Detection of fallen trees in ALS point clouds of a temperate forest by combining point/primitive-level shape descriptors

PRZEMYSLAW POLEWSKI[1], WEI YAO[2], MARCO HEURICH[3], PETER KRZYSTEK[4] & UWE STILLA[5]

Abstract: In this work, we show how local 3D shape descriptors can be used to efficiently detect segments of fallen trees in LiDAR point clouds. Our approach takes advantage of two types of highly expressive classes of shape features known from computer vision: Point Feature Histograms (PFH) at the single point level as well as 3D Shape Contexts (SC) at the primitive (segment) level. The low-level point information forms a basis for determining potential segment candidates, which are then classified by a binary support vector machine (SVM). Based on manual labeling of points, we tested our method on two sample plots from the Bavarian Forest National Park acquired using full waveform LiDAR data with a point density of 30 pts/m^2. The results, obtained using 5-fold validation, show that the studied shape descriptors have a high discriminative capability for the task of detecting fallen tree segments, yielding an overall classification accuracy of above 90%.

1 Introduction

Fallen trees form an important component of forest ecosystems. Aside from supporting biodiversity by providing habitat for a wide variety of species (Freedman et al., 1996), they also play a key role in forest nutrient cycles (Finér et al., 2003) and facilitate tree regeneration (Weaver et al., 2009). Also, since carbon constitutes about 50% of wood by mass, dead wood amounts to a significant portion of forest carbon stocks (Woodall et al., 2008). For these reasons, qualitative and quantitative information about the spatial distribution of dead wood in forests is important for national government agencies as well as other organizations interested in monitoring biodiversity, carbon sequestration and wildlife habitats.

Airborne LiDAR has been shown to be a valuable tool in assessing various forest parameters at different scales. In previous studies, LiDAR data were used to efficiently and accurately derive parameters such as leaf area index (LAI) (Morsdorf et al., 2006), forest land cover characteristics (Antonarakis et al., 2008) and 3D vegetation structure (Lindberg et al., 2012) at the plot level.

1) Przemyslaw Polewski[a,c] , E-Mail: polewski@hm.edu
2) Wei Yao[a,c], E-Mail: yao@hm.edu
3) Marco Heurich[b], E-Mail: Marco.Heurich@npv-bw.bayern.de
4) Peter Krzystek[a], E-Mail: krzystek@hm.edu
5) Uwe Stilla[c], E-Mail: stilla@tum.de

[a] Munich University of Applied Sciences, Faculty of Geoinformatics, Karlstraße 6, 80333 München
[b] Bavarian Forest National Park, Department of Research and Documentation, Freyunger Straße 2, 94481 Grafenau
[c] Photogrammetry and Remote Sensing, Technische Universität München, Arcisstr. 21, 80333 München

There have also been many contributions dealing with extracting individual trees from the LiDAR data. The segmentations of the point clouds were then used for tree species classification (Reitberger et al., 2008; Heurich, 2008), detection of standing dead trees (Yao et al., 2012a) as well as estimation of tree parameters such as stem volume and DBH (Yao et al., 2012b) or biomass (Kankare et al., 2013). In this work, we focus on detecting 3D segments of fallen trees directly from LiDAR point clouds.

Due to the importance of obtaining good estimates of dead wood distributions in forests on the one hand and the high cost of laborious field surveys on the other, the potential in applying remote sensing methods to this task has been noticed by the research community. One of the first studies on this subject is from Blanchard et al. (2011). This approach relies on rasterizing the point cloud with respect to various point statistics as well as generating a vector thematic layer for object based image analysis to detect downed stems. The authors observe that their method has some difficulty in more complex scenarios like proximity of the stems to ground vegetation as well as large clusters of logs. Also, the entire processing pipeline relies on multiple user-defined parameters, making it more cumbersome to apply for an entirely new plot. Mücke 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. This study does not address the more complex scenario of multiple overlapping stems. In a recent study (Lindberg et al., 2013), the authors perform line template matching directly in the point cloud, followed by height homogeneity analysis on rasterized versions of the found lines. An attempt to model the local neighborhood of stem candidates is made using a height statistic on points lying within a fixed radius. Upon validation on a complex test site, they find that although detecting the lines directly in 3D is advantageous, the method sometimes fails in the presence of large clusters of overlapping stems, and can also report false positives in dense vegetation, ditches or road fragments. Also, a multitude of user-defined thresholds restricts its application for other areas.

To mitigate the aforementioned deficiencies, we propose a new method for detecting stem segments directly in the point cloud, without any information loss resulting from a rasterization step. We argue that the 3D information can be leveraged using robust shape descriptors originating in the computer vision and robotics communities: Shape Contexts and Point Feature Histograms. We believe that explicitly modeling the shape of the stem segments and their immediate neighborhood can improve detection capabilities in complex scenarios like piles of overlapping logs and dense ground vegetation or regeneration. We show how to automatically learn the shape characteristics of stem points and segments, and use these learned characteristics to produce a high-quality classifier. The proposed method is designed to be an intermediate step in a comprehensive procedure for detecting entire fallen trees, since its focus is the detection of

parts of stems only. In this work, we do not address the problem of selecting an optimal set of non-overlapping segments or the merging of segments.

This paper is structured as follows. In Section 2, we describe the details of our approach. In Section 3, a description of the study area within the Bavarian Forest National Park is given. Section 4 presents the quantitative classification results. We conclude with pointing out some interesting research directions resulting from this study in Section 5.

2 Proposed method

We develop an automated approach to fallen tree stem detection by combining single point features with stem segment-level features in a mixed probabilistic-discrete classification framework. The processing pipeline consists of several steps, outlined in Fig.1. Our method accepts a set of 3D point coordinates (unstructured point cloud) as input and produces a set of possibly overlapping 3D line segments (corresponding to parts of fallen tree stems) as output.

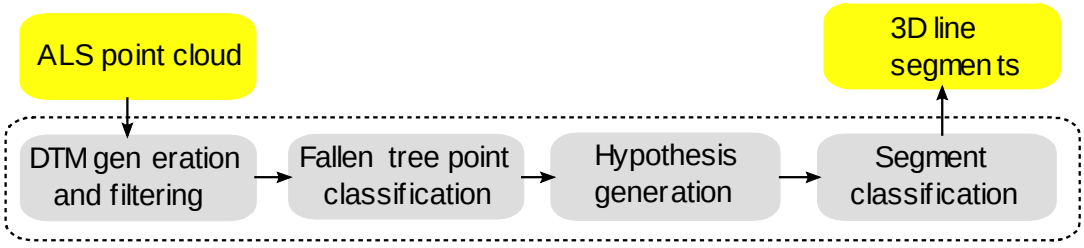

Fig. 1: Outline of proposed method

2.1 DTM generation and filtering

Since objects of the target class (fallen trees) are located in close proximity to the ground, obtaining an accurate DTM is crucial to the overall success of the method. For this purpose, we adapt a DTM calculation method based on Active Shape Models (ASM) reported in (Elmqvist, 2002). In this setting, the terrain model is cast as an elevation function defined over a discrete, evenly spaced grid. We can associate a surface energy with each such elevation function. The energy functional is then defined as:

$$E(z) = \sum_{p} E_{img}(z_p) + E_{\text{int}}(z_p) + E_{ext}(z_p) \qquad (1)$$

The summing is done over the discrete grid points. In the above, z_p is the model height of the terrain at point p. E_{img} denotes the image energy and is usually defined as the sum of distances between the measured heights and the model at every grid point (w.r.t a chosen metric). E_{int} is the internal energy of the surface. It introduces elasticity/rigidity constraints into the model, thereby controlling the surface's smoothness. Finally, E_{ext} can be used to model additional, external constraints. We do not use external energy in this work.

In this study, the pipeline for generating the DTM is as follows. We start by performing a Gaussian decomposition of the input LiDAR waveforms (Reitberger et al., 2008) and retain single and last pulse points only. We then partition the area defined by the 2D bounding box of the input points into square cells of width 10 cm. The cell centers correspond to the unknowns

$\{z_p\}$ in eq. (1). For every cell c_k, we define the measured height as the minimum height of all points p belonging to c_k: $H_k = \min_{p \in C_k} h_p$. Finally, we minimize the functional (1) to obtain the DTM. Once the DTM has been calculated, we proceed to retain only points lying within an interval of 15 to 100 cm over the ground.

2.2 Fallen tree point classification

After the DTM filtering from step 1, usually points from several different object classes are still contained in the input data. Depending on the characteristics of the forest stand, this can be shrubs, ground vegetation, regenerations, standing/fallen tree stems and others. The goal of this step is to perform a soft binary classification of the filtered points into the classes 'fallen tree segment' and 'others'. The classification is soft in the sense that only a pseudo-probability value is obtained for each point, estimating its likelihood of belonging to the fallen tree stem class:

$$p_{stem}(p_k) \equiv P(y_k = 'stem' | \bar{x}_k) \qquad (2)$$

In eq. (2), the value p_{stem} can be understood as the conditional probability of point p_k being a fallen stem point given the vector x_k containing the values of its features (independent variables).

There are several classification methods which provide a pseudo-probability estimate, e.g. probabilistic SVMs (Chang & Lin, 2011), random forests (Breiman, 2001) and algorithms employing Stochastic Gradient Boosting (Friedman, 2002). We use an R implementation of Real AdaBoost based on classification trees, made available through the *ada* package (Culp et al., 2007). A sample stem probability map is depicted in Fig. 2.

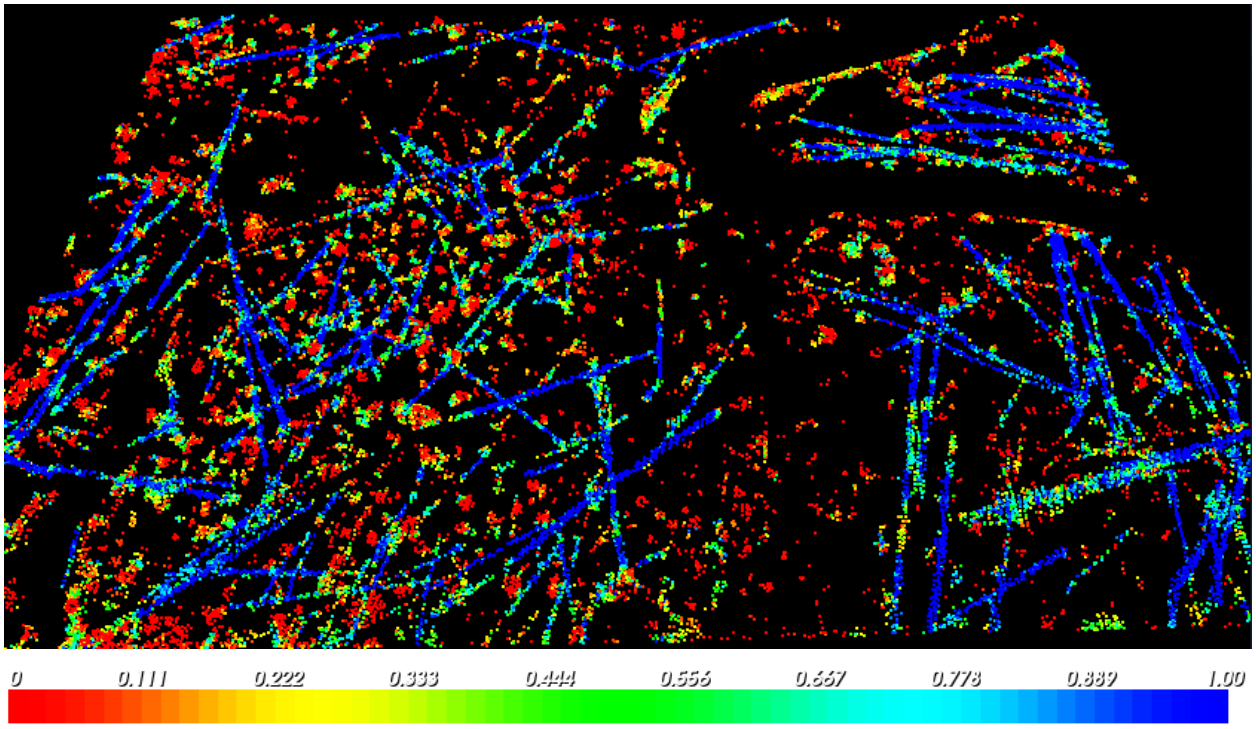

Fig. 2. Fallen tree stem probability map. Color bar encodes probabilities. Low probability points are most likely associated with ground vegetation and forest regeneration.

As features for classification, we use Point Feature Histograms (PFH) (Rusu et al., 2008). This is a local shape descriptor introduced in the robotics community for the purpose of distinguishing between different types of surface classes based on their shape (flat, cylindrical, spherical etc.). It attempts to capture the geometric properties of a surface patch centered on the target point by modeling the relationships between surface normals on that patch. Specifically, the PFH calculation algorithm first estimates normal vectors for each point within a predefined radius around the target point. The estimation is carried out based on the analysis of the eigenvectors and eigenvalues of the covariance matrix formed by points within the neighborhood. Then, for every pair of normals a Darboux frame is defined and the angular differences between the normals are recorded. Finally, a multi-dimensional histogram is calculated on the stored differences, normalized by the total number of point pairs.

Some particularly appealing properties of the PFH are its invariance to 3D translation and rotation, as well as, to a certain degree, robustness to varying point density and noise. This can be attributed to the fact that the PFH captures more information about a point's neighborhood than a single curvature or surface normal estimate.

We employ the Point Cloud Library implementation of PFH (Rusu & Cousins, 2011) with 5 value intervals per angular difference. Since there are 3 angular differences per normal vector pair, the total dimensionality of the feature vector is 5^3=125.

2.3 Segment hypothesis generation

In this step, the goal is to select probable fallen tree segment hypotheses which will be classified in the next stage. We restrict our attention to points whose stem likelihood from the previous step is above p_{thr}=0.5. We then exhaustively examine all pairs of such points whose Euclidean distance lies below a predefined threshold L=3m. Each such point pair determines a line in 3 dimensions (Fig. 3). A segment of length L of the line centered at the centroid of the base point pair is further considered. A cylindrical volume with a fixed radius r=0.3m around the line segment is examined and all points inside the cylinder are considered the support of the segment hypothesis. Only hypotheses whose support size is above a threshold value s_{thr}=20 points are retained for further processing. The rationale behind this method of generating hypotheses is that for 'good' stem segments, there should usually be a line connecting two points that lie near the segment's main axis.

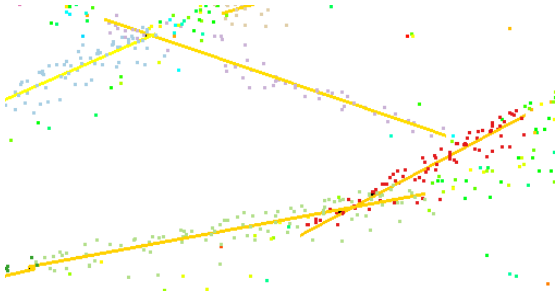

Fig. 3. Example stem segments in point cloud

2.4 Segment classification

The final stage of our method is designed to combine point-level information obtained from step 2.2 with segment-level shape information in order to classify the underlying segments into the classes 'good' and 'bad'. The 'good' class corresponds to segments which are indeed part of a fallen stem, whereas the 'bad' class represents invalid segments, such as fragments of ground vegetation, shrubs etc. We would like the features for classification to capture certain intuitions about 'good' segments: 1) they consist mostly of high-probability stem points and 2) they are sets of points elongated along one direction, surrounded by empty space above and below.

To satisfy Requirement 1, we construct a low-dimensional histogram of the fallen tree stem probabilities p_{stem} (eq. 2) of points comprising a segment's support (=FTPH – fallen tree probability histogram). Specifically, we divide the allowed probability range [0; 1] into N_{div}=5 equally spaced bins. The histogram is normalized by the number of points j belonging to the segment S (eq. 3).

$$F_{FTPH}(S) = \{ f^{i}_{FTPH}(S) \}_{1 \le i \le N_{div}}, \; f^{i}_{FTPH} \equiv \frac{| \, j : j \in S \wedge \Delta(i-1) \le p_{stem}(j) \le \Delta i \, |}{| \, j : j \in S \, |}, \Delta \equiv \frac{1}{N_{div}} \quad (3)$$

Our strategy for implementing Requirement 2 builds upon another highly expressive shape descriptor, known in the computer vision community as the 3D Shape Context (SC) (Frome et al., 2004). The original SC is a point-level descriptor which attempts, not unlike the PFH, to capture the properties of the target point's neighborhood. The outline of this method is as follows: a sphere centered on the target point *p* is defined, having its north pole oriented according to the estimated surface normal at *p*. The sphere is divided into bins equally spaced in the azimuth and elevation dimensions as well as logarithmically spaced in the radial dimension. Each point p_k within the sphere contributes only to the bin which it is enclosed by. The contribution of p_k is inversely proportional to the cube root of the bin's volume and to the estimated point density around p_k. The set of accumulated values in each bin forms a feature vector whose dimension is equal to the product of the number of azimuth, elevation and radial subdivisions of the sphere.

We adapt the 3D SC to describe the shape of a stem segment (=SSC – segment shape context). We make use of the oriented frame of reference which is naturally determined by a line segment in 3D. Instead of the original spherical neighborhood, we use a cylinder whose main axis is the segment itself. Similar to (Frome et al., 2004), we define 3 dimensions for bin boundaries: the evenly spaced angular and axial and the logarithmic radial dimension. The number of angular, axial and radial subdivisions is 6, 10 and 3, respectively, for a total of 180 features. Note that if we consider the cylinder's axis as the X axis of a Cartesian coordinate system, the SSC is invariant under rotation about any axis parallel to either Y or Z, but not X. This means that two symmetrical areas on the stem surface can in general produce different descriptors and thus SSC does not intrinsically handle symmetry. This emphasizes the importance of providing training data covering the entire range of expected object appearances.

We employ a standard binary Support Vector Machine based on FTPH and SSC features to perform crisp classification of the segments generated in step 2.3.

3 Experiment

3.1 Area of study

The developed approach was tested on two nearby sample plots located in the Bavarian Forest National Park (49° 3′ 19″ N, 13° 12′ 9″ E), which is located in South-Eastern Germany along the border to the Czech Republic. 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 not removed from the area and form the basis for the study. 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) to obtain a 3D point cloud. The characteristics of the sample plots are summarized in Table 1. The ALS point cloud associated with Plot B is depicted in Fig. 4.

Table 1: Characteristics of sample plots

	Time	Size [ha]	Trees/ha	Deciduous [%]	Fallen tree [stems/ha]	Overstory cover [%]
Plot A	2011	1.2	230	28	200	20
Plot B	2011	0.86	210	45	150	65

Due to the unavailability of field work data or other parameter estimation sources, the plot parameters were estimated from the ALS data using visual inspection methods.

3.2 Experimental setup

Regarding the point-based classification (Section 2.2), we used Plot B as a basis for generating training/test data. For this purpose, we manually labeled the points from this plot as ‘fallen tree stem points’ and ‘others’ after performing DTM filtering (as described in Section 2.1). The resulting points, numbering 33741, were divided into the train and test sets, with a proportion of 40% to 60% respectively.

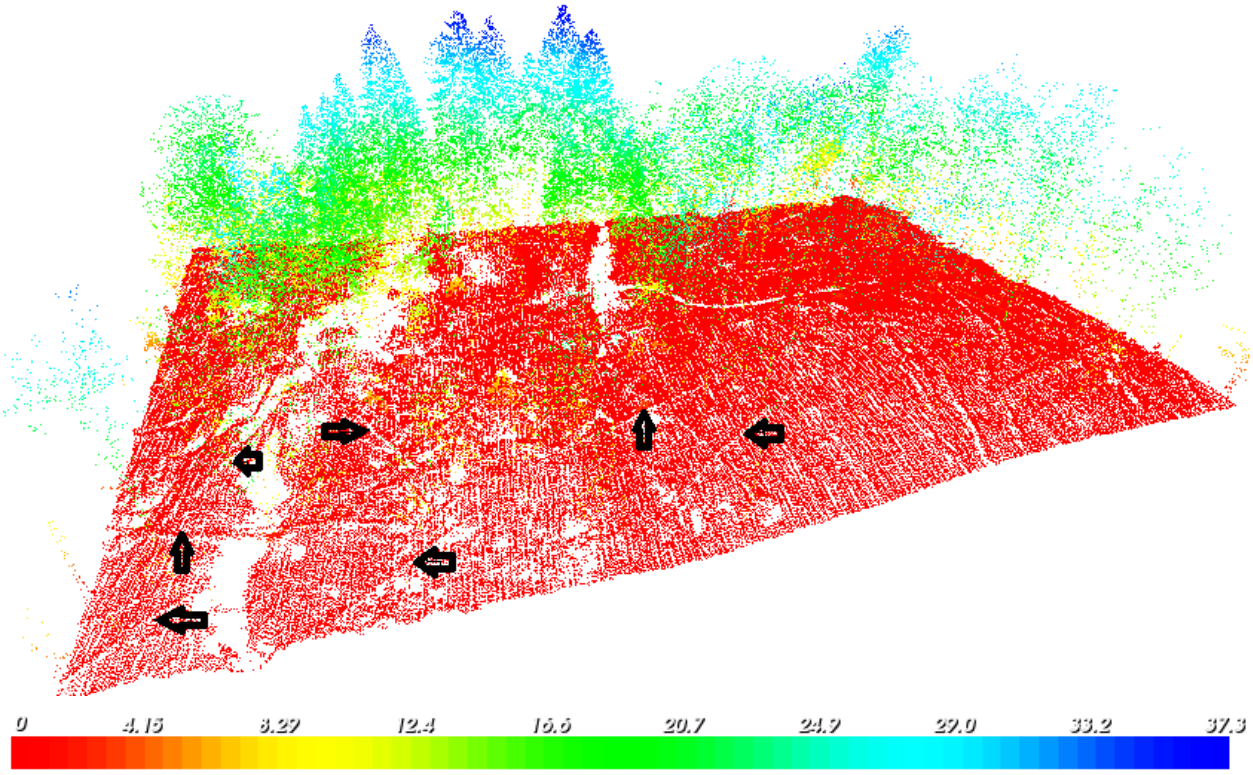

Fig. 4. ALS point cloud of Plot B colored by height over ground. Black arrows show fallen tree stems.

Both the training and test sets were quite class-balanced, with the percentage of fallen stem points being 51.2% and 51.1% respectively. The classifier (AdaBoosted classification trees, see

Section 2.2) trained and tested on Plot B was also used in an unaltered state for processing Plot A. The radius for normals estimation and PFH calculation was 50 cm and 80 cm, respectively.

Both Plot A and Plot B were considered in the segment-based classification (Section 2.4). For each plot, the steps described in Sections 2.1 to 2.3 were executed, yielding a set of segment candidates. Due to the large number of highly overlapping candidates (several hundred thousand), a subset of representative candidates was chosen for each plot. In total, the number of segment candidates selected for labeling was 1105 and 558 for Plot A and Plot B, respectively. These segments were then manually classified as either 'good' or 'bad' (see Section 2.4). The percentage of 'good' segments was 53.0% and 52.5% for Plot A and Plot B, respectively. In contrast to the point-based classification scheme, the labeled segments were not partitioned into training and test sets. Instead, a 5-fold cross-validation strategy was used. A grid search was performed on the SVM parameters γ (kernel parameter) and C (cost of constraint violation) in order to find the best performing classifier.

4 Results and Discussion

In the following, we define the overall classification accuracy as the ratio of the number of correctly classified objects to the total number of objects to classify. The producer's accuracy for a class *c*, also known as completeness in a binary classification setting, is the ratio of the number of true positives (relative to *c*) to the total number of objects actually belonging to *c*. The consumer's accuracy for a class c, also known as correctness in a binary classification setting, is the ratio of the number of true positives (relative to *c*) to the number of objects classified as *c*.

4.1 Point classification

Table 2 summarizes the crisp classification performance of points from Plot B as described in Section 3.2. It should be noted that our algorithm does not perform a discrete classification, but instead it uses the class membership pseudo-probabilities p_{stem} obtained from the classifier (Section 2.2) in several processing steps. These results are provided for the sake of assessing the suitability of the chosen feature set (PFH) as a basis for providing sound estimates of p_{stem}. It is assumed that a high classification accuracy will result in reliable probability estimates.

The overall accuracy and Cohen's kappa coefficient are high for both the training (0.99/0.98) and test (0.88/0.76) sets, indicating that the PFH features indeed capture enough of the characteristics of the points' neighborhoods to produce a good decision boundary. Note that there is a gap of 0.11 between the performance of the classifier on the training and test set. This may be attributed to several factors. First, the point labeling was done by a human expert and is thus inherently subjective. Furthermore, there was much variability in the appearance of the fallen stems in the LiDAR data (proximity to ground vegetation, overlapping with other stems etc.). Also, the high dimensionality of the PFH descriptor means that the feature space is quite sparse, increasing the risk of overfitting. Finally, due to the prohibitive computational cost of training the AdaBoosted classification trees on a large dataset, we did not explore the classifier's tuning parameter space as thoroughly as in case of the much smaller segment dataset, therefore there might be a parameter combination yielding marginally better generalization capabilities.

As illustrated by Table 2, the classifier is not heavily biased towards any of the two classes. The producer's and consumer's accuracies are nearly identical in the training set, whereas their maximum deviation in the test set is 2.3 percentage points.

Table 2: Overall accuracy, kappa coefficient and confusion matrix for stem point classification

	Training set			Test set		
	Non-stem (actual)	Stem (actual)	Consumer's accuracy	Non-stem (actual)	Stem (actual)	Consumer's accuracy
Non-stem (predicted)	6529	77	98.8%	8570	1147	88.2%
Stem (predicted)	54	6837	99.2%	1326	9201	87.4%
Producer's accuracy	99.1%	98.9%		86.6%	88.9%	

Overall accuracy	99.0%	87.7%
Kappa	0.98	0.76

While higher classification rates at this intermediate step are certainly desired, the way we use the output from the point classifier in further processing steps allows for a certain margin of error. Specifically, recall that these pseudo-probability values are used for the generation of probable segment hypotheses as well as for the classification of 'good' and 'bad' segments in the form of a coarse histogram (FTPH, Section 2.4).

As for the former, in order to consider a real segment as a good hypothesis, we require that a single pair of points along the segment's main axis obtain high probability values. Regarding the latter, the coarse histogram F_{FTPH} (Section 2.4) inherently adds smoothing to the inference process. Therefore it would seem intuitively viable that a strong majority of high-probability points, say 80-90%, should suffice to provide a solid basis for subsequent steps.

4.2 Segment classification

Table 3 shows the overall accuracy rates of the best classifiers trained on different sets of features. The accuracy rates were obtained from the 5-fold cross validation procedure. First, we note that all feature combinations attain accuracies well above 80%. Interestingly, the classifier based on the low-dimensional FTPH features performs almost identically to the one based on the high-dimensional SSC features. This indicates that each of these feature families contributes relevant information to the classification process and indeed seems to capture our intuitions mentioned in Section 2.4.

Table 3: Performance of segment classification for different combinations of features – overall accuracy from 5-fold cross validation

	FTPH	SSC	FTPH + SSC
Plot A	0.86	0.85	**0.91**
Plot B	0.86	0.84	**0.92**

Moreover, the classifier trained simultaneously on FTPH and SSC outperforms both classifiers trained on a single feature family.

Table 4 describes the performance of the classifiers which were retrained on the entire datasets. While the results are only slightly higher for Plot A, the classifier for Plot B has benefited from a significant improvement. This could be explained by the fact that Plot A has a larger area, more fallen stems and regenerations, which results in a higher variability. The subjective labeling could also be of influence.

Table 4: Performance of segment classification for best model retrained on entire data

	Plot A			Plot B		
	Bad (actual)	Good (actual)	Consumer's accuracy	Bad (actual)	Good (actual)	Consumer's accuracy
Bad (predicted)	485	46	91.3%	292	7	97.6%
Good (predicted)	40	534	93.0%	4	255	98.4%
Producer's accuracy	92.3%	92.0%		98.6%	97.3%	

Finally, we turn to the issue of the classifiers' generalization ability. We have cross-validated the classifiers on their complementary plots, i.e. tested the classifier trained using Plot A on Plot B and vice versa. The results can be seen in Table 5. The cross-plot performance drops substantially for both plots, with the lowest completeness rate of 76% ('good' segment detection) observed for using Classifier B on Plot A.

Table 5: Generalization ability – (a) classifier trained on Plot A, tested on Plot B, (b) classifier trained on Plot B, tested on Plot A

	(a) Plot A (train) / Plot B (test)			(b) Plot B (train) / Plot A (test)		
	Bad (actual)	Good (actual)	Consumer's accuracy	Bad (actual)	Good (actual)	Consumer's accuracy
Bad (predicted)	289	63	82.1%	416	14	96.7%
Good (predicted)	7	199	96.6%	109	566	83.3%
Producer's accuracy	97.6%	75.9%		79.2%	97.5%	

It is interesting to note that Classifiers A and B are biased in opposite directions. Classifier A clearly prefers labeling segments as 'bad', achieving a completeness of 98% in this class and assigning 25% of 'good' segments the 'bad' label. Conversely, Classifier B exhibits a similar (in terms of proportions) tendency to favor the 'good' class. Despite the seemingly high overall accuracies of 87-88%, the values of Cohen's kappa coefficient are relatively low at 0.37 and 0.38. Possible causes for this difference in performance include the influence of subjective labeling, different plot characteristics as well as limited transferability of the trained classifiers.

5 Conclusion and Outlook

We have demonstrated the suitability of two types of 3D shape descriptors, Point Feature Histograms and Shape Contexts, for the task of detecting segments of fallen tree stems. Although

we used full waveform data in our study, our algorithm does not rely on radiometric information and hence discrete return input data is also admissible. The results show that the aforementioned descriptors, when applied simultaneously, are able to provide somewhat complementary information to the classifier, thereby exhibiting a synergy effect and yielding overall accuracy rates beyond 90%. The proposed method can serve as an intermediate step for a system aiming at object-level detection (stems of entire fallen trees as opposed to merely segments). An interesting research direction concerns picking a set of non-overlapping segments and merging them under some optimality criterion to obtain object-level results. Another interesting question arises regarding the generalization capacity of classifiers based on the aforementioned features, inhibited by the inherent high-dimensionality of expressive shape descriptors. Currently, the results show some potential for generalizing across different forest stands. Perhaps certain dimensionality reduction schemes could be applied in order to improve this ability. Finally, we recognize the necessity to test our approach on data for which ground truth from a different source is available in order to assess its performance in a more objective manner.

6 Bibliography

ANTONARAKIS, A.S.; RICHARDS, K.S. & BRASINGTON, J., B., 2008: Object-based land cover classification using airborne LiDAR. Remote Sensing of Environment, **112** (6), pp. 2988-2998.

BLANCHARD, S.; JAKUBOWSKI, M. & KELLY M., 2011: Object-Based Image Analysis of Downed Logs in Disturbed Forested Landscapes Using Lidar. Remote Sensing, **3** (11), pp. 2420-2439.

BREIMAN, L., 2001: Random forests. Machine Learning, **45** (1), pp. 5-32.

CHANG, C.-C. & LIN, C.-J., 2011: LIBSVM: A Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology, **2** (3), article no. 27.

CULP, M.; JOHNSON, K. & MICHAILIDES, G., 2007: ada: An R Package for Stochastic Boosting. Journal of Statistical Software, **17** (2), pp. 1-27.

ELMQVIST, M., 2002: Ground Surface Estimation from Airborne Laser Scanner Data Using Active Shape Models. Photogrammetric Computer Vision-ISPRS Commission III Symposium, **XXXIV** (3A), pp. 114-118.

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 Ecology and Management, **174** (1-3), pp. 51–63.

FREEDMAN, B.; ZELAZNY, V.; BEAUDETTE, D.; FLEMING, T.; JOHNSON, G.; FLEMMING, S.; GERROW, J. S.; FORBES, G. & WOODLEY, S., 1996: Biodiversity implications of changes in the quantity of dead organic matter in managed forests. Environmental Reviews, **4** (3), pp. 238–265.

FRIEDMAN, J., 2002: Stochastic gradient boosting. Computational Statistics & Data Analysis, **38** (3), pp. 367-378.

FROME, A.; HUBER, D.; KOLLURI, R.; BÜLOW, T. & MALIK, J., 2004: Recognizing Objects in Range Data Using Regional Point Descriptors. Proceedings of the 8th European Conference on Computer Vision, pp. 224-237.

HEURICH, M., 2008: Automatic recognition and measurement of single trees based on data from airborne laser scanning over the richly structured natural forests of the Bavarian Forest National Park. Forest Ecology and Management, **255**, pp. 2416–2433.

KANKARE, V.; RÄTY, M.; YU, X.; HOLOPAINEN, M.; VASTARANTA, M.; KANTOLA, T.; HYYPPÄ, J.; HYYPPÄ, H.; ALHO, P. & VIITALA, R., 2013: Single tree biomass modelling using airborne laser scanning. ISPRS Journal of Photogrammetry and Remote Sensing, **85**, pp. 66–73.

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**, pp. 73– 81.

LINDBERG, E.; OLOFSSON, K.; HOLMGREN, J. & OLSSON, H., 2012: Estimation of 3D vegetation structure from waveform and discrete return airborne laser scanning data. Remote Sensing of Environment, **118**, pp. 151–161.

LINDBERG, E.; HOLLAUS, M.; MÜCKE, W; FRANSSON, J. & PFEIFER, N., 2013: Detection of lying tree stems from airborne laser scanning data using a line template matching algorithm. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, **II-5** (W2), pp. 169-174.

MORSDORF, F.; KÖTZ, B.; MEIER, E.; ITTEN, K.I. & ALLGÖWER, B., 2006: Estimation of LAI and fractional cover from small footprint airborne laser scanning data based on gap fraction. Remote Sensing of Environment, **104** , pp. 50-61.

MÜCKE, W.; DEÁK, B.; SCHROIFF, A.; HOLLAUS, M. & PFEIFER, N., 2013: Detection of fallen trees in forested areas using small footprint airborne laser scanning data. Canadian Journal of Remote Sensing, **39** (S1), pp. 32-40.

REITBERGER, J.; KRZYSTEK, P. & STILLA, U., 2008: Analysis of full waveform LIDAR data for the classification of deciduous and coniferous trees. International Journal of Remote Sensing, **29** (5), pp. 1407 - 1431.

RUSU, R.B.; MARTON, Z.C.; BLODOW, N. & BEETZ, M., 2008: Learning Informative Point Classes for the Acquisition of Object Model Maps. Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision, pp. 643-650.

RUSU, R.B. & COUSINS, S., 2011: 3D is here: Point Cloud Library (PCL). Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) 2011.

WEAVER, J.K.; KENEFIC, L.S.; SEYMOUR, R.S. & BRISSETTE, J.C., 2009: Decaying wood and tree regeneration in the Acadian Forest of Maine, USA. Forest Ecology and Management, **257** (7), pp. 1623–1628.

WOODALL, C.W.; HEATH, L.S. & SMITH, J.E., 2008: National inventories of down and dead woody material forest carbon stocks in the United States: Challenges and opportunities. Forest Ecology and Management, **256** (3), pp. 221–228.

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. ISPRS Annals of the Protogrammetry, Remote Sensing and Spatial Information Sciences, **I-7**, pp. 359-364.

YAO, W.; KRZYSTEK, P. & HEURICH, M., 2012: Tree species classification and estimation of stem volume and DBH based on single tree extraction by exploiting airborne full-waveform LiDAR data. Remote Sensing of Environment, **123**, pp. 368–380.