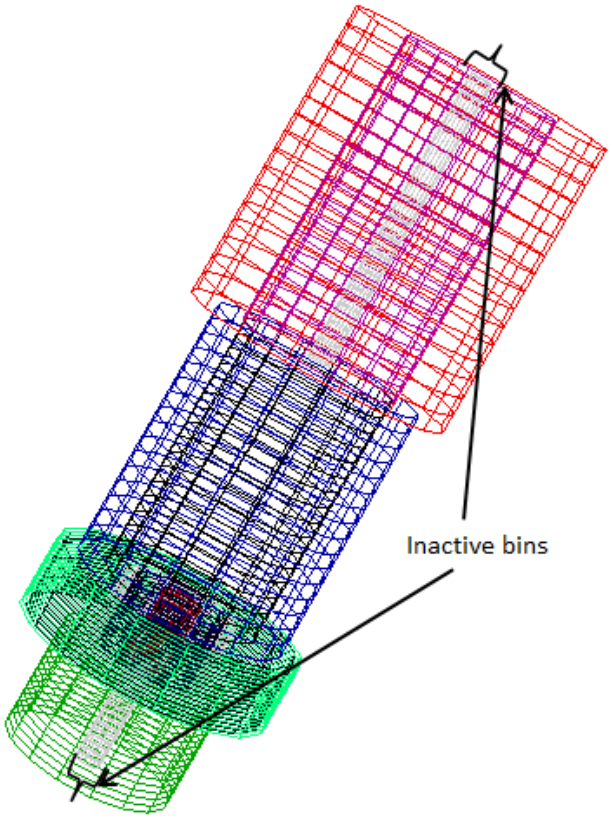


Figure 7.11: Optimal evolved Free Shape Context for standing dead stem detection. The inactive internal bins are shown in gray.

Due to the discussed high computational complexity of optimizing the FSC parameters (Sec. 5.3.2), it was expected that the approximate dynamic algorithm for optimizing FSC would have trouble competing against the GA, despite more allocated resources. This was confirmed by the results. For the sake of fairness, it should be noted that because they are stochastic in nature, genetic algorithms do not always converge to the best found solution. In fact, it was observed that for all three feature types, out of 15 runs with random initializations, the best solution was only obtained 1-2 times. This emphasizes the difficulty in finding an optimal set of GA meta-parameters which reliably leads to convergence under different initializations. On the other hand, even the mean classification accuracies averaged over all restarts outperform their counterparts from grid search for SC and FSC. The GA's failure to optimize the CE parameters is likely caused by these parameters consisting mostly of integers (numbers of feature divisions) as opposed to the SC and FSC parameters being mostly reals (cylinder lengths and radii). This makes it harder for the evolutionary process to find a gradient of the optimized function w.r.t. the more discretized parameter space. Perhaps a genetic representation or recombination method more suitable for integer variables could remedy this problem. In terms of the best-performing solution, the CE features appear to be inferior to both SC and FSC. This indicates that the point density afforded by ALS point clouds is not sufficient so as to support discriminating between different surface types on a scale finer than several meters. This claim is further supported by the fact that the optimal CE performance arose from a neighborhood radius of 4.4 m for the covariance matrix calculation (from an available interval of 0.1-5 m), which implies that the no meaningful way to utilize information based on smaller neighborhoods could be found. As for FSC, considering its superior performance in conjunction with the smaller or equal cardinality of the generated features suggests that, compared to its competitors, the FSC descriptor was able achieve a better balance between finding the relevant, discriminative parts of the target objects and omitting non-informative parts. The optimal evolved Free Shape Context is shown in Fig. 7.11. Apparently, less attention was given to the layer closest to the ground, whereas certain parts of the tree higher along the axis were captured in a more detailed and comprehensive manner. This mirrors the fact that the lower portion of the point cloud delivers less information about whether the object is a dead stem, since the presence of ground vegetation is independent of the tree's health, and on the other hand both living and dead trees consist of a stem up to a certain above-ground height.

7.2.3 Aerial imagery & ALS - active/semi-supervised learning

Evaluation of effects of pre-selection

First, the effects of greedily pre-selecting elements to form a reduced dataset as a pre-processing step for active learning are discussed. The results of this experiment are visualized by Fig. 7.12. For each problem, two pairs of active learning performance curves are displayed, corresponding to four configurations of expected error reduction (EER) and random sampling (RS) with random and greedy pre-selection. Interestingly, the results diverge strongly between the problems. For Problem 1 (see Sec. 6.3.3), EER attained superior results when the randomly sampled data pool was used (R+EER), with an average advantage of 1.7 percentage points (pp) over the scenario when greedy pre-selection was used (G+EER). In contrast, when RS is used as the active learning method, it can make better use of the greedily sampled pool. Although for small training sets (up to 25 instances) the random/random combination (R+RS) outperforms RS with greedy pre-selection (G+RS), for the remaining training set sizes the latter obtains an advantage averaging 1.5 pp. Also of note is the fact that for Problem 1's greedy datasets, the class percentage has been shifted nearly threefold from the original 7.2% to ca. 22% (see Tab. 6.5), biasing the classifier towards more frequent occurrences of the target class. As for Problem 2, the situation is more clear since both learning methods can strongly benefit from the greedy dataset. Here, EER was

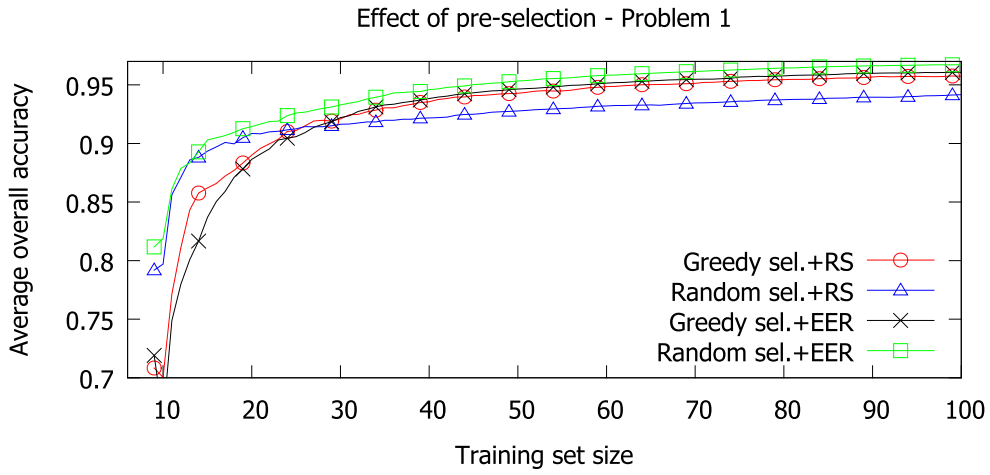
able to better capitalize on the increased data information content, with an average gain w.r.t. the randomly pre-selected pool at 11.8 pp compared to the 7.2 pp of RS. Also, for this problem the class balance of the greedily selected datasets was not shifted as severely, with a change from 16.1% to 19.9%.

The fact that RS was able to better take advantage of the greedily selected data compared to EER on Problem 1 may be attributed to the inherent difference between these two learning methods which concerns independence of selecting a new sample from past choices. For EER, obviously there is a strong dependence between all examples that have been chosen so far, and the next object which will be chosen, because the choice is based on probabilities obtained from a model trained on these examples. Combined with the dramatic shift of the class distribution in the greedily selected samples in favor of the target class, this leads to reinforcing the choice of positive examples in the learning process, resulting in their constant over-representation in the labeled set for EER. In case of RS, the situation is quite different, because the choice of the next training example is by construction independent of any prior choices. Therefore, apparently the training set does not become as biased and RS can make use of more informative examples present in the greedily selected data pools. Regarding Problem 2, the choice of the training pool had a stronger influence on the performance than the choice of the learning method. The qualitative difference between Problem 1 and Problem 2 may perhaps be explained by the fact that (i) for Problem 2, the greedy datasets were not nearly as biased (from 16.1% to 19.9%) and (ii) in Problem 2, the shape descriptor probably produced more diverse instances, since the 3D appearance of the objects within the point cloud is more varied than the average colors of vegetation within a CIR image, therefore the set of informative objects did not saturate as quickly.

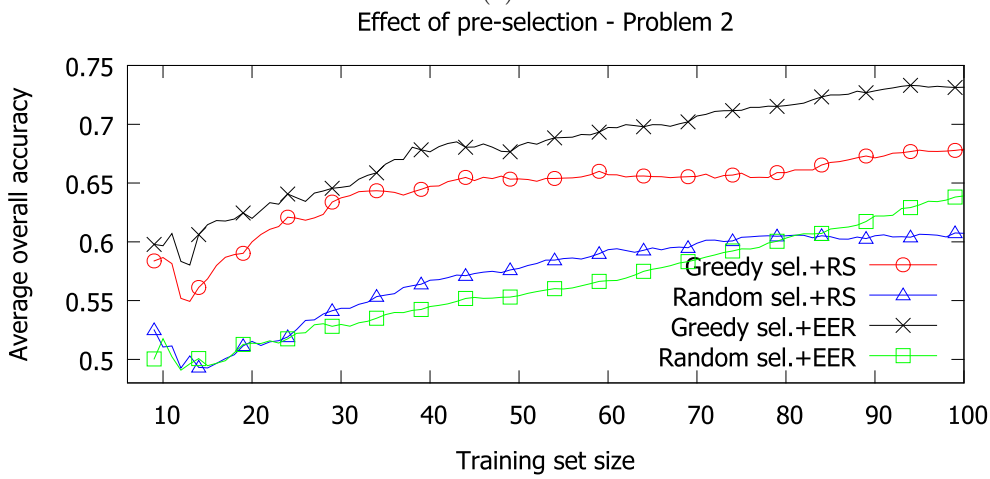
Evaluation of semi-supervised learning

This section describes the influence of entropy regularization on semi-supervised learning performance, including sensitivity to choice of the Renyi parameter α. Figures 7.13a and 7.13b depict the classification rate curve for Problem 1 and Problem 2, respectively. For clarity, curves for only 2 Renyi α values per problem are shown - the ones resulting in the largest average gain over the baseline for $\alpha < 1$ and $\alpha > 1$. The other curves not displayed with $\alpha < 1$ were usually located between the Shannon entropy curve and the optimal Renyi entropy curve. As expected (Eq. 3.37), for α very close to unity (from both sides), results were very similar as for Shannon entropy, whereas $\alpha \gg 1$ generally gave rise to inferior performance. The optimal performance for Problem 1 and 2 was attained respectively at $\alpha_1 = 0.5$ and $\alpha_2 = 0.3$. For Problem 1, this gain is quite small, with the average difference between α_1-Renyi and Shannon regularization at 0.6 pp and the maximum difference attaining 1.1 pp. This gain is more prominent in case of Problem 2, where an average and maximum difference of 1.8 and 2.9 pp in favor of α_2-Renyi entropy can be observed. Both problems exhibit a substantial gain from regularization compared to the unregularized baseline. Particularly for small training sets (below 10 elements), an increase in accuracy of up to 9.8 pp and 14.6 pp for Problem 1 and 2, respectively, is observed. As the system obtains more labeled objects, the significance of the entropy term gradually vanishes. Nevertheless, the regularized model is able to outperform the baseline by at least 1 pp for training sets of up to 24 elements (Problem 1), and at least 2.3 pp for all investigated training set sizes (Problem 2).

The presented results confirm the utility of entropy regularization as reported before in [Grandvalet & Bengio, 2006]. Moreover, they indicate that Shannon entropy is not necessarily the best choice for regularization, and in particular entropies which distribute the weight more uniformly, i.e. which are less peaked ($\alpha < 1$), can yield superior results. Also, it seems that the optimal α value is application-specific. As expected, the influence of the regularization is mostly significant



(a)



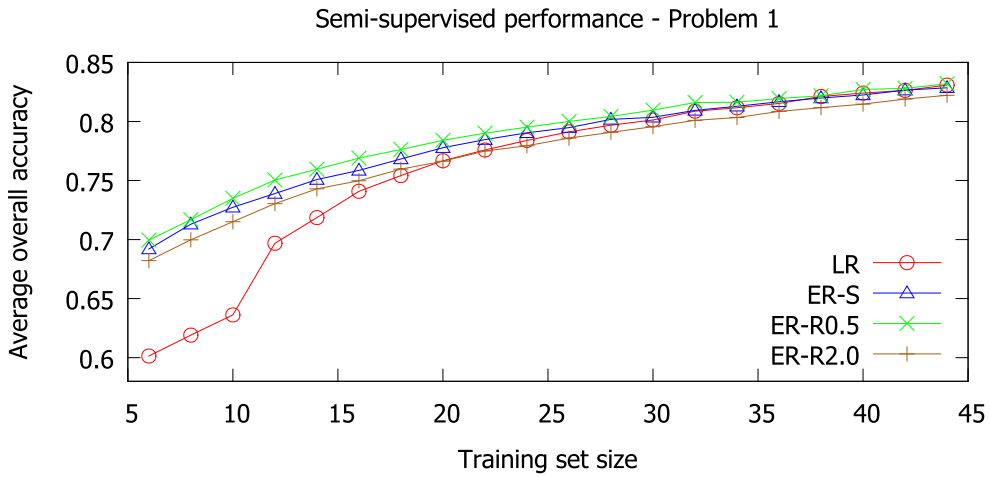
(b)

Figure 7.12: Effects of object pre-selection on performance of active learning via Expected Error Reduction (EER) and Random Sampling (RS) for dead tree detection. For each problem and each active learning strategy, effects of greedy and random pre-selection are shown.

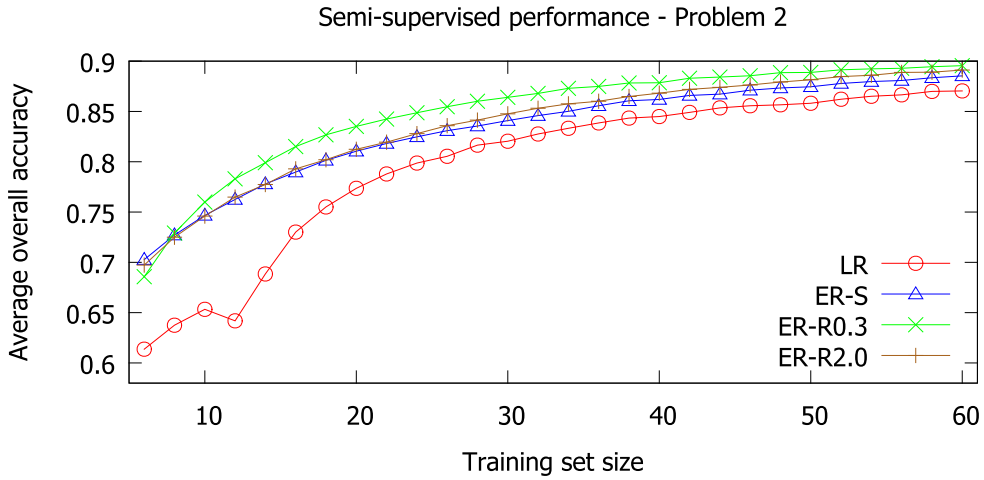
for small datasets, since once a sufficient number of labeled training examples is available, the information content in the unlabeled pool gradually decreases.

Evaluation of combined active and semi-supervised learning

Finally, the matter of active learning performance, depicted in Fig. 7.14, is addressed. In this experiment, the Renyi entropy α for each problem were taken as the values which led to the best semi-supervised performance (see section above). Not surprisingly, random sampling was the poorest-performing method, trailing on average by 2.5 pp (Problem 1) and 5.5 pp (Problem 2) on the entire interval of 9-100 considered training examples. The proposed method of Expected Error Reduction with Entropy Regularization (EER-ER) outperforms standard EER for training set sizes up to 60 (Problem 1) and 88 (Problem 2). This advantage is strongest in the early stages of learning. For Problem 1, the average gain after the first query to the oracle (9 training elements) is 10 pp, gradually diminishing to 5.2 pp at 15 and 1.2 pp at 30 training elements. As a result, the EER-ER method attains the overall accuracy level of 0.8 (or 90.5% of the final classifier accuracy) using 14 queries to the oracle, an improvement of 30% compared to 20 queries for standard EER. Regularizing with Renyi entropy yields a benefit over Shannon entropy similar



(a)



(b)

Figure 7.13: Semi-supervised learning performance for dead tree detection. Axes correspond to training set size and overall accuracy on unlabeled examples. LR denotes baseline logistic regression, while ER refers to regularization with (S)hannon and (R)enyi entropy.

to the semi-supervised case, averaging 1 pp for 19-40 training samples. Regarding Problem 2, the improvement associated with the EER-ER method is stronger, as was also the case for the previous experiment. The gain in overall accuracy peaks at 11.4 pp for 12 labeled examples, continues to exceed 5 pp for up to 23 instances, diminishing to 2.5 pp at 40 and finally falling below 1 pp at 67 instances. This time, it takes 50% of the EER's 30 oracle queries for EER-ER to break the 0.8 classification accuracy mark. As for the difference between Renyi and Shannon entropy, it is also more clearly visible. At 10 labeled objects, Renyi entropy is superior by 3.8 pp. An advantage of over 2 pp is maintained for up to 27 examples, slowly declining towards 1 pp at 47. It is interesting to note that on the range of 8-20 training samples, the average gain is 3.3 pp, compared to 2.2 pp for the semi-supervised case. For larger training set sizes approaching 100, EER-ER is slightly inferior to standard EER, with maximum differences of -0.7 pp (Problem 1) and -0.3 pp (Problem 2).

Apparently, the benefits of regularizing the classification model also apply to active learning, since a clear advantage in the early stages of learning (i.e. for small training sets) compared to standard EER is observed for both problems. In this scenario, utilizing Renyi entropy as the regularizer also proved beneficial compared to standard Shannon entropy. The aforementioned

fact that the gain within the active learning setting was slightly higher compared to the semi-supervised case could indicate that the improved active learning performance is not only due to the regularized model being a better classifier, but in part also because the Expected Error Reduction framework is benefiting from the better probability estimates in the example selection process (Eq. 3.32), particularly for small training sets where these estimates may be uncertain.

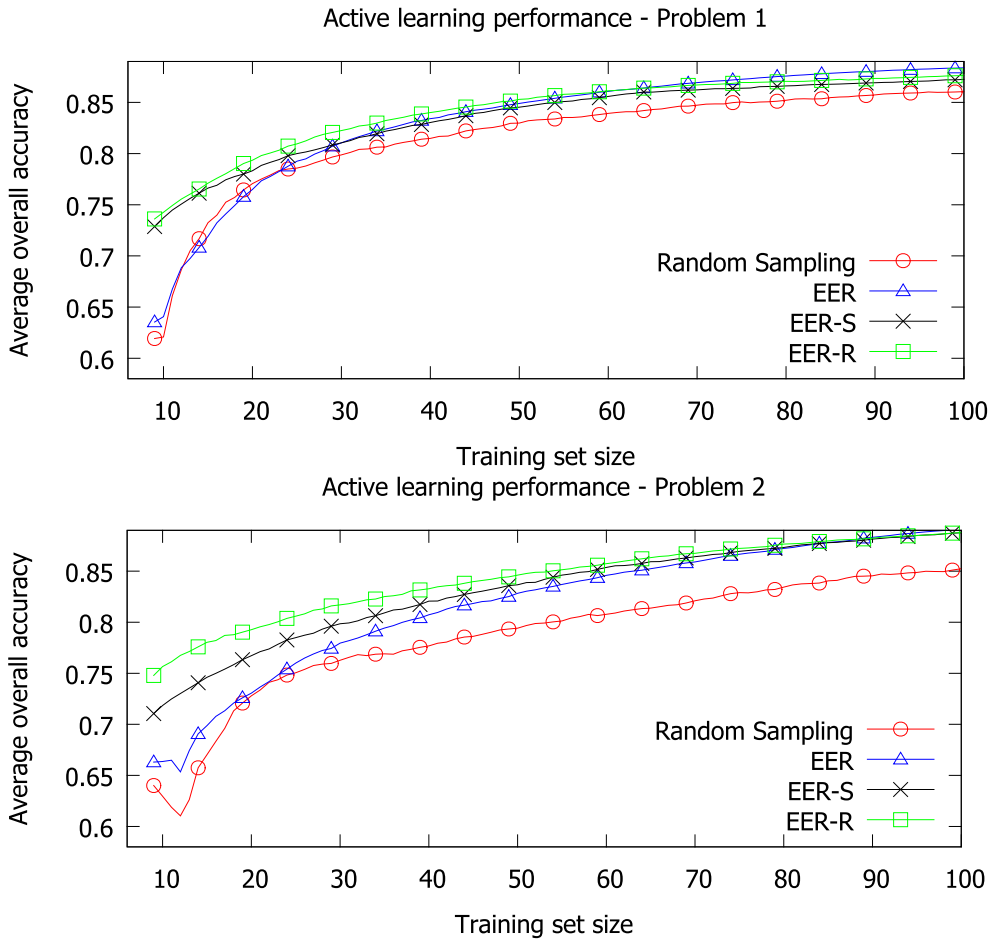


Figure 7.14: Active learning performance for dead tree detection. Axes correspond to training set size and overall accuracy on test set. Expected Error Reduction (EER) is compared to the new (S)hannon and (R)enyi regularized versions as well as random sampling.

8 Conclusions & Outlook

8.1 Conclusions

The purpose of this final chapter is twofold - to state the most important conclusions resulting from the work carried out as the basis for this thesis, as well as to propose new directions for further research based on the limitations associated with the presented methods. The conclusions are grouped according to their relationships to specific goals pursued within the thesis, as well as to the four general research questions posed in Section 1.3.

First, the task of detecting and reconstructing fallen trees is addressed. Here, one of the top-level goals was to propose a framework for the reconstruction of elongated objects within sparse point clouds. This was achieved by introducing a generic bottom-up approach which gradually accumulates information, starting from the level of single points, through point groups, or primitives, serving as building blocks for merging into individual objects. The framework was instantiated for the particular case of fallen tree detection from ALS point clouds, fulfilling the next goal. It was found that under a low to moderate overstory cover, the method was able to recover dead stems reliably (with an accuracy exceeding 80%). Answering part of Question I, the results suggest that the proposed methodology concerning ALS has reached a maturity level permitting large-scale deployment in forests. Regarding Question II, one of the critical factors for the method's success is the degree of overstory cover. Indeed, the performance of the method declines markedly when the overstory cover exceeds ca. 40%. Aside from overstory density, the number of trees per unit area as well as the average fallen stem diameter are parameters which influence the difficulty of the detection task. For a practical application, a point density of at least 20 points/m^2 is recommended, below which the stems are poorly represented in the point cloud. Another important factor in ensuring successful recognition is a sufficiently fine DTM resolution. The results show that a DTM grid cell size of 50 cm is too coarse, and significant loss of quality may occur compared to a 10 cm grid. Moreover, by explicitly modeling the spatial relationships of adjacent primitives in a contextual classification scheme based on a CRF, a considerable gain in final detection accuracy was attained, compared to the case when the primitives were classified independently. It was shown that automatically learning the constituent models of the CRF from labeled training data is possible. Also, the results of the effects of learning the components of the Normalized Cut algorithm used for merging primitives into objects indicate that the learned similarity function and stopping criterion could both generalize quite well across all test plots. This makes it a viable option to perform the learning only once and reuse the result for new input data, instead of carrying out a very computationally expensive grid search on the parameter space.

The next objective of this thesis was to develop a computationally efficient parameter accumulation - based method for reconstruction of cylinders from dense 3D point clouds, which would relieve the user of the necessity to manually determine the accumulation cell size. The proposed method uses kernel density estimators to perform the parameter voting, which has the advantage of automatically determining the kernel bandwidth, and hence parameter cell size,

using established, statistically well-founded algorithms. This voting method was applied to the reconstruction of fallen stems from TLS data. Answering the second part of Question I, under appropriate conditions the TLS methodology for mapping fallen trees can also be operationally deployed in real-world forest plots, reaching completeness values exceeding 75% for error rates below 20%. Revisiting Question II, these appropriate conditions are mostly related to a sufficiently low stem density per unit area, preferably below 300 trees/ha. Furthermore, the results revealed an increased quality of the recovered cylinder parameters, compared to several versions of sample consensus (SAC) methods, mostly for the case when the point density was high enough. This threshold density value most probably lies between 1400 and 300 points per fallen stem. Perhaps contrary to intuition, it was found that the use of surface normals estimated by quadric surface fitting did not yield a significant benefit despite the fact that cylinders are themselves quadric surfaces. In this case, utilizing the standard local plane fitting method for normal estimation proved sufficient. It was also determined that the applied 3D shape descriptors for detecting cylindrical surfaces within the point clouds did not generalize well across datasets with significantly different point densities, therefore care must be taken to ensure that the properties of the training and test plots are possibly similar.

In case of standing dead trees, three principal goals were pursued. Concerning the detection of dead stems without crowns from ALS data, a specialized 3D shape descriptor, the Free Shape Context, was introduced. It was demonstrated that features derived from the FSC afforded a moderate benefit in dead stem detection accuracy, compared to the both the original cylindrical 3D shape contexts and a simple bag-of-features model based on the structure tensor of local point neighborhoods. Due to the heterogeneous structure of the FSC and the ability to introduce buffer elements which do not produce features, the dimensionality of the generated feature space can be significantly lower than in case of standard homogeneous SC, resulting in computational advantages. The FSC descriptor was thus able to achieve a better balance between finding the relevant, discriminative parts of the target objects and omitting non-informative parts. Furthermore, genetic algorithms were found to be an appropriate tool for optimizing the FSC structure for a specific classification task. Referencing Question III, the dead stem detection accuracy attained on the test plots exceeded 84%.

The next considered scenario was based solely on aerial CIR images. The proposed level-set based dead crown segmentation method made use of shape and intensity priors which were automatically learned from tree outline polygons marked within training images. It was confirmed that the NIR spectral band can provide a good basis for deriving features capable of reliably distinguishing dead trees. Based on the introduced shape prior, it was possible to separate adjacent dead tree crowns into single trees. By combining both the shape and intensity prior information and executing the level-set segmentation at various high-probability dead vegetation areas in the image, an implicit, simultaneous segmentation/classification scheme was defined, where the object outline may evolve to an empty contour if not enough information within the image supports an object occurrence. Such a procedure has the advantage of already including target class-specific information into the segmentation, as opposed to first conducting a generic segmentation and only afterwards introducing target class characteristics into the classification step. A certain drawback is the computational cost of the level-set segmentation, which must be carried out many times within the image. Also, the lack of 3D information at times resulted in confusion between the pixel colors of dead tree crowns and patches of open ground due to their similar appearance in the CIR image, leading in some cases to lower detection correctness of ca. 70%. This confirms the importance of 3D information for successful dead tree crown recognition, partially answering Question III.

In the third setting, both aerial imagery and ALS data were available. The utilized method combined both data sources by segmenting the point cloud into single trees and then projecting the points of each tree cluster onto the image plane, to obtain 2D bounding polygons. Compared to the previous CIR image-only case, the advantage was the presence of 3D height information, which made it possible to filter out not only ground areas but also non-pertinent low vegetation. On the other hand, the tree crown polygons obtained from the 3D segmentation were less accurate compared to the level set segmentation above, because the ALS point cloud segmentation operated on coarse voxels instead of individual laser points. Moreover, it was found that a certain shift between the ALS and image data may occur due to perspective effects of the camera, resulting in a number of irrelevant pixels entering the tree polygons. Nevertheless, the classification results show that this was not a major issue, possibly due to the fact that the applied features were averaged over all pixels inside the bounding polygon, of which the majority was assigned correctly. Concerning Question III, the experiment showed that an overall accuracy of up to 88% is attainable in this scenario.

The final set of objectives was related to combining semisupervised and active learning into a unified framework based on Renyi entropy regularization. From a practical standpoint, particularly in remote sensing applications there is usually a vast amount of unlabeled data available, and it is not feasible to apply some of the more computationally demanding active learning methods directly to the entire dataset. One of the insights gained as part of this thesis is that a careful pre-selection of examples which take part in active learning can lead to significant improvements compared to random selection. Specifically, a greedy, low-rank matrix approximation method due to Smola & Schölkopf [2000] was used for this purpose. Regarding semi-supervised learning, the logistic classifier regularized with Shannon entropy, due to Grandvalet & Bengio [2006], was extended to Renyi entropies. It was found that particularly the entropies which distribute the weights more uniformly than standard Shannon entropy, i.e. with parameter $\alpha < 1$, were better suited for regularization compared to their more peaked counterparts ($\alpha > 1$). It seems that the optimal α value is application-dependent. Finally, addressing Question IV, the experimental results confirmed that it is indeed beneficial to combine the entropy-regularized classifier with the expected error reduction (EER) active learning algorithm. Particularly for small training set sizes, a significant increase (up to 10%) in classification accuracy was observed in comparison to standard EER. Also, the number of oracle queries needed to attain over 90% of the classifier's best accuracy (when trained on all available data) was reduced by up to 50%.

8.2 Outlook

The encouraging results obtained for reconstructing fallen stems in ALS point clouds suggest an application of the general linear structure detection framework to both new object categories and new data types. For example, the related problem of detecting fallen trees using only 2D raster imagery could be treated in this manner. An application to other types of elongated objects, such as industrial pipelines or road networks, could also be envisioned. The general method might benefit from introducing higher-order potentials [Kohli et al., 2009] into the CRF for contextual classification of primitives for merging. Concerning specifically the fallen tree detection task, it would be worthwhile to derive a more accurate tree appearance model, which would improve the robustness of the scene ranking step. Alternatively, the balance coefficients of the CRF energy terms yielding the optimal segmentation could be modeled directly, however this may prove to be even more difficult than constructing an appropriate scene ranking model.

It seems that the principal problem of the introduced parameter voting-based cylinder detection method is the somewhat difficult choice of the threshold for accepting local maxima of the kernel density estimator in the parameter space. In future research, several paths may be taken

to reduce this difficulty. First, the current method of separating the scene into independent clusters through connected components could be improved upon by applying a more geometrically aware approach. Prior information about the scene would then be used to avoid the creation of clusters containing many different cylinders, thus reducing the number of local maxima within the KDE. A complementary avenue of research is to incorporate the proposed voting method into the energy-based geometric fitting framework of Isack & Boykov [2012] in the role of a candidate cylinder generator. Then, the need to set a threshold for the KDE local maxima would be eliminated, since the framework handles selecting and disambiguating the candidate shapes on its own. The results of fallen stem mapping in TLS data also indicate that there is still much room for improvement of the cylindrical surface point classifier's transferability to plots with different characteristics. Perhaps a set of more specialized, discriminative features could be developed to address this issue. From an application perspective, it would be interesting to evaluate the performance of the fallen stem reconstruction approach for data originating from different sources than TLS, e.g. dense photogrammetric point clouds, or data from wearable scanners.

The 3D Free Shape Context descriptors have shown some promise in the role of feature generators for object classification tasks. A natural further step would be to apply FSC to related, more complex tasks, such as classification of tree species. Also, the structure of the FSC makes it particularly suitable for detecting objects with a well defined main axis, therefore applications outside of forestry within an urban setting could also be conceivable (e.g. road sign detection). The structure of the descriptors could be further enriched by introducing subdivisions along the angular dimension, however this would require an additional, external method of ensuring rotational invariance, possibly through a specialized distance metric on the FSC such as diffusion distance.

Regarding the level set segmentation, several modifications could be introduced to improve both the computational performance and the segmentation results. First, the currently applied gradient descent might exhibit a slow convergence rate toward the energy's optimum, and therefore a second order method could be preferable in the future. Moreover, the energy formulation could be enriched with a term sensitive to edges within the image, complementing the current region color term. This could help preserve the fine contour details of the tree crowns, which currently may be lost during the contour evolution process. Moreover, the introduction of a per-pixel normalized height channel alongside the spectral bands could resolve the problem of distinguishing between dead vegetation and open ground patches, which showed a somewhat similar spectral signature.

The framework combining active and semisupervised learning has up to now been tested in a two class setting. It would be interesting to apply this method to a problem with a larger number of classes to determine whether the benefits of entropy regularization carry over to the higher-dimensional case. Also, it is not clear how to determine the optimal Renyi α parameter leading to the best performance of the regularized classifier. This is left to future work. Additionally, a further integration of the example pre-selection and proper active learning could be explored. Specifically, the results showed that the greedy pre-selection step may significantly alter the class balance between the (now biased) training set and naturally occurring test data. This could be alleviated by re-weighting the classifier's loss function with instance-specific factors based on ratios of feature probabilities in the biased and unbiased case, following the methods developed for learning under covariate shift [Shimodaira, 2000].

Bibliography

Ackermann F (1999) Airborne laser scanning - present status and future expectations. ISPRS Journal of Photogrammetry and Remote Sensing, 54 (2–3): 64 – 67.

Antonarakis A, Richards K, Brasington J (2008) Object-based land cover classification using airborne LiDAR. Remote Sens. Environ., 112: 2988–2998.

Aronov B, Asano T, Katoh N, Mehlhorn K, Tokuyama T (1998) Polyline fitting of planar points under min-sum criteria. In: Atlanta, GA: U.S. Department of Health and Human Services, Public Health Service: 97–116.

Babaee M, Tsoukalas S, Rigoll G, Datcu M (2015) Visualization-Based Active Learning for the Annotation of SAR Images. Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of, PP (99): 1–1.

Bach FR, Jordan MI (2003) Learning Spectral Clustering. In: Adv. Neur. In., 16.

Baltsavias E (1999) Airborne laser scanning: existing systems and firms and other resources. ISPRS Journal of Photogrammetry and Remote Sensing, 54 (2–3): 164 – 198.

Balzter H, Rowland C, Saich P (2007) Forest canopy height and carbon estimation at Monks Wood National Nature Reserve, UK, using dual-wavelength SAR interferometry. Remote Sensing of Environment, 108 (3): 224 – 239.

Beder C, Förstner W (2006) Direct Solutions for Computing Cylinders from Minimal Sets of 3D Points. In: Proceedings of the 9th European Conference on Computer Vision - Volume Part I: 135–146.

Bellman R (1973) A note on cluster analysis and dynamic programming. Math. Biosci., 18 (3–4): 311 – 312.

Bengtsson I, Zyczkowski K (2006) Geometry of Quantum States: An Introduction to Quantum Entanglement. Cambridge University Press.

Bhattarai N, Quackenbush LJ, Calandra L, Im J, Teale SA (2012) An automated object-based approach to detect Sirex-infestation in pines. In: ASPRS Annual Conference Proceedings

Bioucas-Dias JM, Plaza A, Dobigeon N, Parente M, Du Q, Gader P, Chanussot J (2012) Hyperspectral Unmixing Overview: Geometrical, Statistical, and Sparse Regression-Based Approaches. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 5 (2): 354–379.

Blanchard SD, Jakubowski MK, Kelly M (2011) Object-Based Image Analysis of Downed Logs in Disturbed Forested Landscapes Using Lidar. Remote Sensing, 3 (12): 2420–2439.

Boyd S, Vandenberghe L (2004) Convex Optimization. New York, NY, USA: Cambridge University Press.

Breiman L (2001) Random Forests. Machine Learning, 45 (1): 5–32.

Brox T, Bruhn A, Papenberg N, Weickert J (2004) High Accuracy Optical Flow Estimation Based on a Theory for Warping. In: Pajdla T, Matas J (eds) Computer Vision - ECCV 2004, volume 3024 of Lecture Notes in Computer Science (pp. 25–36). Springer Berlin Heidelberg.

Bütler R, Schlaepfer R (2004) Spruce snag quantification by coupling colour infrared aerial photos and a GIS. Forest Ecol. Manag., 195 (3): 325 – 339.

Camps-Valls G (2009) Machine learning in remote sensing data processing. In: 2009 IEEE International Workshop on Machine Learning for Signal Processing: 1–6.

Camps-Valls G, Bioucas-Dias J, Crawford M (2016) A Special Issue on Advances in Machine Learning for Remote Sensing and Geosciences [From the Guest Editors]. IEEE Geoscience and Remote Sensing Magazine, 4 (2): 5–7.

Casas Á, García M, Siegel RB, Koltunov A, Ramírez C, Ustin S (2016) Burned forest characterization at single-tree level with airborne laser scanning for assessing wildlife habitat. Remote Sensing of Environment, 175: 231 – 241.

Chan T, Vese L (2001) Active contours without edges. IEEE T. Image Process., 10 (2): 266–277.

Cremers D, Rousson M (2007) Efficient Kernel Density Estimation Of Shape And Intensity Priors For Level Set Segmentation. In: Deformable Models, Topics in Biomedical Engineering. International Book Series (pp. 447–460). Springer, New York.

Cremers D, Rousson M, Deriche R (2007) A Review of Statistical Approaches to Level Set Segmentation: Integrating Color, Texture, Motion and Shape. Int. J. Comput. Vision, 72 (2): 195–215.

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B, 39 (1): 1–38.

Díaz-Vilariño L, Conde B, Lagüela S, Lorenzo H (2015) Automatic Detection and Segmentation of Columns in As-Built Buildings from Point Clouds. Remote Sensing, 7 (11): 15651–15667.

Duong T, Hazelton ML (2005) Cross-validation Bandwidth Matrices for Multivariate Kernel Density Estimation. Scandinavian Journal of Statistics, 32 (3): 485–506.

Elmqvist M (2002) Ground surface estimation from airborne laser scanner data using active shape models. In: Photogrammetric Computer Vision-ISPRS Commission III Symposium, XXXIV, Part A: 114–118.

Enzweiler M, Gavrila D (2008) A mixed generative-discriminative framework for pedestrian classification. In: Computer Vision and Pattern Recognition: 1–8.

Erkan AN (2010) Semi-supervised Learning via Generalized Maximum Entropy. PhD thesis, New York, NY, USA. AAI3427925.

Erkan AN, Camps-Valls G, Altun Y (2010) Semi-supervised remote sensing image classification via maximum entropy. In: 2010 IEEE International Workshop on Machine Learning for Signal Processing: 313–318.

Eskelson B, Temesgen H, Hagar J (2012) A comparison of selected parametric and imputation methods for estimating snag density and snag quality attributes. Forest Ecol. Manag., 272 (0): 26 – 34.

Fassnacht FE, Latifi H, Stereńczak K, Modzelewska A, Lefsky M, Waser LT, Straub C, Ghosh A (2016) Review of studies on tree species classification from remotely sensed data. Remote Sensing of Environment, 186: 64 – 87.

Finér L, Mannerkoski H, Piirainen S, Starr M (2003) Carbon and nitrogen pools in an old-growth, Norway spruce mixed forest in eastern Finland and changes associated with clear-cutting. Forest Ecol. Manag., 174.

Fischler MA, Bolles RC (1981) Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Commun. ACM, 24 (6): 381–395.

Fraley C, Raftery AE (2007) Bayesian Regularization for Normal Mixture Estimation and Model-Based Clustering. J. Classif., 24 (2): 155–181.

Freedman B, Zelazny V, Beaudette D, Fleming T, Johnson G, Flemming S, Gerrow JS, Forbes G, Woodley S (1996) Biodiversity implications of changes in the quantity of dead organic matter in managed forests. Environ. Rev., 4 (3): 238–265.

Friedman J, Hastie T, Tibshirani R (2010) Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, 33 (1): 1–22.

Frome A, Huber D, Kolluri R, Bülow T, Malik J (2004) Recognizing Objects in Range Data Using Regional Point Descriptors. Computer Vision - ECCV 2004, 3023: 224–237.

Goldberg DE (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1st edition.

Golovinskiy A, Funkhouser T (2009) Min-cut based segmentation of point clouds. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops: 39–46.

Grandvalet Y, Bengio Y (2006) Entropy Regularization. In: Chapelle O, Schölkopf B, Zien A (eds) Semi-Supervised Learning (pp. 151–168). MIT Press.

Grigillo D, Vrečko A, Mikoš M, Gvozdanović T, Anžur A, Vezočnik R, Petrovič D (2015) Determination of Large Wood Accumulation in a Steep Forested Torrent Using Laser Scanning, (pp. 127–130). Springer International Publishing: Cham.

Guo Y, Greiner R (2007) Optimistic Active Learning Using Mutual Information. In: Proceedings of the 20th International Joint Conference on Artifical Intelligence: 823–829.

Hammer PL, Hansen P, Simeone B (1984) Roof duality, complementation and persistency in quadratic 0–1 optimization. Mathematical Programming, 28 (2): 121–155.

Harmon M, Franklin J, Swanson F, Sollins P, Gregory S, Lattin J, Anderson N, Cline S, Aumen N, Sedell J, Lienkaemper G, Jr. KC, Cummins K (1986) Ecology of Coarse Woody Debris in Temperate Ecosystems. volume 15 of Advances in Ecological Research (pp. 133 – 302). Academic Press.

Hastie T, Tibshirani R, Friedman J (2001) The Elements of Statistical Learning. Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

Heurich M, Ochs T, Andresen T, Schneider T (2010) Object-orientated image analysis for the semi-automatic detection of dead trees following a spruce bark beetle (Ips typographus) outbreak. European Journal of Forest Research, 129 (3): 313–324.

Huo LZ, Tang P, Zhang Z, Tuia D (2015) Semisupervised Classification of Remote Sensing Images With Hierarchical Spatial Similarity. IEEE Geosci. Remote S., 12 (1): 150–154.

Hyyppä J (2011) State of the Art in Laser Scanning. In: Fritsch D (ed) Photogrammetric Week, volume '11 (pp. 133 – 302). Wichmann/VDE Verlag.

Hyyppä J, Holopainen M, Olsson H (2012) Laser Scanning in Forests. Remote Sensing, 4 (10): 2919–2922.

Isack H, Boykov Y (2012) Energy-Based Geometric Multi-model Fitting. International Journal of Computer Vision, 97 (2): 123–147.

Jensen JR (2006) Remote Sensing of the Environment: An Earth Resource Perspective (2nd Edition). Prentice Hall, Upper Saddle River.

Jutzi B, Stilla U (2006) Range determination with waveform recording laser systems using a Wiener Filter. ISPRS Journal of Photogrammetry and Remote Sensing, 61 (2): 95 – 107.

Kärkkäinen T, Äyrämö S (2005) On computation of spatial median for robust data mining. In: Schilling R, Haase W, Periaux J, Baier H, Bugeda G (eds) Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems

Ke Y, Quackenbush LJ (2011) A review of methods for automatic individual tree-crown detection and delineation from passive remote sensing. International Journal of Remote Sensing, 32 (17): 4725–4747.

Ke Y, Quackenbush LJ, Im J (2010a) Synergistic use of QuickBird multispectral imagery and LIDAR data for object-based forest species classification. Remote Sensing of Environment, 114 (6): 1141 – 1154.

Ke Y, Zhang W, Quackenbush LJ (2010b) Active Contour and Hill Climbing for Tree Crown Detection and Delineation. Photogramm. Eng. Rem. S., 76 (10): 1169–1181.

Kelly M, Shaari D, Guo Q, Liu D (2004) A Comparison of Standard and Hybrid Classifier Methods for Mapping Hardwood Mortality in Areas Affected by Sudden Oak Death. Photogrammetric Engineering & Remote Sensing, 70 (11): 1229–1239.

Klasing K, Althoff D, Wollherr D, Buss M (2009) Comparison of surface normal estimation methods for range sensing applications. In: 2009 IEEE International Conference on Robotics and Automation: 3206–3211.

Knopp J, Prasad M, Willems G, Timofte R, Van Gool L (2010) Hough Transform and 3D SURF for Robust Three Dimensional Classification, (pp. 589–602). Springer Berlin Heidelberg: Berlin, Heidelberg.

Kohli P, Ladický L, Torr PHS (2009) Robust Higher Order Potentials for Enforcing Label Consistency. International Journal of Computer Vision, 82 (3): 302–324.

Koller D, Friedman N (2009) Probabilistic Graphical Models: Principles and Techniques. MIT Press.

Kolmogorov V, Rother C (2007) Minimizing Nonsubmodular Functions with Graph Cuts-A Review. IEEE Trans. Pattern Anal. Mach. Intell., 29 (7): 1274–1279.

Korpela I, Orka HO, Maltamo M, Tokola T, Hyyppa J (2010) Tree Species Classification Using Airborne LiDAR - Effects of Stand and Tree Parameters, Downsizing of Training Set, Intensity Normalization, and Sensor Type. Silva Fennica, 44: 319 – 339.

Krzystek P, Polewski P (2017) Objektbasierte Segmentierung und Klassifikation von LiDAR-Punktwolken, (pp. 645–684). Springer Berlin Heidelberg: Berlin, Heidelberg.

Lafferty JD, McCallum A, Pereira FCN (2001) Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proceedings of the Eighteenth International Conference on Machine Learning: 282–289.

Lan G, DePuy GW, Whitehouse GE (2007) An effective and simple heuristic for the set covering problem. Eur. J. Oper. Res., 176: 1387–1403.

Lari Z, Habib A (2014) An adaptive approach for the segmentation and extraction of planar and linear/cylindrical features from laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing, 93: 192 – 212.

Lausch A, Heurich M, Fahse L (2013) Spatio-temporal infestation patterns of Ips typographus (L.) in the Bavarian Forest National Park, Germany. Ecological Indicators, 31: 73–81.

Lee S, Cho H (2006) Detection of the pine trees damaged by pine wilt disease using high spatial remote sensing data. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 36, Part 7.

Leng L, Zhang T, Kleinman L, Zhu W (2007) Ordinary least square regression, orthogonal regression, geometric mean regression and their applications in aerosol science. J. Phys. Conf. Ser., 78 (1): 012084.

Leventon M, Grimson W, Faugeras O (2000) Statistical shape influence in geodesic active contours. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1: 316–323.

Lewis DD, Catlett J (1994) Heterogeneous uncertainty sampling for supervised learning. In: Cohen WW, Hirsh H (eds) Proceedings of ICML-94, 11th International Conference on Machine Learning: 148–156.

Liang X, Hyyppä J (2013) Automatic Stem Mapping by Merging Several Terrestrial Laser Scans at the Feature and Decision Levels. Sensors, 13 (2): 1614–1634.

Liang X, Kankare V, Hyyppä J, Wang Y, Kukko A, Haggrén H, Yu X, Kaartinen H, Jaakkola A, Guan F, Holopainen M, Vastaranta M (2016) Terrestrial laser scanning in forest inventories. ISPRS Journal of Photogrammetry and Remote Sensing, 115: 63 – 77. Theme issue 'State-of-the-art in photogrammetry, remote sensing and spatial information science'.

Liang X, Litkey P, Hyyppä J, Kaartinen H, Vastaranta M, Holopainen M (2012) Automatic Stem Mapping Using Single-Scan Terrestrial Laser Scanning. IEEE Transactions on Geoscience and Remote Sensing, 50 (2): 661–670.

Lidskog R, Sjödin D (2016) Extreme events and climate change: the post-disaster dynamics of forest fires and forest storms in Sweden. Scandinavian Journal of Forest Research, 31 (2): 148–155.

Lim Y, Jung K, Kohli P (2010) Energy Minimization under Constraints on Label Counts, (pp. 535–551). Springer Berlin Heidelberg: Berlin, Heidelberg.

Lim Y, Jung K, Kohli P (2011) Constrained Discrete Optimization via Dual Space Search. In: NIPS Workshop on Discrete Optimization on Machine Learning (DISCML)

Lindberg E, Hollaus M, Mücke W, Fransson JES, Pfeifer N (2013) Detection of lying tree stems from airborne laser scanning data using a line template matching algorithm. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, II-5, Part W2: 169–174.

Lindberg E, Holmgren J (2017) Individual Tree Crown Methods for 3D Data from Remote Sensing. Current Forestry Reports, 3 (1): 19–31.

Lindberg E, Holmgren J, Olofsson K, Olsson H (2012a) Estimation of stem attributes using a combination of terrestrial and airborne laser scanning. European Journal of Forest Research, 131 (6): 1917–1931.

Lindberg E, Olofsson K, Holmgren J, Olsson Hk (2012b) Estimation of 3D vegetation structure from waveform and discrete return airborne laser scanning data. Remote Sens. Environ., 118: 151–161.

Liu TY (2009) Learning to Rank for Information Retrieval. Found. Trends Inf. Retr., 3 (3): 225–331.

Luhmann T, Robson S, Kyle S, Boehm J (2013) Close-range Photogrammetry and 3D Imaging. De Gruyter.

Magnard C, Morsdorf F, Small D, Stilla U, Schaepman ME, Meier E (2016) Single tree identification using airborne multibaseline SAR interferometry data. Remote Sensing of Environment, 186 (Supplement C): 567 – 580.

Man Q, Dong P, Guo H, Liu G, Shi R (2014) Light detection and ranging and hyperspectral data for estimation of forest biomass: a review. Journal of Applied Remote Sensing, 8 (1): 081598.

Martinuzzi S, Vierling LA, Gould WA, Falkowski MJ, Evans JS, Hudak AT, Vierling KT (2009) Mapping snags and understory shrubs for a LiDAR-based assessment of wildlife habitat suitability. Remote Sensing of Environment, 113 (12): 2533 – 2546.

McCallum A, Nigam K (1998) Employing EM and Pool-Based Active Learning for Text Classification. In: Proceedings of the Fifteenth International Conference on Machine Learning: 350–358.

Monnet J (2012) Airborne Laser Scanning for Forest Applications - State-of-the-Art. IRSTEA, Technical report.

Morsdorf F, Kötz B, Meier E, Itten KI, Allgöwer B (2006) Estimation of LAI and fractional cover from small footprint airborne laser scanning data based on gap fraction. Remote Sens. Environ., 104: 50–61.

Muecke W, Deak B, Schroiff A, Hollaus M, Pfeifer N (2013) Detection of fallen trees in forested areas using small footprint airborne laser scanning data. Can. J. Remote Sens., 39: S32–S40.

Munoz-Mari J, Tuia D, Camps-Valls G (2012) Semisupervised Classification of Remote Sensing Images With Active Queries. IEEE T. Geosci. Remote, 50 (10): 3751–3763.

Naesset E (1997) Determination of mean tree height of forest stands using airborne laser scanner data. ISPRS J. Photogramm., 52 (2): 49 – 56.

Nelson R, Krabill W, MacLean G (1984) Determining forest canopy characteristics using airborne laser data. Remote Sensing of Environment, 15 (3): 201 – 212.

Ng AY, Jordan MI (2002) On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In: Dietterich TG, Becker S, Ghahramani Z (eds) Advances in Neural Information Processing Systems 14 (pp. 841–848). MIT Press.

Nielsen MM, Heurich M, Malmberg B, Brun A (2014) Automatic Mapping of Standing Dead Trees after an Insect Outbreak Using the Window Independent Context Segmentation Method. Journal of Forestry, 112 (6): 564–571.

Niemeyer J, Rottensteiner F, Soergel U (2014) Contextual classification of lidar data and building object detection in urban areas. ISPRS Journal of Photogrammetry and Remote Sensing, 87: 152 – 165.

Nyström M, Holmgren J, Fransson JE, Olsson H (2014) Detection of windthrown trees using airborne laser scanning. Int. J. Appl. Earth Obs., 30 (0): 21 – 29.

Olofsson K, Holmgren J, Olsson H (2014) Tree Stem and Height Measurements using Terrestrial Laser Scanning and the RANSAC Algorithm. Remote Sensing, 6 (5): 4323–4344.

Pardo L (2005) Statistical Inference Based on Divergence Measures. Abingdon: CRC Press.

Pasher J, King DJ (2009) Mapping dead wood distribution in a temperate hardwood forest using high resolution airborne imagery. Forest Ecol. Manag., 258 (7): 1536 – 1548.

Pasolli E, Melgani F, Tuia D, Pacifici F, Emery W (2014) SVM Active Learning Approach for Image Classification Using Spatial Information. IEEE T. Geosci. Remote, 52 (4): 2217–2233.

Persello C, Boularias A, Dalponte M, Gobakken T, Naesset E, Schölkopf B (2014) Cost-Sensitive Active Learning With Lookahead: Optimizing Field Surveys for Remote Sensing Data Classification. IEEE T. Geosci. Remote, 52 (10): 6652–6664.

Persello C, Bruzzone L (2014) Active and Semisupervised Learning for the Classification of Remote Sensing Images. IEEE Transactions on Geoscience and Remote Sensing, 52 (11): 6937–6956.

Pesonen A, Maltamo M, Eerikäinen K, Packalèn P (2008) Airborne laser scanning-based prediction of coarse woody debris volumes in a conservation area. Forest Ecol. Manag., 255 (8–9): 3288 – 3296.

Polewski P, Erickson A, Yao W, Coops N, Krzystek P, Stilla U (2016a) Object-based coregistration of terrestrial photogrammetric and ALS point clouds in forested areas. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, III-3: 347–354.

Polewski P, Yao W, Heurich M, Krzystek P, Stilla U (2014) Detection of fallen trees in ALS point clouds by learning the Normalized Cut similarity function from simulated samples. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, II-3: 111–118.

Polewski P, Yao W, Heurich M, Krzystek P, Stilla U (2015a) Active learning approach to detecting standing dead trees from ALS point clouds combined with aerial infrared imagery. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW): 10–18.

Polewski P, Yao W, Heurich M, Krzystek P, Stilla U (2015b) Detection of fallen trees in ALS point clouds using a Normalized Cut approach trained by simulation. ISPRS Journal of Photogrammetry and Remote Sensing, 105: 252–271.

Polewski P, Yao W, Heurich M, Krzystek P, Stilla U (2015c) Detection of single standing dead trees from aerial color infrared imagery by segmentation with shape and intensity priors. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, II-3/W4: 181–188.

Polewski P, Yao W, Heurich M, Krzystek P, Stilla U (2015d) Free Shape Context descriptors optimized with genetic algorithm for the detection of dead tree trunks in ALS point clouds. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, II-3/W5: 41–48.

Polewski P, Yao W, Heurich M, Krzystek P, Stilla U (2016b) Combining Active and Semisupervised Learning of Remote Sensing Data Within a Renyi Entropy Regularization Framework. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 9 (7): 2910–2922.

Polewski P, Yao W, Heurich M, Krzystek P, Stilla U (2017a) Learning a constrained conditional random field for enhanced segmentation of fallen trees in ALS point clouds. ISPRS Journal of Photogrammetry and Remote Sensing. Article in press.

Polewski P, Yao W, Heurich M, Krzystek P, Stilla U (2017b) A voting-based statistical cylinder detection framework applied to fallen tree mapping in terrestrial laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 129: 118–130.

Rabbani T, van den Heuvel F (2005) Efficient hough transform for automatic detection of cylinders in point clouds. In: IAPRS, XXXVI, 3/W19: 60–65.

Rabbani T, van den Heuvel FA, Vosselmann G (2006) Segmentation of point clouds using smoothness constraint. In: IEVM06

Rajan S, Ghosh J, Crawford M (2008) An Active Learning Approach to Hyperspectral Data Classification. Geoscience and Remote Sensing, IEEE Transactions on, 46 (4): 1231–1242.

Raumonen P, Kaasalainen M, Åkerblom M, Kaasalainen S, Kaartinen H, Vastaranta M, Holopainen M, Disney M, Lewis P (2013) Fast Automatic Precision Tree Models from Terrestrial Laser Scanner Data. Remote Sensing, 5 (2): 491–520.

Reitberger J, Krzystek P, Stilla U (2008) Analysis of full waveform LIDAR data for the classification of deciduous and coniferous trees. Int. J. Remote Sens., 29: 1407–1431.

Reitberger J, Schnörr C, Krzystek P, Stilla U (2009) 3D segmentation of single trees exploiting full waveform LIDAR data. ISPRS J. Photogramm., 64 (6): 561 – 574.

Riegl GmbH (2017) LMSQ680i Data Sheet.

Rother C, Kolmogorov V, Lempitsky V, Szummer M (2007) Optimizing Binary MRFs via Extended Roof Duality. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition: 1–8.

Roy N, McCallum A (2001) Toward Optimal Active Learning Through Sampling Estimation of Error Reduction. In: Proceedings of the Eighteenth International Conference on Machine Learning: 441–448.

Rusu R, Marton Z, Blodow N, Beetz M (2008) Learning informative point classes for the acquisition of object model maps. In: 10th International Conference on Control, Automation, Robotics and Vision: 643–650.

Rusu RB, Cousins S (2011) 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA)

Sandberg G, Ulander L, Fransson J, Holmgren J, Toan TL (2011) L- and P-band backscatter intensity for biomass retrieval in hemiboreal forest. Remote Sensing of Environment, 115 (11): 2874 – 2886. DESDynI VEG-3D Special Issue.

Schilling A, Schmidt A, Maas HG (2011) Automatic Tree Detection and Diameter Estimation in Terrestrial Laser Scanner Point Clouds. In: Proceedings of the 16th Computer Vision Winter Workshop. Mitterberg, Austria

Schmitt M, Stilla U (2014) Maximum-likelihood estimation for multi-aspect multi-baseline SAR interferometry of urban areas. ISPRS Journal of Photogrammetry and Remote Sensing, 87 (Supplement C): 68 – 77.

Schnabel R, Wahl R, Klein R (2007) Efficient RANSAC for Point-Cloud Shape Detection. Computer Graphics Forum, 26 (2): 214–226.

Settles B (2009) Active Learning Literature Survey. University of Wisconsin–Madison, Computer Sciences Technical Report 1648.

Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE T. Pattern Anal., 22 (8): 888—-905.

Shimodaira H (2000) Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of Statistical Planning and Inference, 90 (2): 227 – 244.

Smith R (2007) Open Dynamics Engine.

Smola AJ, Schölkopf B (2000) Sparse Greedy Matrix Approximation for Machine Learning. In: Proceedings of the Seventeenth International Conference on Machine Learning: 911–918.

Standish M (1945) The Use of Aerial Photographs in Forestry. Journal of Forestry, 43 (4): 252–257.

Stilla U, Schmitt M, Maksymiuk O, Auer S (2014) Towards the recognition of individual trees in decimeter-resolution airborne millimeterwave SAR. In: 2014 8th IAPR Workshop on Pattern Reconition in Remote Sensing: 1–4.

Stoker JM, Abdullah QA, Nayegandhi A, Winehouse J (2016) Evaluation of Single Photon and Geiger Mode Lidar for the 3D Elevation Program. Remote Sensing, 8 (9).

Stokland J, Siitonen J, Jonsson B (2012) Biodiversity in Dead Wood. Ecology, Biodiversity and Conservation. Cambridge University Press.

Stone CJ (1980) Optimal Rates of Convergence for Nonparametric Estimators. Ann. Statist., 8 (6): 1348–1360.

Sundseth K, Creed P (2008) Natura 2000: Protecting Europe's Biodiversity. European Commission, Directorate General for the Environment.

Swatantran A, Tang H, Barrett T, DeCola P, Dubayah R (2016) Rapid, High-Resolution Forest Structure and Terrain Mapping over Large Areas using Single Photon Lidar. Scientific Reports, 6: 28277.

Szeliski R, Zabih R, Scharstein D, Veksler O, Kolmogorov V, Agarwala A, Tappen M, Rother C (2006) A Comparative Study of Energy Minimization Methods for Markov Random Fields, (pp. 16–29). Springer Berlin Heidelberg: Berlin, Heidelberg.

Toldo R, Castellani U, Fusiello A (2009) A Bag of Words Approach for 3D Object Categorization. In: Gagalowicz A, Philips W (eds) Computer Vision/Computer Graphics Collaboration Techniques, volume 5496 of Lecture Notes in Computer Science (pp. 116–127). Springer Berlin Heidelberg.

Torr PHS, Zisserman A (2000) MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. Computer Vision and Image Understanding, 78: 138–156.

Tran TT, Cao VT, Laurendeau D (2015) Extraction of cylinders and estimation of their parameters from point clouds. Computers & Graphics, 46: 345 – 357. Shape Modeling International 2014.

Tsai A, Yezzi A, III WW, Tempany C, Tucker D, Fan A, Grimson W, Willsky A (2003) A Shape-based Approach to the Segmentation of Medical Imagery using Level Sets. IEEE T. Med. Imaging, 22 (2): 137–154.

Tucker CJ (1979) Red and photographic infrared linear combinations for monitoring vegetation. Remote Sens. Environ., 8: 127–150.

Tuia D, Merenyi E, Jia X, Grana-Romay M (2014) Foreword to the Special Issue on Machine Learning for Remote Sensing Data Processing. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 7 (4): 1007–1011.

Tuia D, Ratle F, Pacifici F, Kanevski MF, Emery WJ (2009) Active Learning Methods for Remote Sensing Image Classification. IEEE Transactions on Geoscience and Remote Sensing, 47 (7): 2218–2232.

Tuia D, Volpi M, Copa L, Kanevski M, Munoz-Mari J (2011) A Survey of Active Learning Algorithms for Supervised Remote Sensing Image Classification. IEEE J. Sel. Top. Signa., 5 (3): 606–617.

Tur G, Hakkani-Tür D, Schapire RE (2005) Combining active and semi-supervised learning for spoken language understanding. Speech Commun., 45 (2): 171–186.

Ueda N, Nakano R (1998) Deterministic annealing EM algorithm. Neural Networks, 11 (2): 271 – 282.

Vanco M (2002) A Direct Approach for the Segmentation of Unorganized Points and Recognition of Simple Algebraic Surfaces. PhD thesis, University of Technology Chemnitz.

Vatsavai RR, Shekhar S, Burk TE (2005) A semi-supervised learning method for remote sensing data mining. In: 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05): 5 pp.–211.

Vehmas M, Packalén P, Maltamo M, Eerikäinen K (2011) Using airborne laser scanning data for detecting canopy gaps and their understory type in mature boreal forest. Ann. For. Sci., 68 (4): 825–835.

Vogelmann J (1990) Comparison between two vegetation indices for measuring different types of forest damage in the north-eastern United States. Int. J. Remote Sens., 11 (12): 2281–2297.

Volpi M, Ferrari V (2015) Semantic Segmentation of Urban Scenes by Learning Local Class Interactions. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops

Wagner W, Ullrich A, Ducic V, Melzer T, Studnicka N (2006) Gaussian decomposition and calibration of a novel small-footprint full-waveform digitising airborne laser scanner. ISPRS Journal of Photogrammetry and Remote Sensing, 60 (2): 100 – 112.

Wahba G (1975) Optimal Convergence Properties of Variable Knot, Kernel, and Orthogonal Series Methods for Density Estimation. Ann. Statist., 3 (1): 15–29.

Wan L, Tang K, Li M, Zhong Y, Qin AK (2015) Collaborative Active and Semisupervised Learning for Hyperspectral Remote Sensing Image Classification. IEEE Transactions on Geoscience and Remote Sensing, 53 (5): 2384–2396.

Wand MP, Jones C (1994) Multivariate plug-in bandwidth selection. Computational Statistics, 9 (2): 97–116.

Wang B, Wang X (2007) Bandwidth Selection for Weighted Kernel Density Estimation. ArXiv e-prints.

Wang C, Lu Z, Haithcoat TL (2007) Using Landsat images to detect oak decline in the Mark Twain National Forest, Ozark Highlands. Forest Ecol. Manag., 240 (1–3): 70 – 78.

Wang D, Hollaus M, Puttonen E, Pfeifer N (2016) Automatic and Self-Adaptive Stem Reconstruction in Landslide-Affected Forests. Remote Sensing, 8 (12).

Wang D, Kankare V, Puttonen E, Hollaus M, Pfeifer N (2017) Reconstructing Stem Cross Section Shapes From Terrestrial Laser Scanning. IEEE Geoscience and Remote Sensing Letters, 14 (2): 272–276.

Wang L, Hao S, Wang Y, Lin Y, Wang Q (2014) Spatial-Spectral Information-Based Semisupervised Classification Algorithm for Hyperspectral Imagery. Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of, 7 (8): 3577–3585.

Waske B, Fauvel M, Benediktsson JA, Chanussot J (2009) Machine Learning Techniques in Remote Sensing Data Analysis, (pp. 1–24). John Wiley & Sons, Ltd.

Wasserman L (2010) All of Statistics: A Concise Course in Statistical Inference. Springer Publishing Company, Incorporated.

Weaver JK, Kenefic LS, Seymour RS, Brissette JC (2009) Decaying wood and tree regeneration in the Acadian Forest of Maine, USA. Forest Ecol. Manag., 257: 1623–1628.

Wegner JD, Hansch R, Thiele A, Soergel U (2011) Building Detection From One Orthophoto and High-Resolution InSAR Data Using Conditional Random Fields. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 4 (1): 83–91.

Wegner JD, Montoya-Zegarra JA, Schindler K (2015) Road networks as collections of minimum cost paths. ISPRS Journal of Photogrammetry and Remote Sensing, 108: 128 – 137.

Weinmann M, Jutzi B, Hinz S, Mallet C (2015a) Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. ISPRS Journal of Photogrammetry and Remote Sensing, 105: 286 – 304.

Weinmann M, Schmidt A, Mallet C, Hinz S, Rottensteiner F, Jutzi B (2015b) Contextual classification of point cloud data by exploiting individual 3D neighborhoods. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, II-3/W4: 271–278.

Wilkes P, Lau A, Disney M, Calders K, Burt A, de Tanago JG, Bartholomeus H, Brede B, Herold M (2017) Data acquisition considerations for Terrestrial Laser Scanning of forest plots. Remote Sensing of Environment, 196 (Supplement C): 140 – 153.

Wing BM, Ritchie MW, Boston K, Cohen WB, Olsen MJ (2015) Individual snag detection using neighborhood attribute filtered airborne lidar data. Remote Sensing of Environment, 163: 165 – 179.

Woodall C, Heath L, Smith J (2008) National inventories of down and dead woody material forest carbon stocks in the United States: Challenges and opportunities. Forest Ecol. Manag., 256 (3): 221–228.

Woodford OJ, Rother C, Kolmogorov V (2009) A global perspective on MAP inference for low-level vision. In: 2009 IEEE 12th International Conference on Computer Vision: 2319–2326.

Xie Y, Sha Z, Yu M (2008) Remote sensing imagery in vegetation mapping: a review. Journal of Plant Ecology, 1 (1): 9.

Xu D, Erdogmuns D (2010) Renyi's Entropy, Divergence and Their Nonparametric Estimators. In: Information Theoretic Learning, Information Science and Statistics (pp. 47–102). Springer New York.

Yang MY, Förstner W (2011) A hierarchical conditional random field model for labeling and classifying images of man-made scenes. In: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on: 196–203.

Yao W, Krzystek P, Heurich M (2012) Identifying Standing Dead Trees in Forest Areas Based on 3D Single Tree Detection From Full Waveform Lidar Data. In: ISPRS Annals of the Protogrammetry, Remote Sensing and Spatial Information Sciences, I-7: 359–364.

Yu D, Varadarajan B, Deng L, Acero A (2009) Active Learning and Semi-supervised Learning for Speech Recognition: A Unified Framework using the Global Entropy Reduction Maximization Criterion. Computer Speech and Language - Special Issue on Emergent Artificial Intelligence Approaches for Pattern Recognition in Speech and Language Processing.

Zhang Y, Yang H, Prasad S, Pasolli E, Jung J, Crawford M (2015) Ensemble Multiple Kernel Active Learning For Classification of Multisource Remote Sensing Data. Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of, 8 (2): 845–858.

Zhen Z, Quackenbush LJ, Zhang L (2016) Trends in Automatic Individual Tree Crown Detection and Delineation—Evolution of LiDAR Data. Remote Sensing, 8 (4).

Zhu X, Lafferty J, Ghahramani Z (2003) Combining Active Learning and Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In: ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining: 58–65.

Curriculum Vitae

Name	Przemyslaw Polewski
Geburtstag und -ort	03.02.1984 in Poznań, Polen
Staatsangehörigkeit	polnisch

Ausbildung/Tätigkeit

1999 – 2003	Gymnasium in Poznań, Polen Abschluss: Allgemeine Hochschulreife
2003 – 2007	Studium der Informatik, Poznań University of Technology, Polen. Abschluss: BSc.-Eng.
2007 – 2008	Studium der Informatik, Poznań University of Technology, Polen. Abschluss: MSc.-Eng.
2009 – 2012	Mitarbeiter bei Emerson Process Management PWS, R&D Department, Warszawa, Polen
2012 – 2013	Mitarbeiter bei BMW AG, Abteilung Softwarearchitektur Fahrdynamik, München
2013 – 2017	Wissenschaftlicher Mitarbeiter im Labor Photogrammetrie und Fernerkundung, Fakultät für Geoinformation, Hochschule für angewandte Wissenschaften München
2014 – 2017	Mitglied der Graduiertenschule der Technischnen Universität München

Acknowledgments

First, I would like to extend my gratitude to Prof. Peter Krzystek, the director of the Laboratory of Photogrammetry and Remote Sensing at the Munich University of Applied Sciences, for giving me a chance to be a part of his research team. The optimal conditions for conducting research within this team contributed significantly to my successful completion of the doctoral thesis within the planned time interval of 4 years. Alongside helpful technical and scientific discussions, I would also like to thank him for constantly providing new, often unique and one-of-a-kind datasets, which, together with the prospect of presenting our research at leading international remote sensing conferences, motivated all team members to pursue new research avenues and give our 100%. In retrospect, I can only wish any remote sensing PhD student to be as lucky as myself with choosing such a stimulating and engaging research environment.

I would also like to convey my sincere thanks to Prof. Uwe Stilla, who agreed to formally supervise my doctoral work at the Technische Universität München. During the 4 years of our cooperation, Prof. Stilla was always available for a discussion regarding both scientific and formal, administrative matters, for which I am very grateful. In particular, I would like to thank him for many useful tips regarding the publishing process of scientific manuscripts, including responding to reviewer comments, properly structuring the paper, etc.

Additionally, I would like to thank Prof. Helmut Mayer of the Universität der Bundeswehr München for agreeing to act as a reviewer for my doctoral dissertation, and in particular for finding the time to read it extremely carefully, which resulted in locating a large number of errors in the text that slipped through my own quality control process. I am very thankful for his effort, which contributed significantly to enhancing the quality of the thesis text. Also, I would like to thank Prof. Thomas Wunderlich for acting in the role of the Chairman of the doctoral examination committee, and for explaining to me the intricacies of the university's regulations for awarding doctoral degrees.

Research is a team effort, and at this point I wanted to render my deepest thanks and best wishes to my friends and former colleagues from the Laboratory of Photogrammetry and Remote Sensing for a great and fruitful cooperation, but also for providing a kind and fun working environment. In particular my special thanks go to Mrs. Nina Amiri for her support and for making the greatest PhD hat ever conceived of by man. Also I would like to thank Prof. Wei Yao for many interesting discussions (not only scientific) and fun times. All this contributed to an unforgettable 4 years.

Last, but not least, I would like to thank my parents for always believing in me and for their continual support.