Active learning approach to detecting standing dead trees from ALS point clouds combined with aerial infrared imagery

Przemyslaw Polewski[1,3], Wei Yao[1], Marco Heurich[2], Peter Krzystek[1], Uwe Stilla[3]
[1]Hochschule München, 80333 München, Germany
[2]Bavarian Forest National Park, 94481 Grafenau, Germany
[3]Technische Universität München, 80333 München, Germany

Abstract

Due to their role in certain essential forest processes, dead trees are an interesting object of study within the environmental and forest sciences. This paper describes an active learning-based approach to detecting individual standing dead trees, known as snags, from ALS point clouds and aerial color infrared imagery. We first segment individual trees within the 3D point cloud and subsequently find an approximate bounding polygon for each tree within the image. We utilize these polygons to extract features based on the pixel intensity values in the visible and infrared bands, which forms the basis for classifying the associated trees as either dead or living. We define a two-step scheme of selecting a small subset of training examples from a large initially unlabeled set of objects. In the first step, a greedy approximation of the kernelized feature matrix is conducted, yielding a smaller pool of the most representative objects. We then perform active learning on this moderate-sized pool, using expected error reduction as the basic method. We explore how the use of semi-supervised classifiers with minimum entropy regularizers can benefit the learning process. Based on validation with reference data manually labeled on images from the Bavarian Forest National Park, our method attains an overall accuracy of up to 89% with less than 100 training examples, which corresponds to 10% of the pre-selected data pool.

1. Introduction

Dead wood is known to sustain biodiversity in forest environments through providing habitat for plants and animals [4]. It also contributes to forest carbon stocks [26] and serves as a source of coarse woody debris, a factor in stand succession [6]. For these reasons, carrying out forest maintenance and management tasks as well as performing carbon dynamics and biodiversity investigations relies on the knowledge of the spatial distribution of dead wood, including standing dead trees referred to as snags.

Several approaches to the automatic detection of individual snags from remote sensing data have been proposed. In [27], the authors first perform a 3D segmentation within the ALS point cloud in order to produce individual tree clusters. Each cluster is then classified as either dead or living using radiometric and geometric features derived purely from ALS data. An alternative approach [1] relies entirely on multispectral imagery. An active contour method [10] is applied to delineate individual tree crowns (ITC) within the image. They are subsequently classified via features obtained from the available spectral bands (including infrared channels). Approaches for delineating ITCs specialized for hyperspectral imagery are also available, *e.g.* [22]. In a recent study [15], a segmentation of infrared images is conducted with prior information about the specific shape and appearance of dead trees. A potential problem is associated with the lack of 3D height information, which leads to confusing dead trees with patches of roads or open ground areas having a similar pixel intensity representation within the image.

The approach presented in this paper is based on a combination of the ALS point clouds with aerial infrared imagery. We integrate the good discriminative capabilities of the infrared channels for distinguishing dead and living vegetation [9] with a segmentation procedure that takes advantage of 3D information. We propose a two-step strategy for detecting individual dead trees. In the first step, similarly to [27] we segment the 3D point cloud into individual trees. For each tree segment, we find the corresponding patch in the georeferenced aerial image. We then extract features from the patch based on the spectral channel values and perform classification into dead or living trees.

In order to efficiently select examples for training the classifier while taking into account the limited resources of the human expert, we make use of the *active learning* paradigm. In pool-based active learning, the system is given a pool of unlabeled examples, a small initial training set, and a classifier family capable of producing a continuous

measure of confidence alongside the class label. The goal is for the system to iteratively select training examples from the pool, ask an external 'oracle' to label them, and retrain the classifier on the augmented training set. The chosen examples should be representative of the entire object pool and carry information which helps the classifier attain a higher generalization capability. The various active learning algorithms differ mostly by how they assess the utility, or 'informativeness' of the unlabeled samples. For a review of existing methods, see [19]. Active learning methods have also received much attention within the remote sensing community in recent years. Approaches tailored to the specific properties of remote sensing data have been developed. Pasolli *et al*. [13] propose to combine spectral and spatial information in a sample selection framework based on SVMs. Persello *et al*. [14] consider the cost of labeling a sample in addition to its informative value, resulting in a cost-sensitive selection scheme. Tuia *et al*. [23] perform a comparative study of active learning methods applied to remote sensing imagery. In this work, we utilize the algorithm known as *expected error reduction* [18], which is built around the idea to pick the sample which mostly reduces the estimated generalization error on the rest of the unlabeled pool. The error is approximated by the total entropy of the posterior class probabilities output by the classifier.

A technique related and somewhat complementary to active learning is *semi-supervised learning*, which attempts to make use of the unlabeled pool directly (without labeling). In this case, standard supervised learning methods are modified to integrate information about the unlabeled examples. A number of diverse methodologies exist within this broad category [2]. In the field of remote sensing, semi-supervised learning has been explored mainly for optical and multi/hyperspectral image classification (*e.g*. [8]) as well as segmentation (*e.g*. [12]). We turn our attention to entropy regularization methods [5]. This framework allows to enrich any discriminative classification model with an entropy term that encodes a preference for decision boundaries with classes that overlap as little as possible. The maximized function which yields the optimal classifier under these assumptions is a weighted difference of the likelihood on the labeled data and the total entropy over the unlabeled data. The relationship between expected error reduction and the entropy regularization framework is that the former selects examples based on the minimum total entropy over posterior distributions, while the latter actively minimizes this same entropy on the unlabeled set. Therefore, we hypothesize that using an entropy-regularized classifier within the error reduction framework may lead to faster convergence of the learning process and can thus be seen as a way of combining the semi-supervised and active learning paradigms. Note that the idea of combining these two frameworks is not new (*e.g*. [24], [29]). In particular, [3] describes a strategy similar to ours in the setting of learning the grasping motion of a robot. The authors also employ a logistic classifier with an entropy regularization term, however they apply an uncertainty sampling scheme as opposed to the expected error reduction framework.

We consider the main contributions of this paper (i) the entire processing pipeline for detecting standing dead trees from ALS point clouds and aerial infrared imagery as well as (ii) the idea to use the entropy-regularized logistic regression classifier with the expected error reduction algorithm for active learning along with the investigation of its performance on this real-life problem. The rest of this paper is structured as follows: in Section 2 we explain the technical details of our approach. Section 3 describes the study area, experiments and evaluation strategy. The results are presented and discussed in Section 4. Finally, the conclusions are stated in Section 5.

2. Methodology

The input data for the presented method consists of an ALS point cloud and a set of corresponding georeferenced aerial multispectral images resulting from a standard flight. The spectral bands should contain information which facilitates distinguishing between living and dead vegetation, such as the near infrared band. The ALS data can be obtained from either a discrete return or a full waveform system. Note that while the radiometric information such as point intensities and pulse widths is not required, the tree segmentation algorithm is able to make use of it if available.

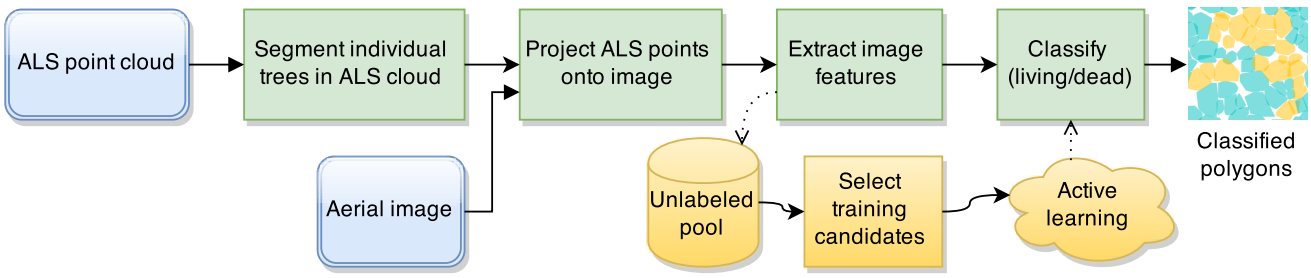


Figure 1: Overview of snag detection strategy.

For the aerial images, it is assumed that the exterior camera orientation as well as the camera parameters are known. The snag detection strategy is composed of two stages. In the first stage, point clusters associated with individual trees are found solely based on the 3D point cloud. The next stage uses the obtained tree segments to calculate approximate bounding polygons on the image plane and extract features from the corresponding image patches, thus enabling classification. The selection of examples for training the classifier is part of our method and is achieved in two steps. First, an unsupervised pre-selection of training set candidates is performed based on a greedy approximation of the feature matrix. Then, we carry out an active learning procedure. The resulting classifier can be reused for processing new test plots providing they have similar properties to the original input plot. The output of the entire pipeline is the set of bounding polygons (and associated 3D segments) along with their classification as either dead or living trees. The entire processing pipeline is visualized in Fig. 1.

2.1. Dead tree detection pipeline

2.1.1 Individual tree segmentation. In the first step of the strategy, we start with the raw ALS point cloud. We apply the single tree segmentation approach by Reitberger *et al*. [17]. We chose this method based on our prior investigations which revealed that performing a segmentation on the 3D point cloud can significantly increase the tree detection rate in all canopy layers compared to using only the Canopy Height Model (CHM). This method first determines the local maxima of the CHM and regards them as the initial tree positions. The point cloud is then segmented using the Normalized Cut clustering algorithm [20], based on a point similarity function which incorporates spatial proximity, prior knowledge about tree positions from the CHM and radiometric information (if available). As a result of this step, we obtain subsets of the original point cloud which correspond to individual segmented trees.

Calculating bounding polygons. For each 3D tree segment, we calculate a corresponding region on the aerial image plane. Since there is usually a large margin of overlap between neighboring aerial photographs, the same area is represented in more than one image. To minimize perspective distortion, we pick the image whose center is closest to the planimetric centroid of the segment's points. The convex hull of the points projected onto the image is then calculated, yielding an approximate bounding polygon for the tree. An example 3D segment and its associated polygon are depicted in Fig. 2.

2.1.2 Feature extraction. The availability of the bounding polygon makes it possible to define features at the object (tree) level, as opposed to only the pixel level. Specifically,

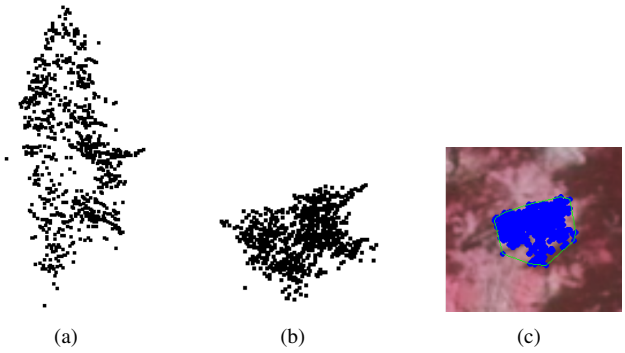


Figure 2: Projecting 3D points onto image. (a) Tree point cloud - side view, (b) Tree point cloud - top view, (c) Projected points and bounding polygon.

we utilize the per-channel intensity means of pixels inside the polygon as well as their cross-channel covariance matrix. At this stage, it is also possible to include LiDAR-derived features such as the ones used in [27], *e.g*. mean point intensity/pulse width, tree geometry, and gap fraction, as well as properties associated with the shape of the bounding polygon itself. However, our preliminary investigations indicated that introducing these auxiliary features does not significantly bolster the discriminative capabilities compared to pure image features.

2.1.3 Classification. We apply logistic regression [7] to classify the bounding polygons as either dead or living trees based on the features discussed in Sec. 2.1.2. This choice is motivated by several factors. First, the active learning approach which is used in this work relies on retraining the classifier a vast number of times. Fortunately, the optimization objective associated with training a logistic regression model is convex in the model weights and amenable to solving via the iteratively reweighted least squares method, which results in feasible computation times. Furthermore, the aforementioned convexity makes it easier to optimize the entropy-regularized variation of the classifier through deterministic annealing (see Sec. 2.3.2). Finally, although we deliberately perform the investigation using ordinary logistic regression on a small feature set for performance reasons, the methods presented in this work are easily generalizable to a kernelized model, which is known to produce classification performance on par with state-of-the-art methods such as the Support Vector Machine (see [28]).

Logistic regression. Let $x_i \in \mathcal{X}, i = 1..N$ denote N d-dimensional feature vectors of training examples and $y_i \in \{0, 1\}$ their corresponding binary labels. The conditional mean $E(Y_i|x_i)$ of the decision variable given the features is equal to the probability that object i has the label 1. The

logistic regression model relates the conditional mean to a linear regressor of the features through the logistic function:

$$E(Y_i|x_i) = P(Y_i = 1|x_i) = (1 + exp(-(\beta_0 + \beta x_i)))^{-1} \quad (1)$$

Training the model amounts to maximizing the joint log-likelihood of the training examples with respect to the weights $\theta = (\beta_0, \beta)$. Let $g_\theta(x_i) = P(Y_i = 1|x_i; \theta)$:

$$\begin{aligned} \ell(\theta) &= \sum_{i=1}^{N} lnP(Y = y_i|x_i; \theta) \\ &= \sum_{i=1}^{N} y_i ln[g_\theta(x_i)] + (1 - y_i) ln[1 - g_\theta(x_i)] \end{aligned} \quad (2)$$

The log-likelihood function in Eq. 2 can be optimized using iteratively reweighted least squares (IRLS) updates:

$$\theta^{k+1} \leftarrow \theta^k + (X^T W X)^{-1} X^T (y - p) \quad (3)$$

In the above, p denotes the vector of probabilities from the previous iterations: $p_i = g_{\theta^k}(x_i)$, X is the row-wise matrix of object feature vectors x_i with a prepended column of ones to account for the intercept term β_0, and W is a diagonal matrix with $W_{ii} = p_i(1 - p_i)$.

2.2. Pre-selecting training candidates

In a large area application of our method, we expect a vast number of initial unlabeled objects. Our active learning method of choice (Sec. 2.3.1) requires significant computational effort and cannot be realistically applied to the unfiltered data pool. Therefore, we introduce the candidate pre-selection step for the purpose of reducing the input object set for the active learning procedure to a feasible size from a computational complexity perspective. The authors list random sampling as a possible pre-selection method [18]. However, our target class (dead trees) is quite rare in the data (below 10%) and hence random sampling could potentially fail to capture its entire variability. Instead, we apply a greedy and unsupervised method for selecting the most representative objects based solely on their properties in the feature space. Specifically, we make use of the sparse greedy matrix approximation technique developed by Smola and Schölkopf [21]. This 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 distances 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:

$$||\bar{K} - K||^2_{Frob} = \sum_{i,j=1}^{N} (\bar{K} - K)^2_{ij} \quad (4)$$

This approximation will be done in a greedy fashion, where we iteratively pick a column K_i from K according to an evaluation criterion, orthogonalize the remaining columns on K_i, and check for convergence in the residual sum (Eq. 4). These ideas are formalized by Alg. 1.

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
```

Let $\bar{K}_i$ denote the approximation of the ith column of K. Using the expansion coefficient matrix T, we can write:

$$\bar{K}_i = \sum_{j=1}^{n} K_{i(j)} T_{ji} \quad (5)$$

The Frobenius norm (Eq. 4) can be expressed in terms of the column residuals:

$$||\bar{K} - K||^2_{Frob} = \sum_{i=1}^{N} ||K_i - \bar{K}_i||^2 \quad (6)$$

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

$$\sum_{i=1}^{N} ||K_i - \bar{K}_i^{I \cup \{j\}}||^2 = \sum_{i=1}^{N} ||K_i - \bar{K}_i^I - t_i(K_j - \bar{K}_j^I)||^2 \quad (7)$$

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):

$$t_i = ||K_j - \bar{K}_j^I||^{-2} < K_i - \bar{K}_i^I, K_j - \bar{K}_j^I > \quad (8)$$

Putting Eqs. 7 and 8 together, we can write the final form of the evaluation criterion $Q(j)$:

$$Q(j) = ||K_j - \bar{K}_j^I||^{-2} \sum_{i=1}^{N} < K_i - \bar{K}_i^I, K_j - \bar{K}_j^I >^2 \quad (9)$$

We execute Algorithm 1, iteratively picking the column which attains the highest value on the criterion given by Eq. 9 and calculating the residuals using Eq. 6. We use a standard Gaussian kernel with bandwidth parameter σ equal to the mean inter-sample distance to transform our input feature vectors into the design matrix. As an output of the algorithm, we obtain a list of the indices of the most representative objects in the unlabeled pool.

2.3. Active learning

2.3.1 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 (EER) criterion, introduced by Roy and McCallum [18]. To specify this method, we first assume a probabilistic setting in which 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 large pool U of unlabeled examples. Let $\hat{P}_T(Y|x)$ denote the learned distribution. Moreover, define the expected error of the classifier:

$$E(\hat{P}_T) = \int L(P(Y|x), \hat{P}_T(Y|x))P(x)dx \quad (10)$$

In Eq. 10, L indicates a *loss function* which quantifies the cost or loss associated with the discrepancy between the true and learned distributions. One of the more popular loss functions is the *log-loss* (assuming binary classification):

$$L(x) = -[P^1(x)ln(\hat{P}_T^1(x)) + (1 - P^1(x))ln(1 - \hat{P}_T^1(x))] \quad (11)$$

In the above, $P^1(x) = P(Y = 1|x)$. The log-loss is thus equivalent to the cross entropy between distributions P and $\hat{P}_T$. Note that average cross entropy is the effective log-likelihood function for logistic regression (cf. Eqs. 2, 11). In order to calculate the classifier error (Eq. 10), 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:

$$E(\hat{P}_T) = -\frac{1}{|U|} \sum_{x \in U} \sum_{y \in \{0,1\}} \hat{P}_T(y|x)ln(\hat{P}_T(y|x)) \quad (12)$$

Therefore, the expected error is effectively approximated using the entropy of the posterior class label probability of the 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. 13). The entire EER procedure is detailed by Alg. 2.

$$e(x) = \hat{P}_T(1|x)E(\hat{P}_{T \cup (x,1)}) + \hat{P}_T(0|x)E(\hat{P}_{T \cup (x,0)}) \quad (13)$$

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 ∈ {0, 1} do
6:                 (c', P̂) ← trainClassifier(T ∪ (x, y))
7:                 e(x) ← e(x) + P̂_T(y|x)E(P̂)
8:         x* ← argmin 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. The considerable effort of constantly retraining the classifier is mitigated by our example pre-selection step (Sec. 2.2) as well as the choice of using logistic regression. Finally, note that we used bagging to reduce the variance of the posterior probability estimates as suggested by the authors.

2.3.2 Entropy regularization. The entropy regularization framework, introduced by Grandvalet and Bengio [5], 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

the standard Shannon entropy conditioned on the features x as a measure of class overlap:

$$H(Y|X) = -\int \sum_{i\in\{0,1\}} ln[P(Y=i|x)]P(Y=i,x)dx \quad (14)$$

To avoid explicit modeling of the joint probability $P(y,x)$, the sample average over objects in the unlabeled pool is plugged into Eq. 14, resulting in an 'empirical' entropy:

$$H_{emp}(Y|X) = -\frac{1}{|U|}\sum_{x\in U}\sum_{i\in\{0,1\}} P(i|x)ln[P(i|x)] \quad (15)$$

The entropy term defined by Eq. 15 may be applied as a regularizer to any posterior distribution model:

$$C(\theta,\lambda;T,U) = \ell(\theta;T) - \lambda H_{emp}(Y|X;U) \quad (16)$$

The new optimization objective is therefore a sum of the original model's log-likelihood function 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. We use the aforementioned logistic regression classifier (Sec. 2.1.3) as the basic model. Unfortunately, the function C is no longer convex in θ, therefore local maxima may occur. To remedy this, we follow the authors' suggestion and apply the deterministic annealing expectation-maximization algorithm [25]. This approach implements an annealing process using an analogue of temperature $T = 1-\lambda$. The function C is made convex by assigning soft labels (probabilities) to all objects in the unlabeled pool. These probabilities are influenced by the inverse of the current temperature $\beta = 1/T$:

$$y_{soft}(x_i;\theta) = \frac{g_\theta(x_i)^\beta}{g_\theta(x_i)^\beta + (1-g_\theta(x_i))^\beta} \quad (17)$$

Starting at a high temperature $T = 1$, the log-likelihood term dominates in the function C. As the temperature is gradually decreased, the entropy term gains influence until the desired tradeoff (λ coefficient) is achieved. This final state corresponds to a local minimum of $C(\theta,\lambda)$. At every temperature level, the convex logistic regression subproblem is solved iteratively by updating the weights θ according to the IRLS algorithm (Eq. 3) until convergence.

2.3.3 Error Reduction with Entropy Regularization. Comparing Eqs. 12 and 15, we see that both expected error reduction and entropy regularization rely on entropy of the classifier's posterior on the unlabeled data pool. Although the former uses it as an estimate of the expected error, and the latter perceives it as a measure of class overlap, both framework seek to minimize this quantity. We hypothesize that using an entropy regularized classifier, which actively attempts to minimize the unlabeled pool entropy, with the EER framework could benefit the learning process.

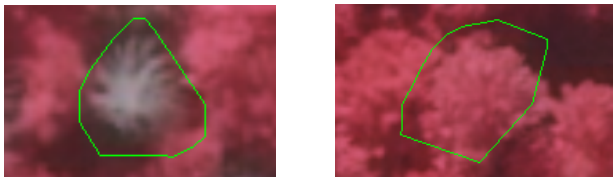
3. Experiments

3.1. Material

The basis for testing our approach was a 1x1 km plot located in 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. The study was performed in the mountain mixed forests zone consisting mostly of Norway spruce (*Picea abies*) and European beech (*Fagus sylvatica*). The dead wood originated from an outbreak of the spruce bark beetle (*Ips typographus*) in recent years [11]. Color infrared images were acquired in the leaf-on state during a flight campaign carried out 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 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 [16] to obtain a 3D point cloud.

3.2. Preparation of data

To mimic a real-world application in which the selection of training data is part of the problem, we did not pick an explicit test and training set. Instead, we first segmented the entire 1x1 km plot and extracted the image features for all polygons as described in Secs. 2.1.1-2.1.2. This resulted in ca. 44000 unlabeled polygons corresponding to single trees. We then performed two independent runs of the informative example selection strategy (Sec. 2.2), obtaining two subsets of polygons, referred to as **dataset A** and **dataset B**. Finally, for comparison purposes, we also created **dataset C** through random sampling. Each of the 3 datasets contains 1000 polygons. Fig. 3 depicts an aerial photograph with sample marked projected polygons.



(a) (b)

Figure 3: Polygons obtained from projecting 3D segments onto color infrared aerial image - (a) snag, (b) living tree.

3.3. Reference data and evaluation

We manually labeled all of the polygons in datasets A,B, and C using visual interpretation. To evaluate classification performance, we use the standard measure of overall accuracy, defined as the ratio of correctly classified objects to the size of the test set.

3.4. Detailed experiments

Effect of pre-selection. We investigate the effect of applying the informative training candidate strategy (Sec. 2.2). To do this, we train the classifier on subsets of each dataset and test it on the other two. We are interested in observing whether the classifier trained on the randomly sampled dataset C will prove inferior.

Semi-supervised learning performance. Here we concern ourselves with the classification performance of the entropy-regularized logistic classifier (Sec. 2.3.2). In particular, we want to find out if the semi-supervised version offers an improvement over the standard classifier. We report the classification accuracies on the entire dataset at various counts of the labeled instances which the classifier has access to, averaged over 100 random initializations.

Active learning performance. Finally, we examine the active learning performance for the proposed method (Sec. 2.3.3), compared to the original EER method as well as random sampling (RS). The algorithm starts with an initial training set of 4 positive and 4 negative examples and is given dataset A as the unlabeled pool. The active learning is performed for all methods up to a training sample count of 100, with dataset B used for testing. The results are averaged over 30 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.

4. Results and discussion

We address the performance of the pre-selection method. First, it should be noted that the pre-selection procedure significantly altered the class proportions in the datasets. While the randomly selected set C contained 7.6% dead trees, sets A and B nearly tripled this value to respectively 21.4% and 22.2%. This indicates that although the dead trees occur less frequently, they are more diverse in appearance than the living vegetation. For each dataset, we trained the logistic regression model on 50 random samples of 100 objects, and tested the classifiers on the other two datasets. The averaged results are shown in Table 1. We see that when the classifier is trained using datasets A and B, a loss of only 1-3 percentage points (pp) with respect to the baseline

Train \ Test	Dataset A	Dataset B	Dataset C
Dataset A	0.88	0.86	0.95
Dataset B	0.87	0.89	0.96
Dataset C	0.76	0.80	0.97

Table 1: Cross-dataset classification accuracy after pre-selection filter. Rows correspond to the training, columns to the test dataset (diagonal values from cross-validation).

cross-validation performance on the test set is observed. On the other hand, when training on dataset C, the deviations are now within 9-12 pp. This suggests that compared to the pre-selection method, RS extracted less information from the unlabeled pool about the variability of the target class's appearance. We drop dataset C from further analyses.

We now turn to the semi-supervised learning experiment. Fig. 4 depicts the averaged classification accuracy of the baseline LR and entropy-regularized LR models as a function of labeled training set size. The accuracy is always computed on the part of the dataset which is not labeled at the given step. For each size on the horizontal axis, a new training set is randomly sampled. Since the results are similar for both test sets, we only present the curve for dataset A. As expected, the impact of the unlabeled examples is greatest for small training set sizes, providing up to 10 pp gain over LR with 6 training examples. As the system obtains more labeled objects, the significance of the entropy term gradually vanishes since the classifier is able to generalize better using the newly added training items. The semi-supervised approach outperforms the baseline method for training set sizes of 5-20 elements, and therefore we hypothesize that it could also benefit the active learning during the first few iterations. Regarding the active learning

Figure 4: Average performance of logistic regression (LR) and its semi-supervised variation (ER-LR) on dataset A.

experiment, we considered two versions of integrating entropy regularization with error reduction (ER-ER). On the 'deep' level, the regularized classifier was used in Alg. 2 both in the external loop (line 3) and for the internal re-

training when assessing the expected errors (line 6). On the 'shallow' level, it was used only in line 3, while the internal loop employed the standard LR model. Fig. 5 shows the learning curves for all tested active learning methods. Both

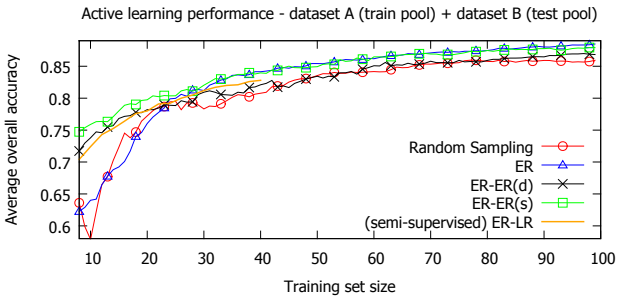


Figure 5: Average performance of active learning. ER refers to error reduction, while ER-ER denotes error reduction with entropy regularization: (d)eep and (s)hallow version. Semi-supervised ER-LR curve provided for comparison.

ER-ER methods outperform standard ER and RS for up to 23 training examples. This agrees with our expectations acquired from the semi-supervised experiment, because this is the training set size for which the influence of the unlabeled data term is strongest in the regularized classifier. For larger training sets, the performance of the ER-ER methods diverges: the s variant's becomes virtually identical to that of standard ER, while d's degenerates to RS levels. This indicates that the gain is mostly due to the improvement of the classifier through introduction of the unlabeled data, but the entropy-based selection criterion does not benefit from the classifier minimizing its entropy. For two classifiers c_1, c_2, the fact that c_1 was able to attain a lower entropy than c_2 on the unlabeled set U at its local maximum of the entropy and likelihood sum with fixed λ (Eq. 16) is not a good predictor for c_1 actually having a better accuracy than c_2 on U. This could be due to the fact that for a fixed λ, the resulting solutions of Eq. 16 for different training sets may not be comparable since they are trading off different log-likelihood values for reducing the entropy. In other words, the points of equilibrium between log-likelihood and entropy are not 'objective' in the sense that we can always adjust the λ coefficient to trade a little bit more of one for the other. A strategy which could help alleviate this problem is to instead try to minimize the entropy with a hard constraint stating that the log-likelihood may not decrease by more than ϵ % with respect to the original value obtained from the pure LR classifier (without the entropy term). On the whole, the gain of the ER methods over RS is moderate (2-4 pp), perhaps because the unlabeled pool was already biased towards informative examples through the greedy pre-selection step.

Finally, we compare the performance of our entire detection pipeline to existing methods. Yao *et al.* [27] used the same approach for segmenting trees in the ALS data, therefore comparing their reported classification accuracy of 73% to our rate of 89% indicates that features derived from infrared imagery are more discriminative for this task than pure LiDAR features. In light of Bhattarai *et al.*'s result of 81% [1], the gain of 8 pp. can perhaps be attributed to the used classifier and additional image channel covariance features. Comparison to Polewski *et al.* [15] is more difficult, because in that work reference polygons independent of the detection procedure were used. In contrast, our performance metrics are with respect to the polygons delineated by the tree segmentation algorithm. Therefore, errors introduced by over/undersegmentation carried over from the tree delineation step are not reflected in the accuracy. We were not able to quantify this erroneous segmentation influence due to lack of objective ground truth data.

5. Conclusions

We presented a method for detecting standing dead trees using ALS point clouds combined with aerial infrared imagery. This two-step approach first segments individual trees in the 3D point cloud, and subsequently projects each individual tree's points onto an infrared image. The convex hull of the projected points serves to extract the relevant image region and derive features for classification. We then proposed a two-tiered scheme for selecting training samples. The first stage is based on the greedy approximation of the kernelized feature matrix and does not require user interaction. We found that this method was able to identify informative examples more reliably than random sampling (RS), accounting for up to 12 percentage points (pp) of difference in performance. We observed that for kernel bandwidth values σ in the range between the average inter-sample distance μ and the average nearest-neighbor distance, the class distribution of the selected examples was stable. The farther the distance between σ and μ, the less snag examples were selected. In the second stage, we proposed an active learning procedure based on expected error reduction (ER), enriched with a semi-supervised classifier model. The overall gain from standard ER over RS was between 2-4 pp, however the introduction of the semi-supervised classifier resulted in enhancing the classification rate by up to 10 pp for small training sets. Our dead tree detection pipeline is able to achieve an overall accuracy of 89% using fewer than 100 training examples, which constitutes 10% of the data pool. As a future direction, we would like to evaluate the method using reference data independent of the 3D segmentation algorithm. Also, it would be interesting to find better ways to exploit the interaction of entropy minimization between ER and the entropy-regularized semi-supervised model. Finally, an assessment of the proposed active learning strategy on other types of remote sensing data could shed more light on its utility.

References

[1] N. Bhattarai, L. J. Quackenbush, L. Calandra, J. Im, and S. A. Teale. An automated object-based approach to detect Sirex-infestation in pines. In *ASPRS Annual Conference Proceedings*, 2012.

[2] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[3] A. N. Erkan. *Semi-supervised Learning via Generalized Maximum Entropy*. PhD thesis, New York, NY, USA, 2010. AAI3427925.

[4] J. Fridman and M. Walheim. Amount, structure, and dynamics of dead wood on managed forestland in Sweden. *Forest Ecol. Manag.*, 131(1-3):23–36, June 2000.

[5] Y. Grandvalet and Y. Bengio. Entropy regularization. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning*, pages 151–168. MIT Press, 2006.

[6] M. Harmon, J. Franklin, F. Swanson, P. Sollins, S. Gregory, J. Lattin, N. Anderson, S. Cline, N. Aumen, J. Sedell, G. Lienkaemper, K. C. Jr., and K. Cummins. Ecology of coarse woody debris in temperate ecosystems. volume 15 of *Advances in Ecological Research*, pages 133 – 302. Academic Press, 1986.

[7] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., 2001.

[8] L.-Z. Huo, P. Tang, Z. Zhang, and D. Tuia. Semisupervised classification of remote sensing images with hierarchical spatial similarity. *IEEE Geosci. Remote S.*, 12(1):150–154, Jan 2015.

[9] J. R. Jensen. *Remote Sensing of the Environment: An Earth Resource Perspective (2nd Edition)*. Prentice Hall, Upper Saddle River, May 2006.

[10] Y. Ke, W. Zhang, and L. J. Quackenbush. Active contour and hill climbing for tree crown detection and delineation. *Photogramm. Eng. Rem. S.*, 76(10):1169–1181, 2010.

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

[12] J. Munoz-Mari, D. Tuia, and G. Camps-Valls. Semisupervised classification of remote sensing images with active queries. *IEEE T. Geosci. Remote*, 50(10):3751–3763, Oct 2012.

[13] E. Pasolli, F. Melgani, D. Tuia, F. Pacifici, and W. Emery. SVM active learning approach for image classification using spatial information. *IEEE T. Geosci. Remote*, 52(4):2217–2233, April 2014.

[14] C. Persello, A. Boularias, M. Dalponte, T. Gobakken, E. Naesset, and B. Schölkopf. Cost-sensitive active learning with lookahead: Optimizing field surveys for remote sensing data classification. *IEEE T. Geosci. Remote*, 52(10):6652–6664, Oct 2014.

[15] P. Polewski, W. Yao, M. Heurich, P. Krzystek, and U. Stilla. 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, 2015.

[16] J. Reitberger, P. Krzystek, and U. Stilla. Analysis of full waveform LIDAR data for the classification of deciduous and coniferous trees. *Int. J. Remote Sens.*, 29:1407–1431, 2008.

[17] J. Reitberger, C. Schnörr, P. Krzystek, and U. Stilla. 3D segmentation of single trees exploiting full waveform LIDAR data. *ISPRS J. Photogramm.*, 64(6):561 – 574, 2009.

[18] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 441–448, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[19] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[20] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE T. Pattern Anal.*, 22(8):888–905, 2000.

[21] A. J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 911–918, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[22] G. Tochon, J. Fret, S. Valero, R. Martin, D. Knapp, P. Salembier, J. Chanussot, and G. Asner. On the use of binary partition trees for the tree crown segmentation of tropical rainforest hyperspectral images. *Remote Sens. Environ.*, 159(0):318 – 331, 2015.

[23] D. Tuia, M. Volpi, L. Copa, M. Kanevski, and J. Munoz-Mari. A survey of active learning algorithms for supervised remote sensing image classification. *IEEE J. Sel. Top. Signa.*, 5(3):606–617, June 2011.

[24] G. Tur, D. Hakkani-Tür, and R. E. Schapire. Combining active and semi-supervised learning for spoken language understanding. *Speech Commun.*, 45(2):171–186, Feb. 2005.

[25] N. Ueda and R. Nakano. Deterministic annealing EM algorithm. *Neural Networks*, 11(2):271 – 282, 1998.

[26] C. Woodall, L. Heath, and J. Smith. 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, July 2008.

[27] W. Yao, P. Krzystek, and M. Heurich. 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*, volume 1-7, pages 359–364, 2012.

[28] J. Zhu and T. Hastie. Kernel logistic regression and the import vector machine. In *J. Comput. Graph. Stat.*, pages 1081–1088. MIT Press, 2001.

[29] X. Zhu, J. Lafferty, and Z. Ghahramani. 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*, pages 58–65, 2003.