A SUPERVISED APPROACH FOR OBJECT EXTRACTION FROM TERRESTRIAL LASER POINT CLOUDS DEMONSTRATED ON TREES

Shahar Barnea [a], Sagi Filin [a], Victor Alchanatis [b]

[a] Dept. of Transportation and Geo-Information, Civil and Environmental Engineering Faculty, Technion – Israel Institute of Technology, Haifa, 32000, Israel - (barneas, filin)@technion.ac.il
[b] Institute of Agricultural Engineering, The Volcani Center, Bet Dagan, 50250, Israel - victor@volcani.agri.gov.il

Commission III, WG III/4

KEY WORDS: Object Recognition, Feature Extraction, Terrestrial Laser Scanner, Point Cloud, Algorithms

ABSTRACT:

Terrestrial laser scanning is becoming a standard for 3D modeling of complex scenes. Results of the scan contain detailed geometric information about the scene; however, the lack of semantic details is still a gap in making this data useable for mapping. In this paper we propose a framework for object recognition in laser scans. The 3D point cloud, which is the natural representation of scanners outcome, is a complex data structure to process, as it does not have an inherent neighborhood structure. We propose a polar representation which facilitates low-level image processing tasks, e.g. segmentation and texture modeling. Using attributes of each segment a feature space analysis is used to classify segments into objects. This process is followed by a fine-tuning stage based on graph-cut algorithm, which takes into consideration the 3D nature of the data. The proposed algorithm is demonstrated on tree extraction and tested on 18 urban scans containing complex objects in addition to trees. The experiments show the feasibility of the proposed framework.

1. INTRODUCTION

We address in this paper the problem of object extraction from 3D terrestrial laser point clouds. Such extraction becomes relevant with the growing use of terrestrial laser scanners for mapping purposes and for the reconstruction of objects in 3D space. Object extraction from terrestrial laser scanners has indeed been a research topic in recent years, ranging from reverse engineering problems, to building reconstruction, and forestry applications. In most cases a model driven approach is applied, where domain knowledge about the sought after object shape drives the reconstruction and recognition process. Rabanni (2006) models industrial installations by making use of predefined solid object model properties. Bienert et al. (2006) propose an ad-hoc approach for tree detection based on trimming the laser data at a certain height to separate the canopy from the ground and searching for stem patches. Such approaches cannot be generalized to other objects, and usually assume well defined shape of the sought after objects.

Alternative approaches, which can still be categorized as model driven, involve generating a database consisting of diverse instantiations of 3D objects. Upon the arrival of a new unseen data, they search for a good matching score between regions in the new data and the database objects. The matching score is usually calculated via key-features and spatial descriptors. Such models are reported in (Huber and Hebert, 2003; Huber et al., 2004) that show good results while using the spin image based descriptors, Frome et al. (2004) that introduce 3D shape and harmonic shape contexts descriptors for the recognition, and Mian et al. (2006) that present a matching score which is based on robust multidimensional table representation of objects. These methods require the generation of a massive object instantiations databases and are relatively specific to the modeled objects. As such they can hardly be considered applicable for natural objects and data arriving from terrestrial scans. Another approach, which is model driven as well, is based on the extraction of primitives (points, sticks, patches) and modeling inter-relation among them as a means to recover the object class. Pechuk et al. (2005) propose the extraction of primitives followed by mapping the links among them as cues for the recognition part. This is demonstrated on scenes containing a small number of well defined objects with relatively small number of primitives (e.g., chair, table).

Differing from model driven approaches we examine in this paper the possibility to extract objects form highly detailed geometric information using a small number of training data and with limited domain knowledge. We demonstrate this approach on tree detection primarily because of the shape complexity of trees. The approach we propose is based on 3D geometric variability measures and on learning shape characteristics. The proposed method begins with segmentation of the scans into regions which are then being classified into "object" and "not-object" segments. This classification generates a proposal of candidate objects that are then being refined. As we show, the choice of descriptive features makes the classification part, which is the core of the proposed model, successful even when based on a relatively small training.

2. METHODOLOGY

2.1 Data Representation

When dealing with range data, most approaches are applied to the point cloud in 3D space aiming to recover the 3D relationship between scans. The hard task is to calculate the descriptive information in the irregularly distributed laser point cloud. Nonetheless, as the angular spacing is fixed (defined by system specifications), regularity can be established when the data is transformed into a polar representation (Equation 1)

$$(x, y, z)^T = (\rho\cos\theta\cos\varphi, \rho\cos\theta\sin\varphi, \rho\sin\theta)^T \qquad (1)$$

with x, y and z the Euclidian coordinates of a point, θ and φ are the latitudinal and longitudinal coordinates of the firing direction respectively, and ρ is the measured range. When transformed, the scan will form a panoramic range image in which ranges are "intensity" measures. Figure 1 shows range data in the form of an image where the x axis represents the φ value, $\varphi \in (0,2\pi]$, and the y axis represents the θ value, $\theta \in (-\pi/4,\pi/4]$. The range image offers a compact, lossless, representation, but more importantly, makes data manipulations (e.g., derivative computation and convolution-like operations) simpler and easier to perform.

2.2 Segmentation

The transformation of the data panoramic range image allows the segmentation of the data using common image segmentation procedures. Recent works (e.g., Russell et al., 2006) have demonstrated how the application of segmentation processes for recognition tasks yields promising results both related in relation to object class recognition and to correct segmentation of the searched objects. Before segmenting the range images comes a data-cleaning phase that concerns filling void regions and the removal of isolated range measurements. Void regions are mainly the result of no-return areas in the scene (e.g., the skies) or object parts from which there is no reflectance. Isolated ranges appear detached from the ground and will relate to noise, leaves, or other small objects. No return regions are filled with a background value (maximal range), and for "no-reflectance" regions, ranges are assigned by neighboring objects. In Figure 1 the "no return" and the "no-reflectance" pixels marked with red.

For segmentation we use the Mean-Shift segmentation (Comaniciu and Meer, 2002), an adaptation of the mean-shift clustering algorithm that has proven successful for clustering non-parametric and complex feature space. The mean shift segmentation performs well in identifying homogeneous regions in the image. As can be seen in Figure 2, because of surface continuity and the general smoothness that characterize range data a tendency to join bigger regions into a single surface may exist. The algorithm can be controlled by two dominant parameters, the kernel size and permissible variability (range) within the segment. Tuning the variability to a small magnitude was useful in extracting "tree" segments (which are vertically dominant objects) as independent segments in the data. We note that even though under-segmented regions can be seen in other parts of the scan, this has little relevance to us.

2.3 Feature Space

The current part concerns isolating the tree related segments from the rest via classification. To perform the segment classification, a set of descriptive features for each of the segments should be computed. To keep the framework as general as possible we limit our search to low-level features. The sought after features should describe both the internal textural characteristics of the segment and characteristics of its silhouette shape. To keep the description simple, we seek a small set of descriptive features for characterizing the object. Limiting the set of features is useful for avoiding dimensionality related problems as well as overfitting concerns. The features we choose, consist of i) the sum the first-order derivatives, ii) absolute sum of the first-order derivatives, iii) the cornerness of the segment. These features (denoted *f1*, *f2* and *f3*) are computed per segment (L_i) as follows

$$\begin{aligned} f_1(L_i) &= \sum\left(d_\varphi(L_i) + d_\theta(L_i)\right) \\ f_2(L_i) &= \sum\left(\left|d_\varphi(L_i)\right| + \left|d_\theta(L_i)\right|\right) \\ f_3(L_i) &= \sum cornerness(L_i) \end{aligned} \tag{2}$$

with d_φ and d_θ the first-order derivatives of the polar image in the directions of its two axes. Since all three features involve summation and therefore are area dependent, they are normalized with respect to the segment area.

Analyzing the chosen features, the following observations can be seen. The first two features measure texture characteristics within the segmented area. Since trees have high range variability in all directions, the first feature should have low values (positive and the negative values cancel one another), while the second feature yields high values. The third feature, measures "cornerness" value for the area of the segment and its silhouette. For cornerness measure we use a corner operator we term min-max. The min-max operator considers points as corners when having "strong" gradients in all directions. In another formulation this can be stated as – a point is considered a corner even if the strength of the smallest gradient projection is big enough. With this formulation, corner detection can be seen as a min-max problem, by looking for the gradient projection in the minimal direction as the measure for the point "cornerness" (*Cn*). We leave the full mathematical development outside this text, due to space limitations, and present the formula for the cornerness measure in Equation (3)

$$Cn(\varphi_0,\theta_0,\alpha^*) = \sqrt{\sum W(\varphi-\varphi_0,\theta-\theta_0)\cdot\left(\begin{array}{l} \left(\frac{d\rho}{d\varphi}\right)^2 \frac{\sqrt{T^2+1}+1}{2\sqrt{T^2+1}} + \left(\frac{d\rho}{d\theta}\right)^2 \frac{\sqrt{T^2+1}-1}{2\sqrt{T^2+1}} \pm \\ \pm 2\frac{d\rho}{d\varphi}\frac{d\rho}{d\theta}\frac{T}{2\sqrt{T^2+1}} \end{array}\right)} \tag{3}$$

with

$$T = \frac{\sum W(\varphi-\varphi_0,\theta-\theta_0)\cdot 2\left(\frac{d\rho}{d\varphi}\cdot\frac{d\rho}{d\theta}\right)}{\sum W(\varphi-\varphi_0,\theta-\theta_0)\cdot\left(\left(\frac{d\rho}{d\varphi}\right)^2 - \left(\frac{d\rho}{d\theta}\right)^2\right)}$$

$$\alpha^* = \frac{1}{2}\tan^{-1}(T)$$

and *W*, a Gaussian window. The weighting function can be applied by simple convolution over the image and derivatives by φ and θ can be easily computed numerically. Generally, because of their complex shape and depth variability, tree related segments will tend to have high cornerness values.

In computing gradients, the need to control the varying object-to-background distances arises. The potential mixture between object and background may arise from the 2D representation of the 3D data, and may lead to very steep gradients when the background is distant, or shallower ones for closer ones. To handle this we erode the border pixels and do not sum their derivative value, thereby keeping the texture measures to "within" the segment only. Additionally, we trim the magnitude of possible derivative by a threshold to eliminate background effects, so that backgrounds that are closer and farther from the object (which is irrelevant for the classification task) will have the same contribution to the derivatives related features.

The three features as calculated for the segments of the demonstration scan are presented in Figure 3. One can see that tree related segments have average values with f_1 (in this sub-figure the most negative values is black and the most positive is white), and relatively high values both in f_2 and in f_3 (bright).

2.4 Classification

The computation of the features for each segment in the training set allows the creation of the feature space. Such feature space is illustrated in Figure 4 via three projections and an isometric view. The four views show the separablity of the tree and non-tree classes as achieved through these features. Green dots are segments that were marked as "trees", red dots are "not-tree" segments. As can be seen in Figure 4 even tough the two classes are separated, the data do not follow the classical form of two, well separated, hyper-Gaussian distributions. We therefore apply a non-parametric method for classification, using the k-Nearest Neighbors (k-NN) algorithm. Our choice is motivated by its simplicity and efficiency, but we note that other methods may prove suitable as well. The k-NN model is based on evaluating cardinality of a sample (unseen data) compared to the neighborhood in the training data. Following the extraction of the *k* nearest neighbors for the data sample, a voting procedure among them is performed. If more than *h* class I segments are within this subset, the unseen segment is recorded belonging to class I if not, class II is recorded. The k-NN model is greatly affected by the distance measures between elements, particularly when the different axes measure quantities in different units and scales. Because of the different measures we use, great differences are expected in scale and distribution, motivating the need to normalize the data. For normalization we use the whitening (Mahalanobis) transformation (Duda et al., 2000) that transforms data into the same scale and variance in all dimensions. If **X** is a training set of size Nx3, with N the number of segments distributed with ~{μ, $\mathbf{\Sigma}$}; using the SVD, $\mathbf{\Sigma}$ can be factored into $\mathbf{\Sigma}=\mathbf{UDV}^T$, where **U** is orthonormal, $\mathbf{UV}^T=\mathbf{I}$, and **D** a diagonal matrix. The transformed **X** is calculated by:

$$\mathbf{X}' = (\mathbf{D}^{-1/2}\mathbf{U}^T\mathbf{X}^T)^T \qquad (4)$$

with $\mathbf{X}'$ the transformed set. Following the whitening transformation the data is distributed with zero mean and unit variance in all three dimensions of the feature space. Distance measures in this space become uniform in all directions.

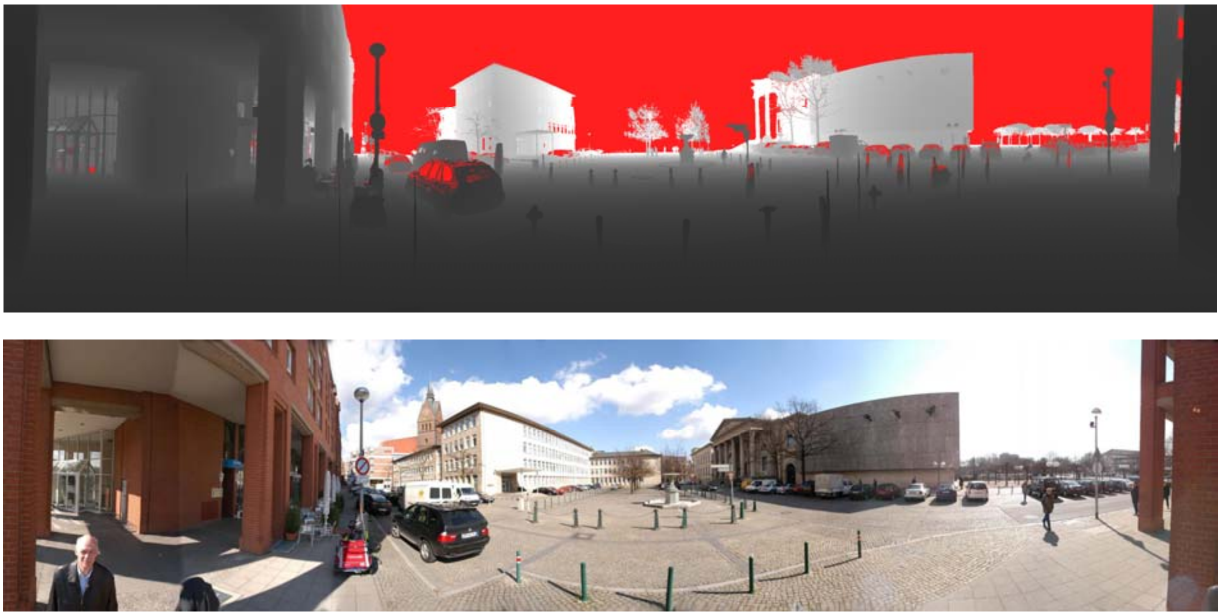

Figure 1. Top: Polar representation of terrestrial laser scans; the horizontal and vertical axes of the image represent the values of φ, θ respectively and intensity values as distances ρ (bright=far). "No-return" and "no-reflectance" pixels are marked in red. Bottom: panoramic view of the scanned scene acquired by a camera mounted on the scanner.

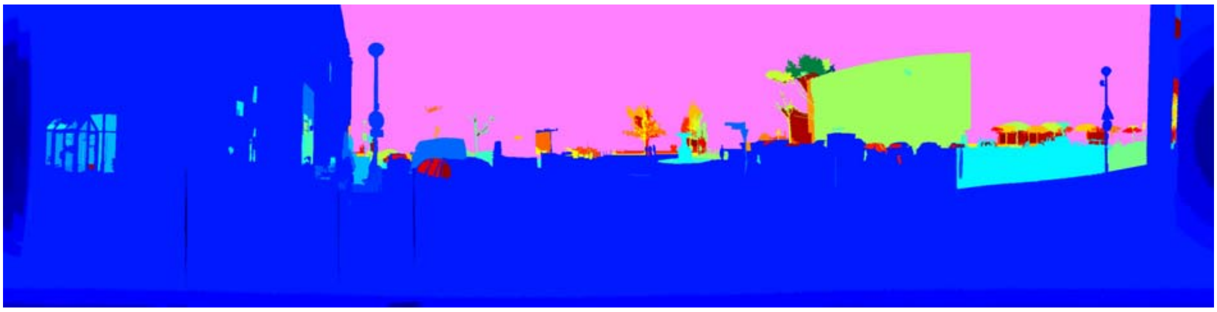

Figure 2. Results of the data segmentation using the mean-shift algorithm.

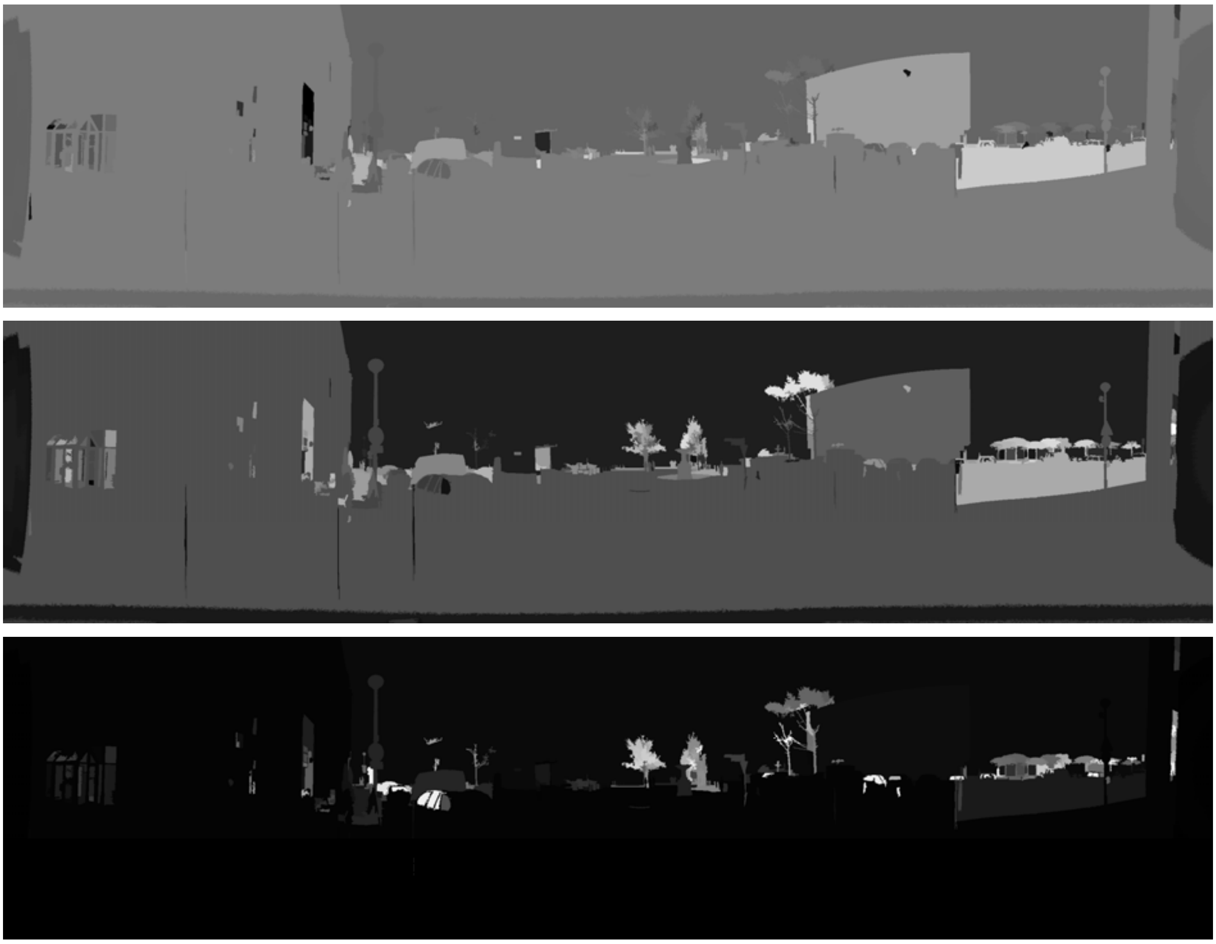

Figure 3. Segments weighted score for the three proposed features. Top: sum the first-order derivatives, middle: absolute sum of the first-order derivatives, bottom: the cornerness of the segment.

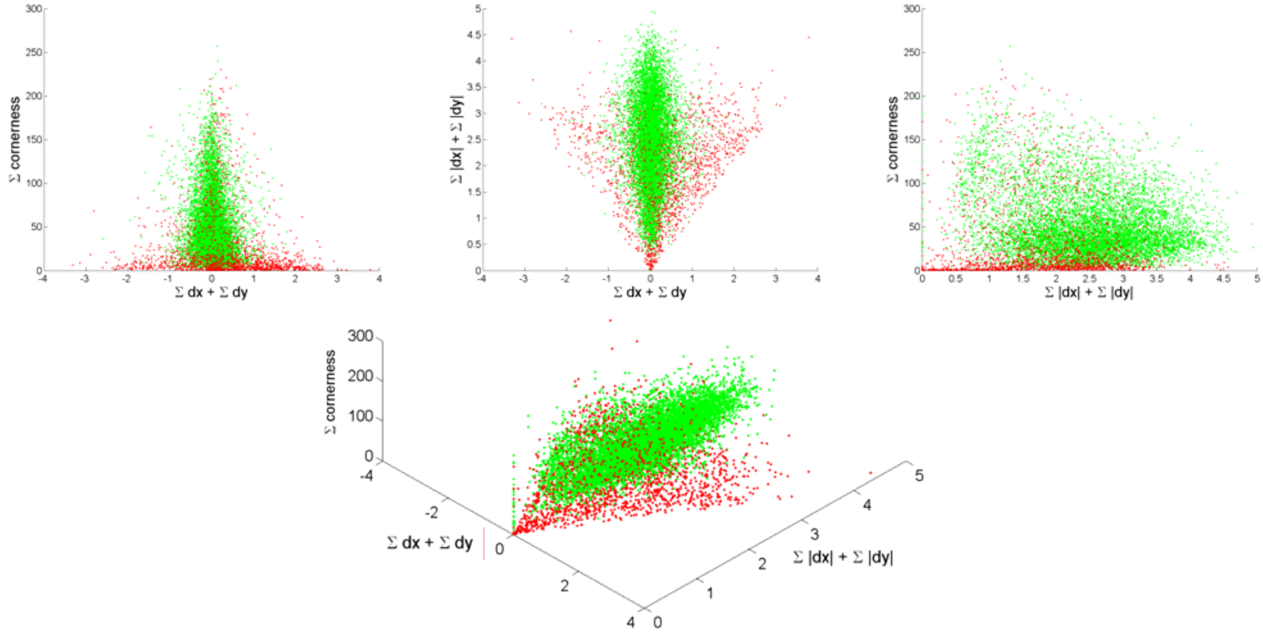

Figure 4. Four views of the feature space. The experiment contained 12351 segments that were manually classified. In green tree related segments and in red non-tree related segments.

The k-NN framework depends on number of neighbors checked (k), and on the cardinality parameter (h). Bigger k will make the model more general (when more samples are used to decide more information is weighted in) but less accurate (the extreme is where all samples are always used as neighbors). The choice of h affects the accuracy of the classification model. Setting h to a too small value, the model can become error prone, setting h too strictly, the number of false positives will decrease but on the expense of a large number of false negatives. An optimal value for h can be based on many considerations; our choice is based on finding a value that leads to the highest level of accuracy (ACC) as defined by

$$ACC = \frac{\text{True-Positive} + \text{True-Negative}}{\text{Positive} + \text{Negative}} \tag{5}$$

Such values can be derived by experimenting with different values for k and h. For each such trail a confusion matrix, C, is recorded

$$\mathrm{C} \equiv \begin{bmatrix} \textit{true positive} & \textit{false negative} \\ \textit{false positive} & \textit{true negative} \end{bmatrix} \tag{6}$$

and the one with the highest accuracy value (Eq. 5) determines both the h and k parameters.

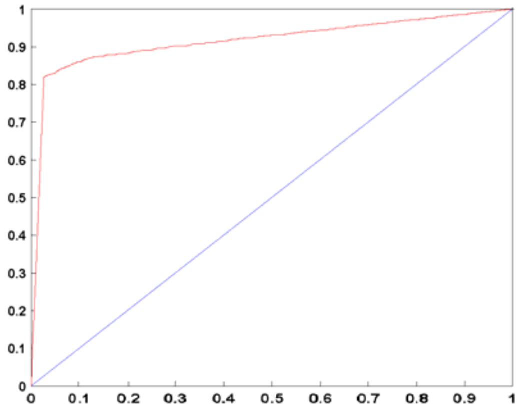

Figure 5. The ROC curve of the K-NN classifier.

2.5 Fine Tuning

So far, regions that have been identified via segmentation in 2D space have been classified as either trees or non-trees. Some of these segments are in fact sub-segments of the same tree (different part of the canopy or the stem), some segments may be a mixture of tree the background, and some segments may hold tree characteristics but are in fact non-tree objects. The fine-tuning phase aims linking segments that are part of the same tree, reducing to a minimum the number of false alarm detections, and separating mixture segments into object and background. Generally, this can be described as a split and merge problem among segments. We approach it differently by weighting the inter-relation between the individual points, so that neighboring points (by 3D proximity measures) will indicate potentially tight relations and therefore stronger utility in their link. The refinement phase revolves around an energy function of the form:

$$E = E_{data}(labeling) + E_{smooth}(labeling) \tag{7}$$

with E the total energy, E_{data} the energy related to the "wish" of laser point to maintain its original classification, and E_{smooth} the "wish" of highly connected points to have the same label. Labeling here refers to the binary value of the classified point in the point cloud and not to the outcome of the classification process. This energy function can be modeled by a graph, where each point in the cloud, i, is a vertex (V_i), and additionally, two more vertices, a source (s) and the sink (t) are added. The E_{data} elements are modeled through the weights assigned to edges linking each point and the source and each point and the sink. Each point (P_i) can have values of 0 or 1, depending on the output of the classification process. The weights on the edges are set according to

$$\begin{aligned} w(s,v_i) &= |p_i - \alpha| \\ w(v_i,t) &= 1 - |p_i - \alpha| \end{aligned} \tag{8}$$

with α the possible error in assigning a point. For representing the E_{smooth} part we search for the nearest neighbor point, j, for each point i in the cloud, and for each such pair (i,j) we build a link between the v_i and v_j whose weight is the inverse to the 3D Euclidian distance between the two points (the search for the nearest neighbor is performed via the Approximate Nearest Neighbor (ANN) method, (Arya et al., 1998)). Following the preparation of the graph, a graph-cut algorithm (Ford and Fulkerson, 1962) is applied to find the minimal cut (and the maximal flow) of the graph which also minimizes the energy function. The outcome of the graph cut refinement algorithm is separating between "tree" and "non-tree" points.

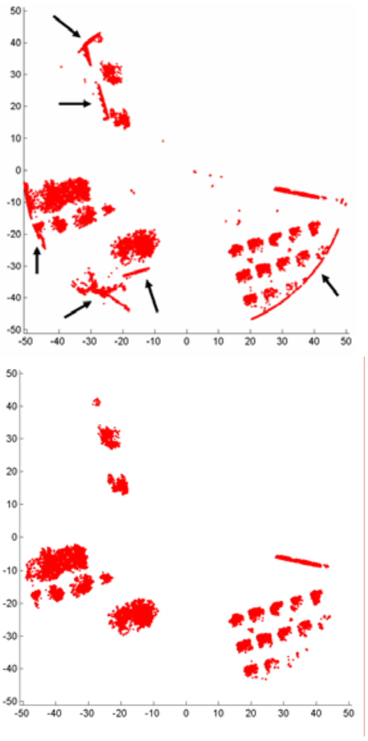

Figure 6. Fine tuning results, top: before, the arrows show areas that are not related to the object but lay on the background of it, bottom: the result of the execution of the algorithm. One can see how unwanted regions are filtered out.

3. RESULTS AND DISCUSSION

The algorithm was tested on 18 scans that were acquired in urban environment and, in addition to trees, contain cars, buildings, and other complex objects (see Figure 1). Each of the scans was segmented using the mean-shift segmentation; results of a typical segmentation can be seen in Figure 2. For the experiment tree objects in those scans were manually marked and all related points were assigned as the ground-truth. In all, the eighteen scans generated 12351 segments (~700 segments per scan).

As was noted the k-NN classification model depends on the choice of *k* and *h*. Following the formation of the feature space those parameters were studied by letting *k* vary between 1-11 while for each *k*, potential values for *h* ranged from 1-*k*. The highest accuracy value that was recorded was found to be ACC=0.87 when using *k*=9 and *h*=5. The corresponding confusion matrix was

$$C = \begin{bmatrix} 0.7129 & 0.0347 \\ 0.0977 & 0.1547 \end{bmatrix}$$

All confusion matrices resulting from this experiment (for all *k*'s and *h*'s) were plotted on a ROC curve (Figure 5). The area under the ROC curve is 0.9192, which is an evidence for good classification.

Learning by example models usually require a large training set data. Because of the relatively limited number of the available scans we used leave-one-out cross validation experiments. For each scan the training feature space was recovered from the remaining 17 scans. In this experiment, the algorithm is tested in its holistic form, including the refinement phase. As a performance metric we use the percentage of correctly recognized tree points (true-positive), correctly recognized background points (true-negative). The performance of this procedure is

$$C = \begin{bmatrix} 0.053 & 0.029 \\ 0.005 & 0.913 \end{bmatrix}$$

leading to ACC=0.966. One can see that the results both the high level of success of the complete algorithm and the contribution of the refinement phase. This improvement is also demonstrated in Figure 6. Figure 7 offers the tree classification results in the range image. From Figure 6 one can see how the background objects that were wrongly classified as trees are now eliminated from the results. In addition to the filtering out of wrongly classified points, new points which are highly connected to the tree were added. The results also show how trees in different distances (resolution) and ones that are partially occluded were detected by the algorithm.

4. CONCLUDING REMARKS

The paper has demonstrated that detection of objects with high level of accuracy can be reached by learning object characteristics from a small set of features and a limited number of samples. The detection scheme has managed identifying trees both in different depths (scales) and ones that were partially occluded. The small number of false alarm detections indicates the appropriateness of the selected features for the recognition. Using additional features and slight adaptations, the proposed approach can be further extended to detect different objects like buildings, cars, and others as well.

5. ACKNWOLEDGEMENT

The authors would like to thank Dr. Claus Brenner for making the data used for our tests available.

6. REFERENCES

Arya, S., Mount D. M., Netanyahu N. S., Silverman R., Wu A., 1998. An optimal algorithm for approximate nearest neighbor searching. Journal of the ACM, 45, 891–923.

Bienert, A., Maas, H.-G., Scheller, S. (2006). Analysis of The Information Content of Terrestrial Laserscanner Point Clouds For The Automatic Determination Of Forest Inventory Parameters. Workshop on 3D Remote Sensing in Forestry, 14th-15th Feb 2006, Vienna

Comaniciu D., Meer. P., 2002. Mean shift: A robust approach toward feature space analysis. IEEE trans. PAMI, 24:603–19.

Duda, R. O. Hart P. E. and Stork D. G., 2000. Pattern Classification 2nd ed. Wiley, 2000.

Ford, L., Fulkerson, D., (1962). Flows in Networks Princeton Univ. Press.

Frome A., Huber, R. Kolluri, T. Buelow, and J. Malik (2004). Recognizing Objects in Range Data Using Regional Point Descriptors. in Proc of ECCV 2004, 3, 224-237

Huber, D. Kapuria, A. Donamukkala, R., Hebert, M. (2004) Parts-Based 3D Object Recognition. Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, vol. 2, 82-89.

Huber D. and Hebert, M., (2003). Fully automatic registration of multiple 3D data sets. Image and Vision Computation 21(7):637-650.

Mian, A., Bennamoun, M., Owens, R., (2006). Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes. IEEE transactions on PAMI. 28(10), 1584-1601.

Pechuk M., Soldea O., Rivlin E., (2005) Function based classification from 3D Data via Generic and Symbolic Models. In AAAI, 950-955.

Rabanni T., 2006. Automatic reconstruction of Industrial Installations using point clouds and images. PhD thesis. NCG, publication on Geodesy 62.

Russell, C. B., Efros A., Sivic J., Freeman T. W., Zisserman A., (2006). Using Multiple Segmentations to Discover Objects and their Extent in Image Collections. in Proc. of CVPR 2006 2, 1605-1614.

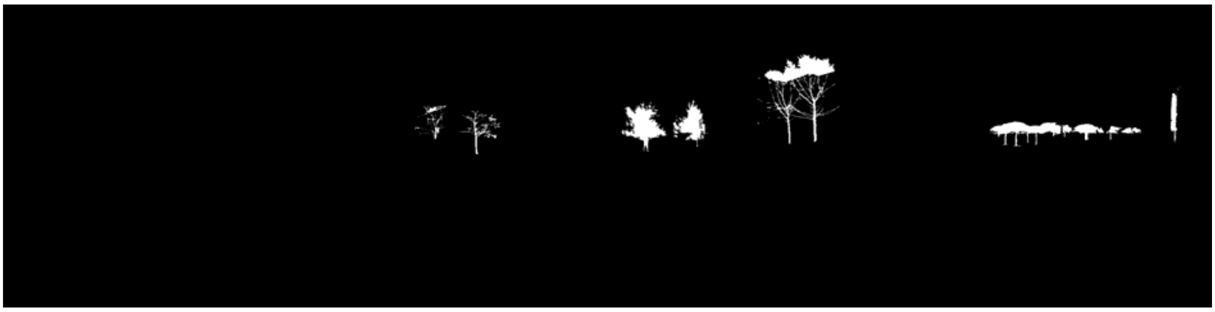

Figure 7. Results of the tree recognition algorithm.